

# Vending Machine with DQMH

---

## Table of Contents

1. Project description
2. Code component descriptions
  - 2.1. DQMH® modules
    - 2.1.1. Preamble
    - 2.1.2. Modules overview
    - 2.1.3. General UI.lvlib
    - 2.1.4. Money Control Pane.lvlib
    - 2.1.5. Product Selecting Pane.lvlib
    - 2.1.6. Message Pane.lvlib
    - 2.1.7. Configuration.lvlib
  - 2.2. Libraries
    - 2.2.1. Config File Management.lvlib
    - 2.2.2. SQLITE Inventory.lvlib
  - 2.3. Classes
3. VI descriptions
  - 3.1. DQMH® modules
    - 3.1.1. General UI.lvlib
    - 3.1.2. Money Control Pane.lvlib
    - 3.1.3. Product Selecting Pane.lvlib
    - 3.1.4. Message Pane.lvlib
    - 3.1.5. Configuration.lvlib
  - 3.2. Libraries
    - 3.2.1. Config File Management.lvlib
    - 3.2.2. SQLITE Inventory.lvlib
  - 3.3. Classes
4. Legal Information
  - 4.1. Document creation
    - 4.1.1. Antidoc
    - 4.1.2. Asciiidoc for LabVIEW™
    - 4.1.3. Graph Builder
  - 4.2. Product used in the project
    - 4.2.1. DQMH®

## 1. Project description

This project emulates a vending machine using the Delacor Queue Message Handler (DQMH) template. It has 5 modules: The UI in which all the other modules will be displayed, the Coin Module, in which the inserted coin are selected, the Product Selection Module, where the available products are displayed and can be selected; the Message Module, where the state of the machine, as well as the movements made, are shown; and the Configuration Module, where all the settings of the machine can be modified.

The configuration module allows the user to interact with the data base and with the .ini file as well. The data base stores all the product information, and the user can add product, delete them, and modify them in the Product Managment section in the Configuratiuon Module.

The .ini file stores the currency information used by the coins module. In this file the 4 coins denomination an values are stored, allowing the user to customize the coins that will be used, but only 4 of them. This configuration can be changed in the Currency Management section in de Configuration Module.

## 2. Code component descriptions

### 2.1. DQMH® modules

This section describes DQMH® module responsibilities and relationships.

#### 2.1.1. 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.



Refer to the DQMH® framework official [documentation](http://delacor.com/documentation/dqmh-html/) (<http://delacor.com/documentation/dqmh-html/>) 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:

```
digraph G611166 {
  rankdir=LR;
  edge[dir=both color=black arrowhead=normal arrowtail=none style=filled penwidth=1]
  node[color=black shape=box]
  "DQMH module / Lvlib"[color=black shape=component]
  "Vi"[color=skyblue shape=note]
}
```

### Events:

```
digraph G171093 {
  rankdir=LR;
  edge[dir=both color=black arrowhead=normal arrowtail=none style=filled penwidth=1]
  node[color=black shape=box]
  " "[color=white shape=box]
  " "[color=white shape=box]
  " "[color=white shape=box]
  " "[color=white shape=box]
  " " -> " " [label="Request to a DQMH module" dir=both color=forestgreen arrowhead=normal arrowtail=none
  style=filled penwidth=1];
  " " -> " " [label="Broadcast from a DQMH module" dir=both color=goldenrod arrowhead=normal arrowtail=none
  style=dashed penwidth=1];
}
```

### Start and Stop module callers:

```
digraph G362755 {
  rankdir=LR;
  edge[dir=both color=black arrowhead=normal arrowtail=none style=filled penwidth=1]
  node[color=black shape=box]
  "Start Module
  caller"[color=black shape=component]
  "Start Module"[color=yellowgreen shape=note]
  "Start Module" -> "Start Module
  caller" [label="Called by" dir=both color=yellowgreen arrowhead=odot arrowtail=inv style=filled penwidth=1];
}
```

```
digraph G937644 {
  rankdir=LR;
  edge[dir=both color=black arrowhead=normal arrowtail=none style=filled penwidth=1]
  node[color=black shape=box]
  "Stop Module
  caller"[color=black shape=component]
  "Stop Module"[color=tomato shape=note]
  "Stop Module" -> "Stop Module
  caller" [label="Called by" dir=both color=tomato arrowhead=odot arrowtail=inv style=dotted penwidth=1];
}
```

2.1.2. Modules overview

This project contains the following modules.

Table 1. Modules list

Singleton	Cloneable
General UI.lvlib	
Money Control Pane.lvlib	
Product Selecting Pane.lvlib	
Message Pane.lvlib	
Configuration.lvlib	

This graph represents the links between all DQMH modules.

```

digraph G23182 {
rankdir=LR;
edge[dir=both color=black arrowhead=normal arrowtail=none style=filled penwidth=1]
node[color=black shape=box]
"General UI"[color=black shape=component]
"Money Control Pane"[color=black shape=component]
"Configuration"[color=black shape=component]
"Product Selecting Pane"[color=black shape=component]
"Message Pane"[color=black shape=component]
"General UI" -> "General UI" [dir=both color=forestgreen arrowhead=normal arrowtail=none style=filled
penwidth=1];
"Money Control Pane" -> "Money Control Pane" [dir=both color=forestgreen arrowhead=normal arrowtail=none
style=filled penwidth=1];
"Configuration" -> "Money Control Pane" [dir=both color=forestgreen arrowhead=normal arrowtail=none style=filled
penwidth=1];
"Product Selecting Pane" -> "Product Selecting Pane" [dir=both color=forestgreen arrowhead=normal arrowtail=none
style=filled penwidth=1];
"Message Pane" -> "Product Selecting Pane" [dir=both color=forestgreen arrowhead=normal arrowtail=none
style=filled penwidth=1];
"Configuration" -> "Product Selecting Pane" [dir=both color=forestgreen arrowhead=normal arrowtail=none
style=filled penwidth=1];
"General UI" -> "Message Pane" [dir=both color=forestgreen arrowhead=normal arrowtail=none style=filled
penwidth=1];
"Message Pane" -> "Message Pane" [dir=both color=forestgreen arrowhead=normal arrowtail=none style=filled
penwidth=1];
"Money Control Pane" -> "Message Pane" [dir=both color=forestgreen arrowhead=normal arrowtail=none style=filled
penwidth=1];
"Product Selecting Pane" -> "Message Pane" [dir=both color=forestgreen arrowhead=normal arrowtail=none
style=filled penwidth=1];
"General UI" -> "Configuration" [dir=both color=forestgreen arrowhead=normal arrowtail=none style=filled
penwidth=1];
"Configuration" -> "Configuration" [dir=both color=forestgreen arrowhead=normal arrowtail=none style=filled
penwidth=1];
}

```

### 2.1.3. General UI.lvlib

**Type:** Singleton

**Responsibility:** The General UI serves as the display space for the rest of the modules to be shown. It has the subpanels where the other modules can be displayed, and only has 2 controls: the Settings button which calls the Configuration Module; and the Exit buttond, which finishes the program.

