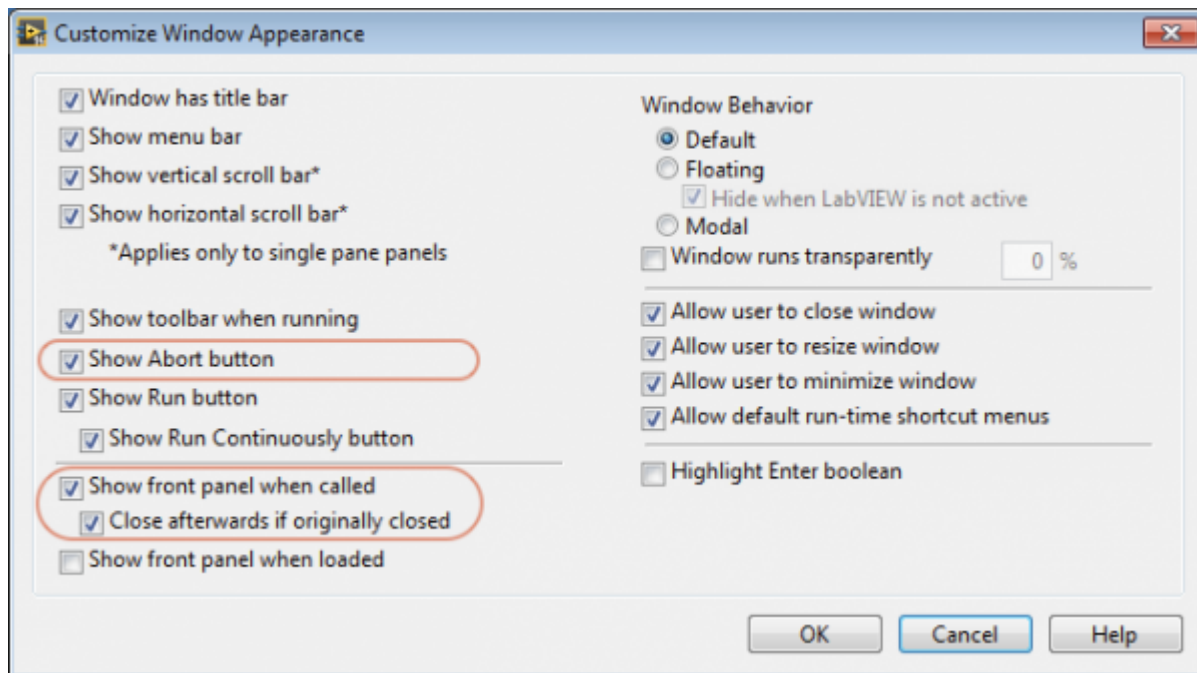


04 API Tester

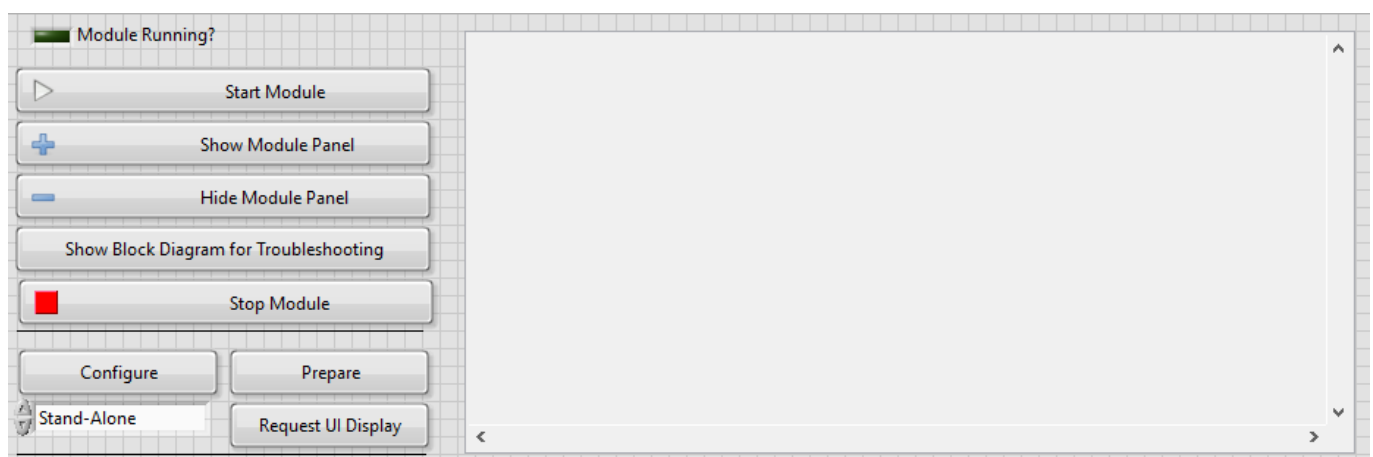
The API Tester of any given HSE DQMH module will come with the following additions to the default DQMH API tester:

Properties



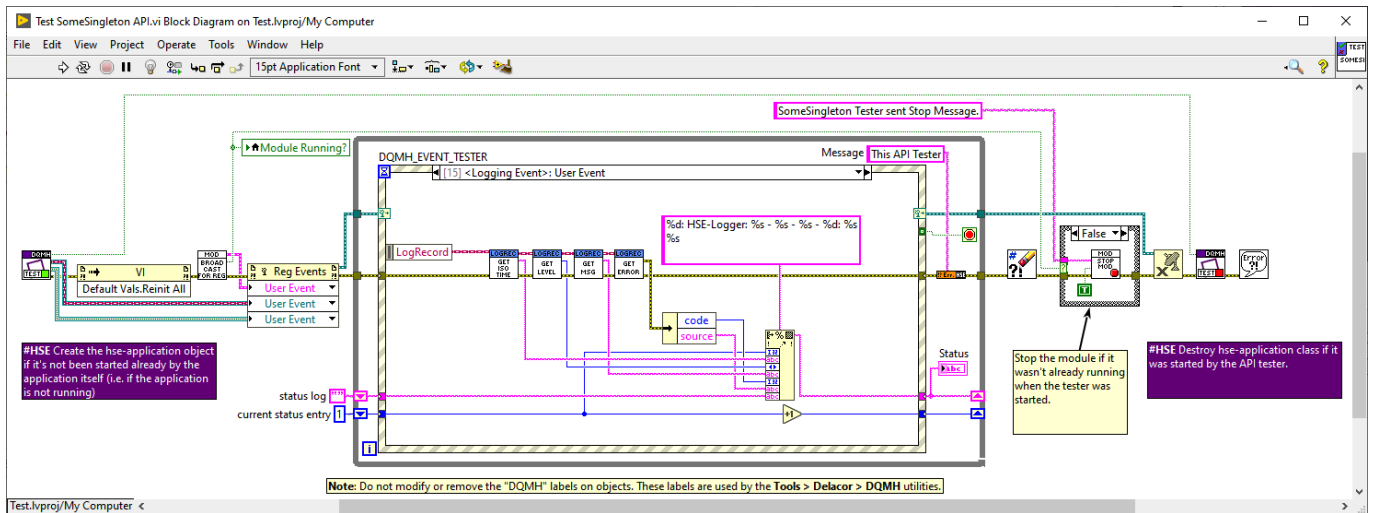
In order to be able to launch the API tester programmatically from our UI Framework, it's necessary to make the VI show its front panel when called and close it afterwards.