### Module Start/Stop calls

```

digraph G1006740 {
rankdir=LR;
edge[dir=both color=black arrowhead=normal arrowtail=none style=filled penwidth=1]
node[color=black shape=box]
"Start Module"[color=yellowgreen shape=note]
"Test General UI API"[color=skyblue shape=note]
"Stop Module"[color=tomato shape=note]
"General UI"[color=black shape=component]
"Start Module" -> "Test General UI API" [dir=both color=yellowgreen arrowhead=odot arrowtail=inv style=filled
penwidth=1];
"Stop Module" -> "General UI" [dir=both color=tomato arrowhead=odot arrowtail=inv style=dotted penwidth=1];
"Stop Module" -> "Test General UI API" [dir=both color=tomato arrowhead=odot arrowtail=inv style=dotted
penwidth=1];
}

```

*Table 2. Start and Stop module callers*

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

### Module relationship

```

digraph G714824 {
    rankdir=LR;
    edge[dir=both color=black arrowhead=normal arrowtail=none style=filled penwidth=1]
    node[color=black shape=box]
    "General UI"[color=slateblue shape=component]
    "Test General UI API"[color=skyblue shape=note]
    "Configuration"[color=black shape=component]
    "Test General UI API" -> "General UI" [dir=both color=forestgreen arrowhead=normal arrowtail=none style=filled penwidth=1];
    "General UI" -> "General UI" [dir=both color=forestgreen arrowhead=normal arrowtail=none style=filled penwidth=1];
    "General UI" -> "Test General UI API" [label=" " dir=both color=goldenrod arrowhead=normal arrowtail=none style=dashed penwidth=1];
    "General UI" -> "General UI" [label=" " dir=both color=forestgreen arrowhead=normal arrowtail=none style=filled penwidth=1];
    "General UI" -> "Configuration" [label=" " dir=both color=forestgreen arrowhead=normal arrowtail=none style=filled penwidth=1];
}

```

*Table 3. Requests callers*

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

*Table 4. Broadcasts Listeners*

Broadcast Name	Listeners
General UI.lvlib:Module Did Init.vi	Test General UI API.vi



Broadcast Name	Listeners
General UI.lvlib:Status Updated.vi	Test General UI API.vi
General UI.lvlib:Error Reported.vi	Test General UI API.vi
General UI.lvlib:Module Did Stop.vi	Test General UI API.vi
General UI.lvlib:Update Module Execution Status.vi	Test General UI API.vi

*Table 5. Used requests*

Module	Brodcasts
General UI.lvlib	General UI.lvlib:Get Module Execution Status.vi
Configuration.lvlib	Configuration.lvlib:Show Panel.vi

*Table 6. Registred broadcast*

Module	Brodcasts
—	—

#### 2.1.4. Money Control Pane.lvlib

**Type:** Singleton

**Responsibility:** The money pane shown 4 controls that simulates the coins configures in the currency management section. When a button is pushed, the credit increments that value in the Message Pane. It also has a cancel button that sets the cancel event and returns the credit.

Module Start/Stop calls

```

digraph G32770 {
rankdir=LR;
edge[dir=both color=black arrowhead=normal arrowtail=none style=filled penwidth=1]
node[color=black shape=box]
"Start Module"[color=yellowgreen shape=note]
"General UI"[color=black shape=component]
"Test Money Control Pane API"[color=skyblue shape=note]
"Stop Module"[color=tomato shape=note]
"Money Control Pane"[color=black shape=component]
"Start Module" -> "General UI" [dir=both color=yellowgreen arrowhead=odot arrowtail=inv style=filled penwidth=1];
"Start Module" -> "Test Money Control Pane API" [dir=both color=yellowgreen arrowhead=odot arrowtail=inv
style=filled penwidth=1];
"Stop Module" -> "General UI" [dir=both color=tomato arrowhead=odot arrowtail=inv style=dotted penwidth=1];
"Stop Module" -> "Money Control Pane" [dir=both color=tomato arrowhead=odot arrowtail=inv style=dotted
penwidth=1];
"Stop Module" -> "Test Money Control Pane API" [dir=both color=tomato arrowhead=odot arrowtail=inv style=dotted
penwidth=1];
}

```

*Table 7. Start and Stop module callers*

Function	Callers
Money Control Pane.lvlib:Start Module.vi	General UI.lvlib:Main.vi Test Money Control Pane API.vi
Money Control Pane.lvlib:Stop Module.vi	General UI.lvlib:Main.vi Money Control Pane.lvlib:Handle Exit.vi Test Money Control Pane API.vi

## Module relationship

```

digraph G23336 {
rankdir=LR;
edge[dir=both color=black arrowhead=normal arrowtail=none style=filled penwidth=1]
node[color=black shape=box]
"Money Control Pane"[color=slateblue shape=component]
"General UI"[color=black shape=component]
"Test Money Control Pane API"[color=skyblue shape=note]
"Configuration"[color=black shape=component]
"Message Pane"[color=black shape=component]
"General UI" -> "Money Control Pane" [dir=both color=forestgreen arrowhead=normal arrowtail=none style=filled
penwidth=1];
"Test Money Control Pane API" -> "Money Control Pane" [dir=both color=forestgreen arrowhead=normal arrowtail=none
style=filled penwidth=1];
"Money Control Pane" -> "Money Control Pane" [dir=both color=forestgreen arrowhead=normal arrowtail=none
style=filled penwidth=1];
"Configuration" -> "Money Control Pane" [dir=both color=forestgreen arrowhead=normal arrowtail=none style=filled
penwidth=1];
"Money Control Pane" -> "Test Money Control Pane API" [label=" " dir=both color=goldenrod arrowhead=normal
arrowtail=none style=dashed penwidth=1];
"Money Control Pane" -> "Money Control Pane" [label=" " dir=both color=forestgreen arrowhead=normal
arrowtail=none style=filled penwidth=1];
"Money Control Pane" -> "Message Pane" [label=" " dir=both color=forestgreen arrowhead=normal arrowtail=none
style=filled penwidth=1];
}

```

*Table 8. Requests callers*

Request Name	Callers
Money Control Pane.lvlib:Show Panel.vi	Test Money Control Pane API.vi
Money Control Pane.lvlib:Hide Panel.vi	Test Money Control Pane API.vi
Money Control Pane.lvlib:Get Module Execution Status.vi	Money Control Pane.lvlib:Start Module.vi Money Control Pane.lvlib:Obtain Broadcast Events for Registration.vi
Money Control Pane.lvlib:Show Diagram.vi	Test Money Control Pane API.vi
Money Control Pane.lvlib:Configuration Change.vi	Money Control Pane.lvlib:Main.vi Configuration.lvlib:Main.vi Test Money Control Pane API.vi