Front Panel

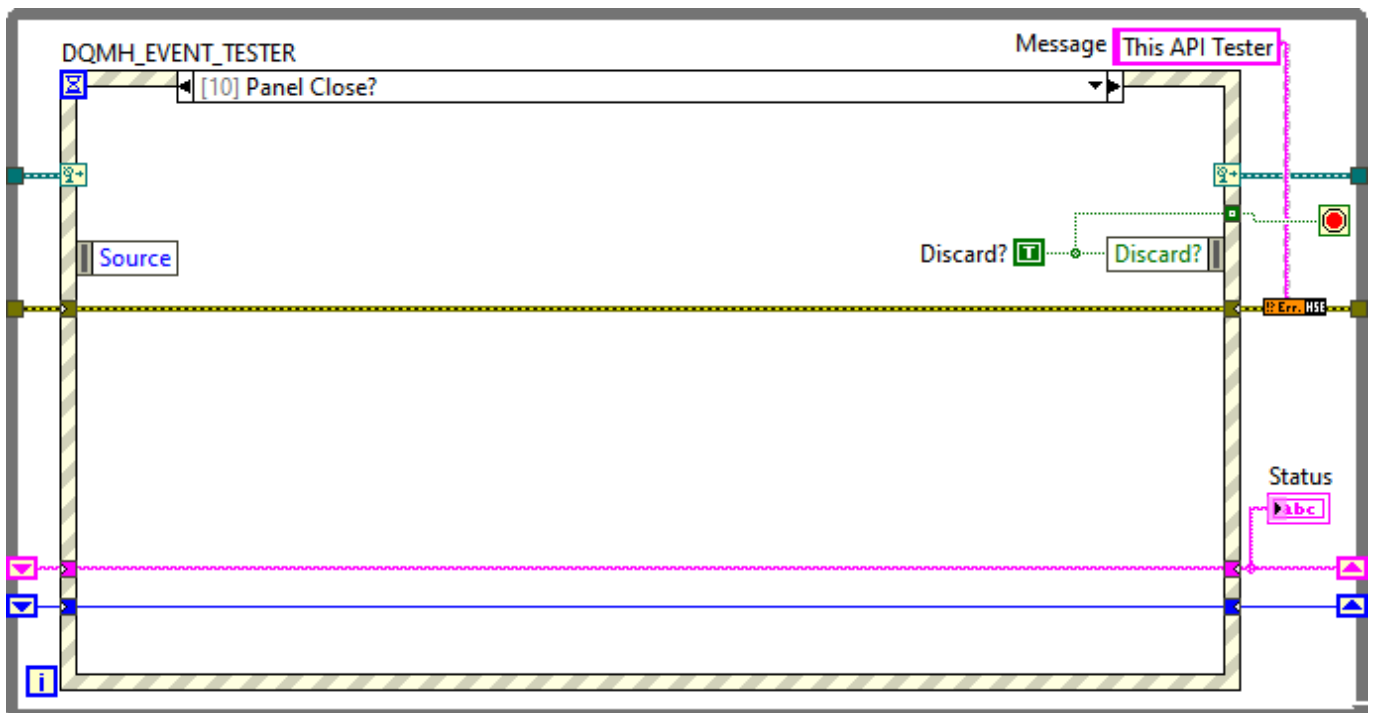


- make good use of space

Block Diagram



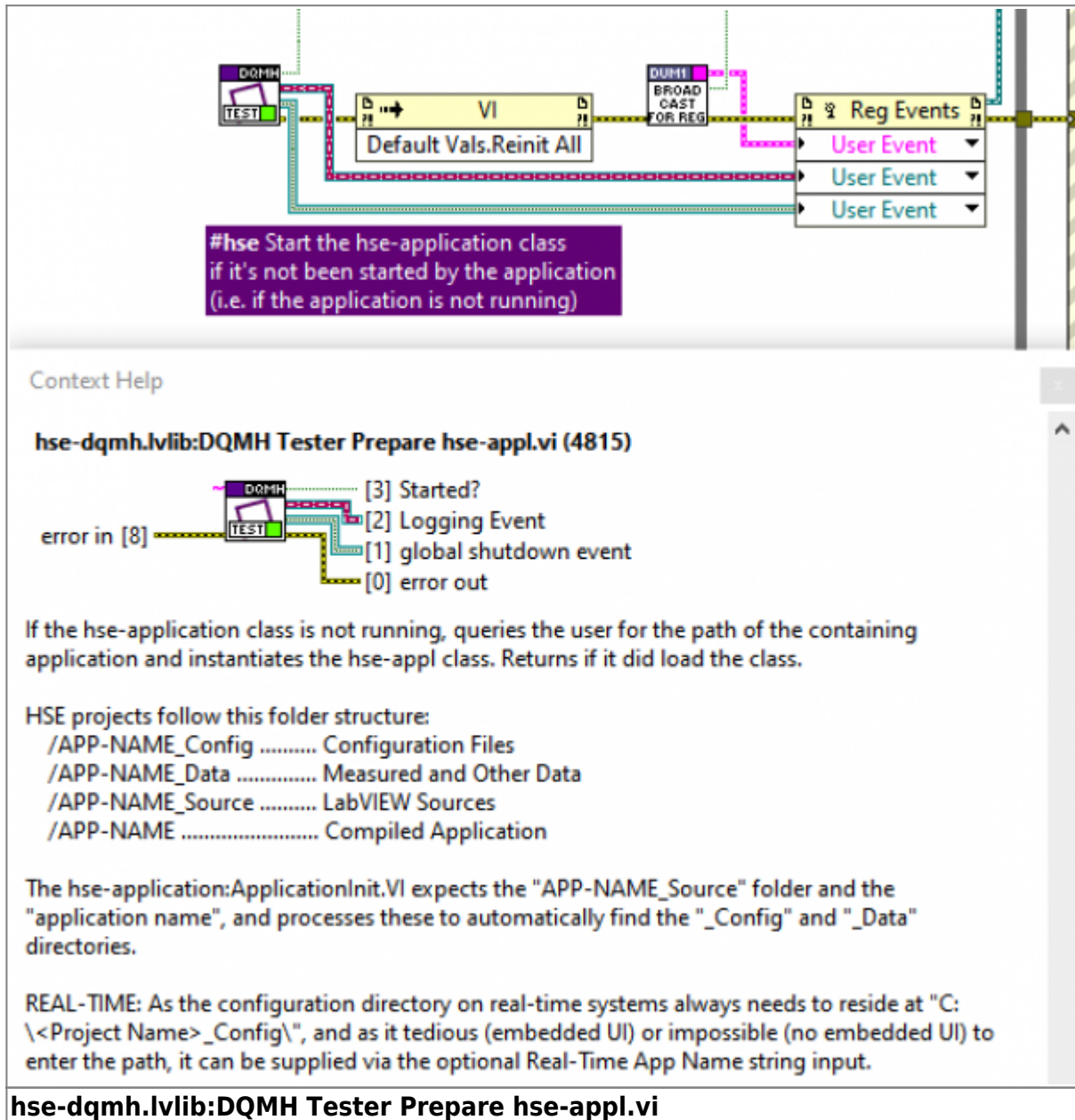
- start and cleanup hse-application class
- don't stop while loop in error case, but use a logger
- handle only errors after while loop has stopped
- have atomic cases for configure, prepare and display
- implement standard display for System Message Broadcast Event
- implement standard display for Logging Event
- implement case for global shutdown event



- do not rename the boolean constant in the "Panel Close?" case labeled "Discard?"

HSE Libraries

For the API tester to be able to start a module that depends on the various HSE Libraries helper functions (eg hse-application, hse-configuration or hse-logger), the API tester needs to initialise the HSE Libraries first.



The image displays a LabVIEW block diagram and its context help for the 'hse-dqmh.lvlib:DQMH Tester Prepare hse-appl.vi' (4815).

Block Diagram: The diagram shows a sequence of operations. It starts with a 'DQMH TEST' block, followed by a 'VI' block labeled 'Default Vals.Reinit All'. This is followed by a 'DQMH BROADCAST FOR REG' block, which then connects to a 'Reg Events' block. The 'Reg Events' block has three outputs labeled 'User Event'. A purple text box below the diagram states: "#hse Start the hse-application class if it's not been started by the application (i.e. if the application is not running)".

Context Help:

hse-dqmh.lvlib:DQMH Tester Prepare hse-appl.vi (4815)

error in [8] — [3] Started?
 [2] Logging Event
 [1] global shutdown event
 [0] error out

If the hse-application class is not running, queries the user for the path of the containing application and instantiates the hse-appl class. Returns if it did load the class.