*Table 9. Broadcasts Listeners*

Broadcast Name	Listeners
Money Control Pane.lvlib:Module Did Init.vi	Test Money Control Pane API.vi
Money Control Pane.lvlib:Status Updated.vi	Test Money Control Pane API.vi
Money Control Pane.lvlib>Error Reported.vi	Test Money Control Pane API.vi
Money Control Pane.lvlib:Module Did Stop.vi	Test Money Control Pane API.vi
Money Control Pane.lvlib:Update Module Execution Status.vi	Test Money Control Pane API.vi

*Table 10. Used requests*

Module	Brodcasts
Money Control Pane.lvlib	Money Control Pane.lvlib:Get Module Execution Status.vi Money Control Pane.lvlib:Configuration Change.vi

Module	Broadcasts
Message Pane.lvlib	Message Pane.lvlib:Credit Update.vi Message Pane.lvlib:Cancel.vi

*Table 11. Registered broadcast*

Module	Broadcasts
—	—

### 2.1.5. Product Selecting Pane.lvlib

**Type:** Singleton

**Responsibility:** This pane allows the user to see and select a product to purchase. It has a listbox in which all existing products in stok will be shown. Products that are not available, said the ones that are not in stok (stock = 0), are not displayed.

When a poroduct is selected, the purchase button should be pushed in order to execute the purchase.

#### Module Start/Stop calls

```

digraph G67916 {
  rankdir=LR;
  edge[dir=both color=black arrowhead=normal arrowtail=none style=filled penwidth=1]
  node[color=black shape=box]
  "Start Module"[color=yellowgreen shape=note]
  "General UI"[color=black shape=component]
  "Test Product Selecting Pane API"[color=skyblue shape=note]
  "Stop Module"[color=tomato shape=note]
  "Product Selecting Pane"[color=black shape=component]
  "Start Module" -> "General UI" [dir=both color=yellowgreen arrowhead=odot arrowtail=inv style=filled penwidth=1];
  "Start Module" -> "Test Product Selecting Pane API" [dir=both color=yellowgreen arrowhead=odot arrowtail=inv style=filled penwidth=1];
  "Stop Module" -> "General UI" [dir=both color=tomato arrowhead=odot arrowtail=inv style=dotted penwidth=1];
  "Stop Module" -> "Product Selecting Pane" [dir=both color=tomato arrowhead=odot arrowtail=inv style=dotted penwidth=1];
  "Stop Module" -> "Test Product Selecting Pane API" [dir=both color=tomato arrowhead=odot arrowtail=inv style=dotted penwidth=1];
}

```

*Table 12. Start and Stop module callers*

Function	Callers
Product Selecting Pane.lvlib:Start Module.vi	General UI.lvlib:Main.vi Test Product Selecting Pane API.vi

Function	Callers
Product Selecting Pane.lvlib:Stop Module.vi	General UI.lvlib:Main.vi Product Selecting Pane.lvlib:Handle Exit.vi Test Product Selecting Pane API.vi

Module relationship



Table 13. Requests callers

Request Name	Callers
Product Selecting Pane.lvlib:Show Panel.vi	Test Product Selecting Pane API.vi
Product Selecting Pane.lvlib:Hide Panel.vi	Test Product Selecting Pane API.vi
Product Selecting Pane.lvlib:Get Module Execution Status.vi	Product Selecting Pane.lvlib:Start Module.vi Product Selecting Pane.lvlib:Obtain Broadcast Events for Registration.vi
Product Selecting Pane.lvlib:Show Diagram.vi	Test Product Selecting Pane API.vi

Request Name	Callers
Product Selecting Pane.lvlib:Update Products.vi	Message Pane.lvlib:Main.vi Product Selecting Pane.lvlib:Main.vi Configuration.lvlib:Main.vi Test Product Selecting Pane API.vi

Table 14. Broadcasts Listeners

Broadcast Name	Listeners
Product Selecting Pane.lvlib:Module Did Init.vi	Test Product Selecting Pane API.vi
Product Selecting Pane.lvlib:Status Updated.vi	Test Product Selecting Pane API.vi
Product Selecting Pane.lvlib:Error Reported.vi	Test Product Selecting Pane API.vi
Product Selecting Pane.lvlib:Module Did Stop.vi	Test Product Selecting Pane API.vi
Product Selecting Pane.lvlib:Update Module Execution Status.vi	Test Product Selecting Pane API.vi

Table 15. Used requests

Module	Broadcasts
Product Selecting Pane.lvlib	Product Selecting Pane.lvlib:Get Module Execution Status.vi Product Selecting Pane.lvlib:Update Products.vi
Message Pane.lvlib	Message Pane.lvlib:Purchase action.vi

Table 16. Registered broadcast

Module	Broadcasts
—	—

### 2.1.6. Message Pane.lvlib

**Type:** Singleton

**Responsibility:** This pane receives information from the Product Selecting Pane and the Money Control Pane. This pane shows the credit that is incremented by the MCP, as well as the product selected when a purchase request is made. When a purchase is requested, the Message Pane displays if the purchase can be done or not according to the available credit and the price of the product.

When a purchase or a cancel is made, the change space displays the returning money to the user.

Module Start/Stop calls

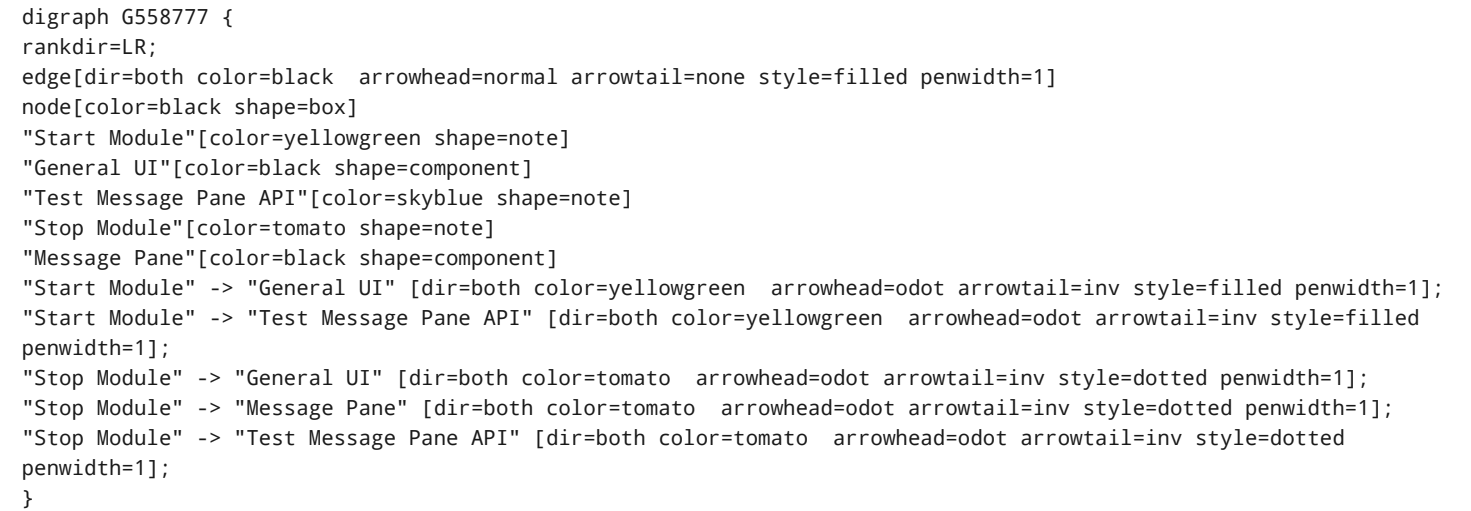


Table 17. Start and Stop module callers

Function	Callers
Message Pane.lvlib:Start Module.vi	General UI.lvlib:Main.vi Test Message Pane API.vi
Message Pane.lvlib:Stop Module.vi	General UI.lvlib:Main.vi Message Pane.lvlib:Handle Exit.vi Test Message Pane API.vi

Module relationship

```

digraph G64924 {
rankdir=LR;
edge[dir=both color=black arrowhead=normal arrowtail=none style=filled penwidth=1]
node[color=black shape=box]
"Message Pane"[color=slateblue shape=component]
"General UI"[color=black shape=component]
"Test Message Pane API"[color=skyblue shape=note]
"Money Control Pane"[color=black shape=component]
"Product Selecting Pane"[color=black shape=component]
"General UI" -> "Message Pane" [dir=both color=forestgreen arrowhead=normal arrowtail=none style=filled penwidth=1];
"Test Message Pane API" -> "Message Pane" [dir=both color=forestgreen arrowhead=normal arrowtail=none style=filled penwidth=1];
"Message Pane" -> "Message Pane" [dir=both color=forestgreen arrowhead=normal arrowtail=none style=filled penwidth=1];
"Money Control Pane" -> "Message Pane" [dir=both color=forestgreen arrowhead=normal arrowtail=none style=filled penwidth=1];
"Product Selecting Pane" -> "Message Pane" [dir=both color=forestgreen arrowhead=normal arrowtail=none style=filled penwidth=1];
"Message Pane" -> "Test Message Pane API" [label=" " dir=both color=goldenrod arrowhead=normal arrowtail=none style=dashed penwidth=1];
"Message Pane" -> "Product Selecting Pane" [label=" " dir=both color=forestgreen arrowhead=normal arrowtail=none style=filled penwidth=1];
"Message Pane" -> "Message Pane" [label=" " dir=both color=forestgreen arrowhead=normal arrowtail=none style=filled penwidth=1];
}

```

**Table 18. Requests callers**

Request Name	Callers
Message Pane.lvlib:Show Panel.vi	Test Message Pane API.vi
Message Pane.lvlib:Hide Panel.vi	Test Message Pane API.vi
Message Pane.lvlib:Get Module Execution Status.vi	Message Pane.lvlib:Start Module.vi Message Pane.lvlib:Obtain Broadcast Events for Registration.vi
Message Pane.lvlib:Show Diagram.vi	Test Message Pane API.vi
Message Pane.lvlib:Credit Update.vi	Money Control Pane.lvlib:Main.vi Test Message Pane API.vi
Message Pane.lvlib:Purchase action.vi	Product Selecting Pane.lvlib:Main.vi Test Message Pane API.vi
Message Pane.lvlib:Cancel.vi	Money Control Pane.lvlib:Main.vi Test Message Pane API.vi



*Table 19. Broadcasts Listeners*

Broadcast Name	Listeners
Message Pane.lvlib:Module Did Init.vi	Test Message Pane API.vi
Message Pane.lvlib:Status Updated.vi	Test Message Pane API.vi
Message Pane.lvlib:Error Reported.vi	Test Message Pane API.vi
Message Pane.lvlib:Module Did Stop.vi	Test Message Pane API.vi
Message Pane.lvlib:Update Module Execution Status.vi	Test Message Pane API.vi

*Table 20. Used requests*

Module	Brodcasts
Product Selecting Pane.lvlib	Product Selecting Pane.lvlib:Update Products.vi
Message Pane.lvlib	Message Pane.lvlib:Get Module Execution Status.vi

*Table 21. Registred broadcast*

Module	Brodcasts
—	—

### 2.1.7. Configuration.lvlib

**Type:** Singleton

**Responsibility:** This panel allow user to change the settings of the machine. it has 2 section: the Product Management, and the Currency Management.

In the first one user can add, delete and modify products on the data base. it allos user to change prices of a single or all products as well as the stock of every product.

In the Currency Management the user can change the denomination and value of the four coind on the Money Control Pane.

### Module Start/Stop calls

```
digraph G902169 {
rankdir=LR;
edge[dir=both color=black arrowhead=normal arrowtail=none style=filled penwidth=1]
node[color=black shape=box]
"Start Module"[color=yellowgreen shape=note]
"General UI"[color=black shape=component]
"Test Configuration API"[color=skyblue shape=note]
"Stop Module"[color=tomato shape=note]
"Configuration"[color=black shape=component]
"Start Module" -> "General UI" [dir=both color=yellowgreen arrowhead=odot arrowtail=inv style=filled penwidth=1];
"Start Module" -> "Test Configuration API" [dir=both color=yellowgreen arrowhead=odot arrowtail=inv style=filled penwidth=1];
"Stop Module" -> "Configuration" [dir=both color=tomato arrowhead=odot arrowtail=inv style=dotted penwidth=1];
"Stop Module" -> "Test Configuration API" [dir=both color=tomato arrowhead=odot arrowtail=inv style=dotted penwidth=1];
}
```

Table 22. Start and Stop module callers

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

Module relationship

```
digraph G476659 {
rankdir=LR;
edge[dir=both color=black arrowhead=normal arrowtail=none style=filled penwidth=1]
node[color=black shape=box]
"Configuration"[color=slateblue shape=component]
"General UI"[color=black shape=component]
"Test Configuration API"[color=skyblue shape=note]
"Money Control Pane"[color=black shape=component]
"Product Selecting Pane"[color=black shape=component]
"General UI" -> "Configuration" [dir=both color=forestgreen arrowhead=normal arrowtail=none style=filled penwidth=1];
"Test Configuration API" -> "Configuration" [dir=both color=forestgreen arrowhead=normal arrowtail=none style=filled penwidth=1];
"Configuration" -> "Configuration" [dir=both color=forestgreen arrowhead=normal arrowtail=none style=filled penwidth=1];
"Configuration" -> "Test Configuration API" [label=" " dir=both color=goldenrod arrowhead=normal arrowtail=none style=dashed penwidth=1];
"Configuration" -> "Money Control Pane" [label=" " dir=both color=forestgreen arrowhead=normal arrowtail=none style=filled penwidth=1];
"Configuration" -> "Product Selecting Pane" [label=" " dir=both color=forestgreen arrowhead=normal arrowtail=none style=filled penwidth=1];
"Configuration" -> "Configuration" [label=" " dir=both color=forestgreen arrowhead=normal arrowtail=none style=filled penwidth=1];
}
```

Table 23. Requests callers

--

<b>Request Name</b>	<b>Callers</b>
Configuration.lvlib:Show Panel.vi	General UI.lvlib:Main.vi Test Configuration API.vi
Configuration.lvlib:Hide Panel.vi	Test Configuration API.vi
Configuration.lvlib:Get Module Execution Status.vi	Configuration.lvlib:Start Module.vi Configuration.lvlib:Obtain Broadcast Events for Registration.vi
Configuration.lvlib:Show Diagram.vi	Test Configuration API.vi

*Table 24. Broadcasts Listeners*

<b>Broadcast Name</b>	<b>Listeners</b>
Configuration.lvlib:Module Did Init.vi	Test Configuration API.vi
Configuration.lvlib:Status Updated.vi	Test Configuration API.vi
Configuration.lvlib>Error Reported.vi	Test Configuration API.vi
Configuration.lvlib:Module Did Stop.vi	Test Configuration API.vi
Configuration.lvlib:Update Module Execution Status.vi	Test Configuration API.vi

*Table 25. Used requests*

<b>Module</b>	<b>Brodcasts</b>
Money Control Pane.lvlib	Money Control Pane.lvlib:Configuration Change.vi
Product Selecting Pane.lvlib	Product Selecting Pane.lvlib:Update Products.vi
Configuration.lvlib	Configuration.lvlib:Get Module Execution Status.vi

*Table 26. Registred broadcast*

Module	Broadcasts
—	—

## 2.2. Libraries

This section describes the libraries contained in the project.

### 2.2.1. Config File Management.lvlb

Gives functions to manipulate, read and write config files.

### 2.2.2. SQLITE Inventory.lvlb

Gives the VIs required to store the inventory in a SQLite data base.

## 2.3. Classes

This section describes the classes contained in the project.

## 3. VI descriptions

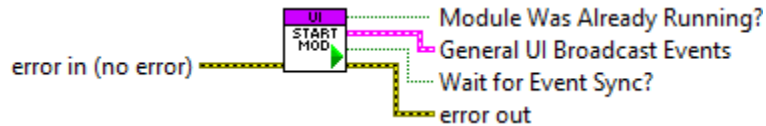
### 3.1. DQMH® modules

This section describes DQMH® modules events.

#### 3.1.1. General UI.lvlib

##### General UI.lvlib:Start Module.vi

**Event type:** Not a DQMH Event

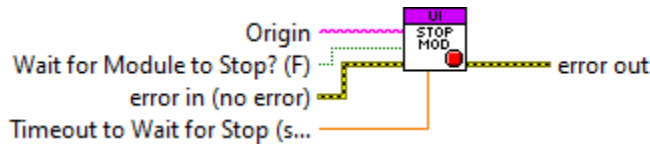


#### Description:

Launches the Module Main.vi. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

##### General UI.lvlib:Stop Module.vi

**Event type:** Not a DQMH Event



#### Description:

Send the Stop request to the Module's Main.vi. If **Wait for Module to Stop?** is TRUE, this VI will wait until the module main VI stops, and will timeout at the **Timeout to Wait for Stop** 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 **Timeout to Wait for Stop** value is ignored if 'Wait for Module to Stop?' is set to FALSE. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

##### General UI.lvlib:Show Panel.vi

**Event type:** Request

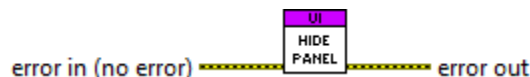


#### Description:

Send the Show Panel request to the Module's Main.vi. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

##### General UI.lvlib:Hide Panel.vi

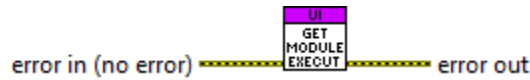
**Event type:** Request



#### Description:

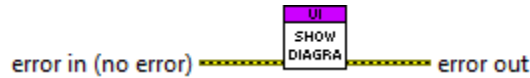
Send the Hide Panel request to the Module's Main.vi. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

## General UI.Mlib:Get Module Execution Status.vi

**Event type:** Request**Description:**

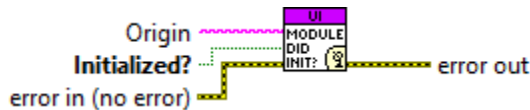
Fire the Get Module Execution Status request. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

## General UI.Mlib:Show Diagram.vi

**Event type:** Request**Description:**

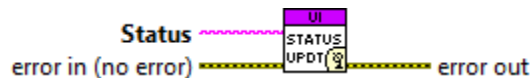
This VI tells the Module to show its block diagram to facilitate troubleshooting (add probes, breakpoints, highlight execution, etc). \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

## General UI.Mlib:Module Did Init.vi

**Event type:** Broadcast**Description:**

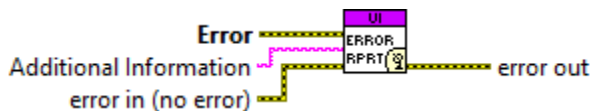
Send the Module Did Init event to any VI registered to listen to this module's broadcast events. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

## General UI.Mlib:Status Updated.vi

**Event type:** Broadcast**Description:**

Send the Status Updated event to any VI registered to listen to events from the owning module. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

## General UI.Mlib&gt;Error Reported.vi

**Event type:** Broadcast**Description:**

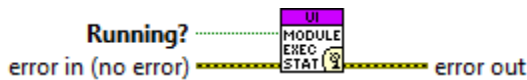
Send the Error Reported event to any VI registered to listen to events from the owning module. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

## General UI.lvlib:Module Did Stop.vi

**Event type:** Broadcast**Description:**

Send the Module Did Stop event to any VI registered to listen to this module's broadcast events. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

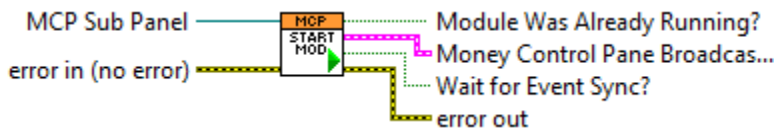
## General UI.lvlib:Update Module Execution Status.vi

**Event type:** Broadcast**Description:**

Broadcast event to specify whether or not the module is running. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

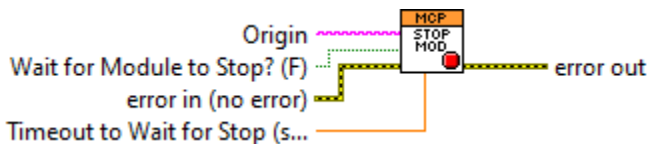
## 3.1.2. Money Control Pane.lvlib

## Money Control Pane.lvlib:Start Module.vi

**Event type:** Not a DQMH Event**Description:**

Launches the Module Main.vi. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

## Money Control Pane.lvlib:Stop Module.vi

**Event type:** Not a DQMH Event**Description:**

Send the Stop request to the Module's Main.vi. If **Wait for Module to Stop?** is TRUE, this VI will wait until the module main VI stops, and will timeout at the **Timeout to Wait for Stop** 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 **Timeout to Wait for Stop** value is ignored if 'Wait for Module to Stop?' is set to FALSE. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

## Money Control Pane.lvlib:Show Panel.vi

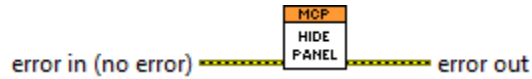
**Event type:** Request

**Description:**

Send the Show Panel request to the Module's Main.vi. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

Money Control Pane.lvlib:Hide Panel.vi

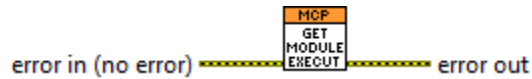
**Event type:** Request

**Description:**

Send the Hide Panel request to the Module's Main.vi. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

Money Control Pane.lvlib:Get Module Execution Status.vi

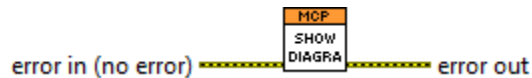
**Event type:** Request

**Description:**

Fire the Get Module Execution Status request. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

Money Control Pane.lvlib:Show Diagram.vi

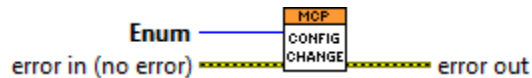
**Event type:** Request

**Description:**

This VI tells the Module to show its block diagram to facilitate troubleshooting (add probes, breakpoints, highlight execution, etc). \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

Money Control Pane.lvlib:Configuration Change.vi

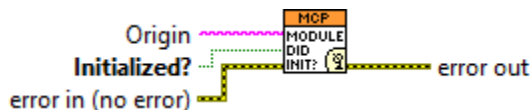
**Event type:** Request

**Description:**

Sets a configuration change don by the settings window or the VI itself. \_\_\_\_ Created using Delacor QMH Event Scriptor 5.0.0.112.

Money Control Pane.lvlib:Module Did Init.vi

**Event type:** Broadcast

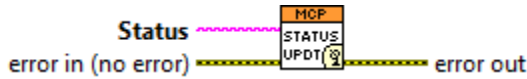
**Description:**



Send the Module Did Init event to any VI registered to listen to this module's broadcast events. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

Money Control Pane.lvlib:Status Updated.vi

**Event type:** Broadcast

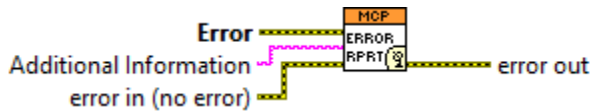


**Description:**

Send the Status Updated event to any VI registered to listen to events from the owning module. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

Money Control Pane.lvlib:Error Reported.vi

**Event type:** Broadcast



**Description:**

Send the Error Reported event to any VI registered to listen to events from the owning module. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

Money Control Pane.lvlib:Module Did Stop.vi

**Event type:** Broadcast

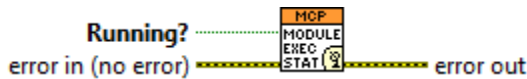


**Description:**

Send the Module Did Stop event to any VI registered to listen to this module's broadcast events. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

Money Control Pane.lvlib:Update Module Execution Status.vi

**Event type:** Broadcast



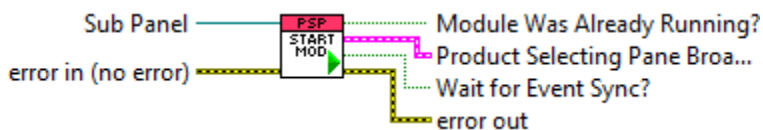
**Description:**

Broadcast event to specify whether or not the module is running. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

3.1.3. Product Selecting Pane.lvlib

Product Selecting Pane.lvlib:Start Module.vi

**Event type:** Not a DQMH Event

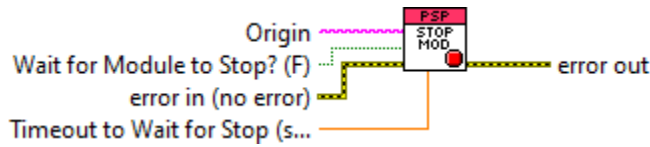


**Description:**

Launches the Module Main.vi. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

Product Selecting Pane.lmib:Stop Module.vi

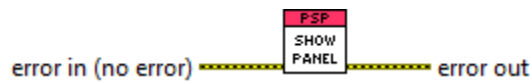
**Event type:** Not a DQMH Event

**Description:**

Send the Stop request to the Module's Main.vi. If **Wait for Module to Stop?** is TRUE, this VI will wait until the module main VI stops, and will timeout at the **Timeout to Wait for Stop** 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 **Timeout to Wait for Stop** value is ignored if 'Wait for Module to Stop?' is set to FALSE. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

Product Selecting Pane.lmib:Show Panel.vi

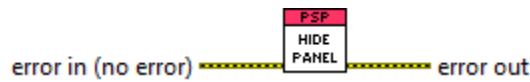
**Event type:** Request

**Description:**

Send the Show Panel request to the Module's Main.vi. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

Product Selecting Pane.lmib:Hide Panel.vi

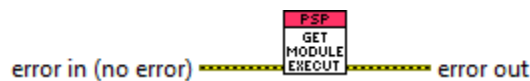
**Event type:** Request

**Description:**

Send the Hide Panel request to the Module's Main.vi. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

Product Selecting Pane.lmib:Get Module Execution Status.vi

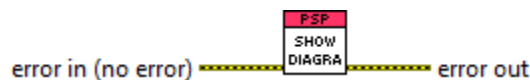
**Event type:** Request

**Description:**

Fire the Get Module Execution Status request. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

Product Selecting Pane.lmib:Show Diagram.vi

**Event type:** Request

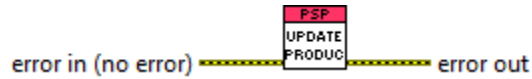


**Description:**

This VI tells the Module to show its block diagram to facilitate troubleshooting (add probes, breakpoints, highlight execution, etc). \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

Product Selecting Pane.lmlib:Update Products.vi

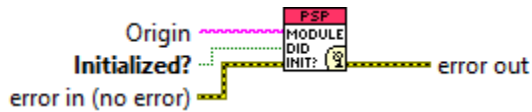
**Event type:** Request

**Description:**

Updates the products displayed in the list \_\_\_\_ Created using Delacor QMH Event Scripter 5.0.0.112.

Product Selecting Pane.lmlib:Module Did Init.vi

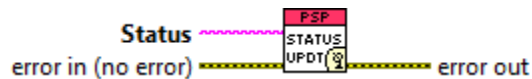
**Event type:** Broadcast

**Description:**

Send the Module Did Init event to any VI registered to listen to this module's broadcast events. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

Product Selecting Pane.lmlib:Status Updated.vi

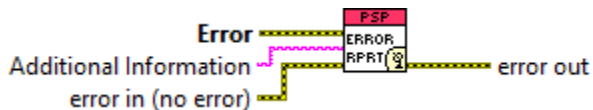
**Event type:** Broadcast

**Description:**

Send the Status Updated event to any VI registered to listen to events from the owning module. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

Product Selecting Pane.lmlib:Error Reported.vi

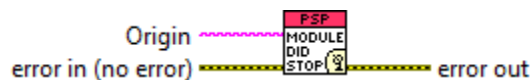
**Event type:** Broadcast

**Description:**

Send the Error Reported event to any VI registered to listen to events from the owning module. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

Product Selecting Pane.lmlib:Module Did Stop.vi

**Event type:** Broadcast

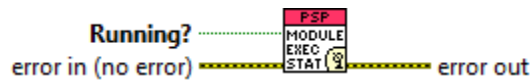


**Description:**

Send the Module Did Stop event to any VI registered to listen to this module's broadcast events. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

Product Selecting Pane.lvlib:Update Module Execution Status.vi

**Event type:** Broadcast

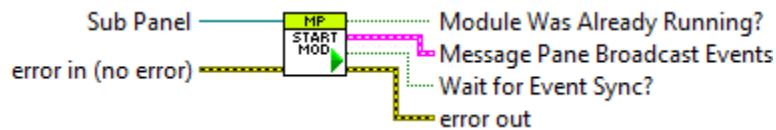
**Description:**

Broadcast event to specify whether or not the module is running. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

3.1.4. Message Pane.lvlib

Message Pane.lvlib:Start Module.vi

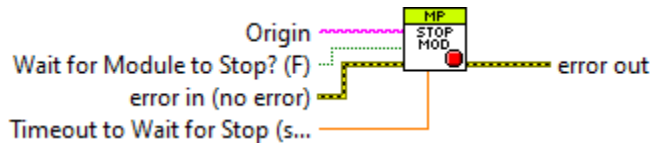
**Event type:** Not a DQMH Event

**Description:**

Launches the Module Main.vi. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

Message Pane.lvlib:Stop Module.vi

**Event type:** Not a DQMH Event

**Description:**

Send the Stop request to the Module's Main.vi. If **Wait for Module to Stop?** is TRUE, this VI will wait until the module main VI stops, and will timeout at the **Timeout to Wait for Stop** 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 **Timeout to Wait for Stop** value is ignored if 'Wait for Module to Stop?' is set to FALSE. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

Message Pane.lvlib:Show Panel.vi

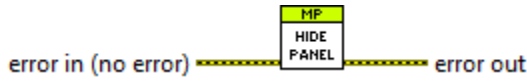
**Event type:** Request

**Description:**

Send the Show Panel request to the Module's Main.vi. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

Message Pane.lvlib:Hide Panel.vi

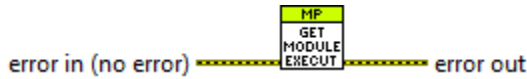
**Event type:** Request

**Description:**

Send the Hide Panel request to the Module's Main.vi. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

Message Pane.lvlib:Get Module Execution Status.vi

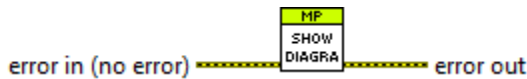
**Event type:** Request

**Description:**

Fire the Get Module Execution Status request. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

Message Pane.lvlib:Show Diagram.vi

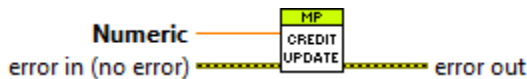
**Event type:** Request

**Description:**

This VI tells the Module to show its block diagram to facilitate troubleshooting (add probes, breakpoints, highlight execution, etc). \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

Message Pane.lvlib:Credit Update.vi

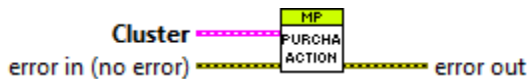
**Event type:** Request

**Description:**

A coin has been inserted, so the credit will be updated in the pane. \_\_\_\_ Created using Delacor QMH Event Scripter 5.0.0.112.

Message Pane.lvlib:Purchase action.vi

**Event type:** Request

**Description:**

A purchase request has been generated. Compares the product selected with the credit. \_\_\_\_ Created using Delacor QMH Event Scripter 5.0.0.112.

Message Pane.lvlib:Cancel.vi

**Event type:** Request

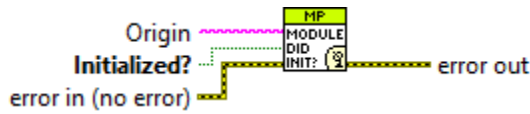


**Description:**

User request to cancel purchase. \_\_\_\_ Created using Delacor QMH Event Scripter 5.0.0.112.

Message Pane.lmlib:Module Did Init.vi

**Event type:** Broadcast

**Description:**

Send the Module Did Init event to any VI registered to listen to this module's broadcast events. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

Message Pane.lmlib:Status Updated.vi

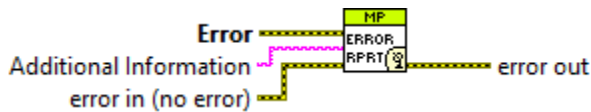
**Event type:** Broadcast

**Description:**

Send the Status Updated event to any VI registered to listen to events from the owning module. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

Message Pane.lmlib:Error Reported.vi

**Event type:** Broadcast

**Description:**

Send the Error Reported event to any VI registered to listen to events from the owning module. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

Message Pane.lmlib:Module Did Stop.vi

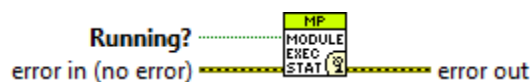
**Event type:** Broadcast

**Description:**

Send the Module Did Stop event to any VI registered to listen to this module's broadcast events. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

Message Pane.lmlib:Update Module Execution Status.vi

**Event type:** Broadcast



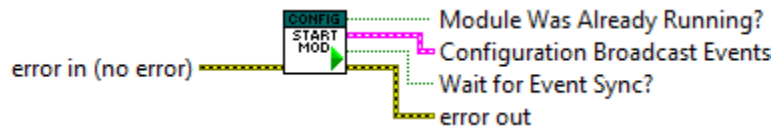
**Description:**

Broadcast event to specify whether or not the module is running. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

## 3.1.5. Configuration.Ivlib

## Configuration.Ivlib:Start Module.vi

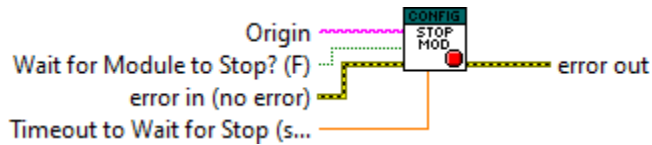
**Event type:** Not a DQMH Event

**Description:**

Launches the Module Main.vi. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

## Configuration.Ivlib:Stop Module.vi

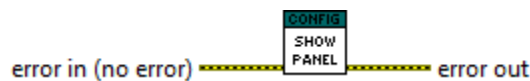
**Event type:** Not a DQMH Event

**Description:**

Send the Stop request to the Module's Main.vi. If **Wait for Module to Stop?** is TRUE, this VI will wait until the module main VI stops, and will timeout at the **Timeout to Wait for Stop** 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 **Timeout to Wait for Stop** value is ignored if 'Wait for Module to Stop?' is set to FALSE. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

## Configuration.Ivlib:Show Panel.vi

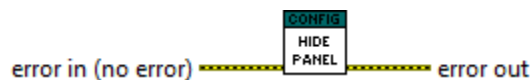
**Event type:** Request

**Description:**

Send the Show Panel request to the Module's Main.vi. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

## Configuration.Ivlib:Hide Panel.vi

**Event type:** Request

**Description:**

Send the Hide Panel request to the Module's Main.vi. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

## Configuration.Ivlib:Get Module Execution Status.vi

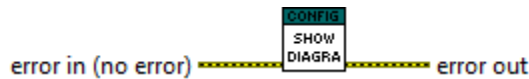
**Event type:** Request

**Description:**

Fire the Get Module Execution Status request. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

[Configuration.lvlib:Show Diagram.vi](#)

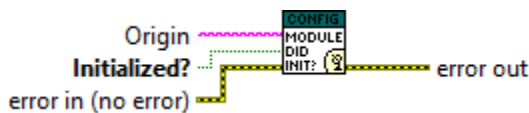
**Event type:** Request

**Description:**

This VI tells the Module to show its block diagram to facilitate troubleshooting (add probes, breakpoints, highlight execution, etc). \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

[Configuration.lvlib:Module Did Init.vi](#)

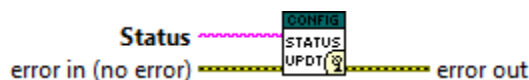
**Event type:** Broadcast

**Description:**

Send the Module Did Init event to any VI registered to listen to this module's broadcast events. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

[Configuration.lvlib:Status Updated.vi](#)

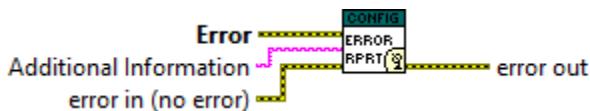
**Event type:** Broadcast

**Description:**

Send the Status Updated event to any VI registered to listen to events from the owning module. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

[Configuration.lvlib>Error Reported.vi](#)

**Event type:** Broadcast

**Description:**

Send the Error Reported event to any VI registered to listen to events from the owning module. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

[Configuration.lvlib:Module Did Stop.vi](#)

**Event type:** Broadcast

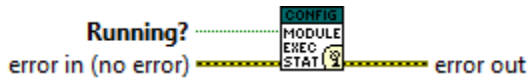


**Description:**

Send the Module Did Stop event to any VI registered to listen to this module's broadcast events. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

Configuration.lvlib:Update Module Execution Status.vi

**Event type:** Broadcast

**Description:**

Broadcast event to specify whether or not the module is running. \_\_\_\_ Based on Delacor QMH Project Template 5.0.0.82.

## 3.2. Libraries

This section describes libraries public VIs.

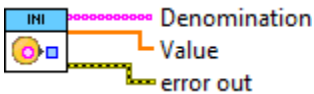
### 3.2.1. Config File Management.lvlib

Config File Management.lvlib:Change Currency.vi

**Description:**

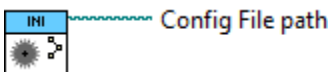
Writes the New Currency Configuration into the Config File.

Config File Management.lvlib:Create Currency Map.vi

**Description:**

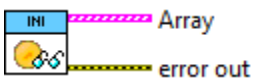
Creates a map with the currency values and strings for buttons.

Config File Management.lvlib:Create Path.vi

**Description:**

Returns the path of the config file used for the vending machine configuration

Config File Management.lvlib:Read Currency.vi

**Description:**

Reads the information in the config file and returns an array of clusters with the information for every coin.

### 3.2.2. SQLITE Inventory.lvlib

#### SQLITE Inventory.lvlib:DB Path.vi



##### Description:

Returns the Data Base File path in the application.

#### SQLITE Inventory.lvlib>Delete Product.vi



##### Description:

Deletes an element in Data Base with the given name.

#### SQLITE Inventory.lvlib:Get Inventory Data.vi



##### Description:

Returns an array of clusters with all the elements in the product table and their information.

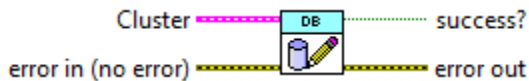
#### SQLITE Inventory.lvlib:Get Product Info.vi



##### Description:

Returns the information of a single product query.

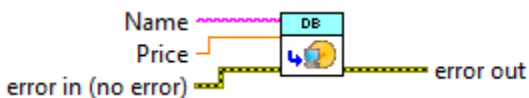
#### SQLITE Inventory.lvlib:Set New Product.vi



##### Description:

Add a new product to the DB table. If the product already exists it will not be stored and a success boolean will turn false.

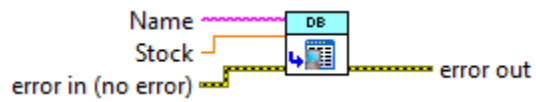
#### SQLITE Inventory.lvlib:Update Price.vi



##### Description:

Updates the price value of a single product according to the given name.

#### SQLITE Inventory.lvlib:Update Stock.vi

**Description:**

Update the value of the stock of the selected product.

### 3.3. Classes

This section describes classes public VIs.

## 4. Legal Information

### 4.1. Document creation

This document has been generated using the following tools.

#### 4.1.1. Antidoc

Project website: [Antidoc](https://wovalab.gitlab.io/open-source/labview-doc-generator/) (https://wovalab.gitlab.io/open-source/labview-doc-generator/)

Maintainer website: [Wovalab](https://wovalab.com) (https://wovalab.com)

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.

#### 4.1.2. AsciiDoc for LabVIEW™

Project website: [AsciiDoc toolkit](https://wovalab.gitlab.io/open-source/asciidoc-toolkit/) (https://wovalab.gitlab.io/open-source/asciidoc-toolkit/)

Maintainer website: [Wovalab](https://wovalab.com) (https://wovalab.com)

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.

#### 4.1.3. Graph Builder

Project website: [Graph Builder](https://gitlab.com/cgambini/graph-builder) (<https://gitlab.com/cgambini/graph-builder>)

BSD 3-Clause License

Copyright (c) 2020, Cyril GAMBINI 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.

#### 4.2. Product used in the project

The documented project has been developed with the following products.

##### 4.2.1. DQMH®

Copyright © 2015-2020 by Delacor, LLC. All Rights Reserved.

Find more details on [Delacor](https://delacor.com/products/dqmh/) (<https://delacor.com/products/dqmh/>) website