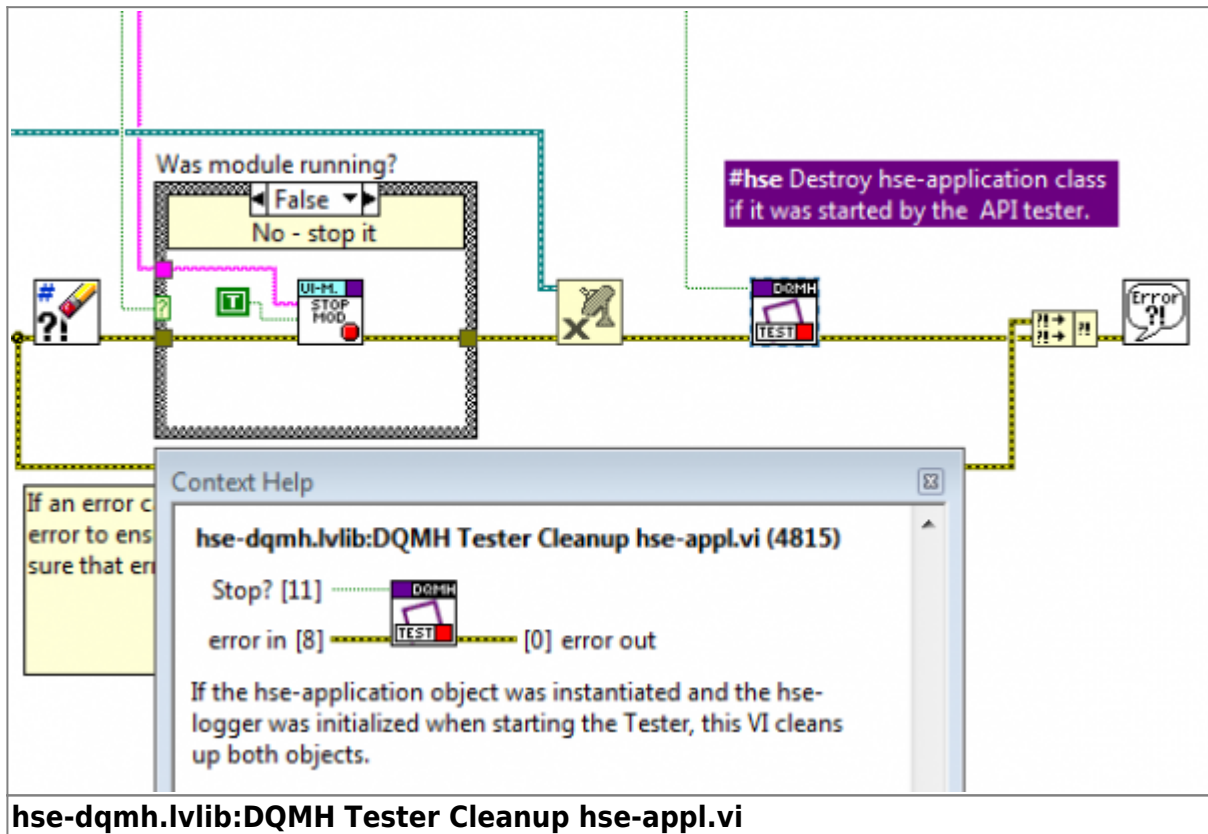
HSE projects follow this folder structure:

- /APP-NAME_Config Configuration Files
- /APP-NAME_Data Measured and Other Data
- /APP-NAME_Source LabVIEW Sources
- /APP-NAME Compiled Application

The hse-application:ApplicationInit.VI expects the "APP-NAME_Source" folder and the "application name", and processes these to automatically find the "_Config" and "_Data" directories.

REAL-TIME: As the configuration directory on real-time systems always needs to reside at "C:\<Project Name>_Config\", and as it tedious (embedded UI) or impossible (no embedded UI) to enter the path, it can be supplied via the optional Real-Time App Name string input.

hse-dqmh.lvlib:DQMH Tester Prepare hse-appl.vi



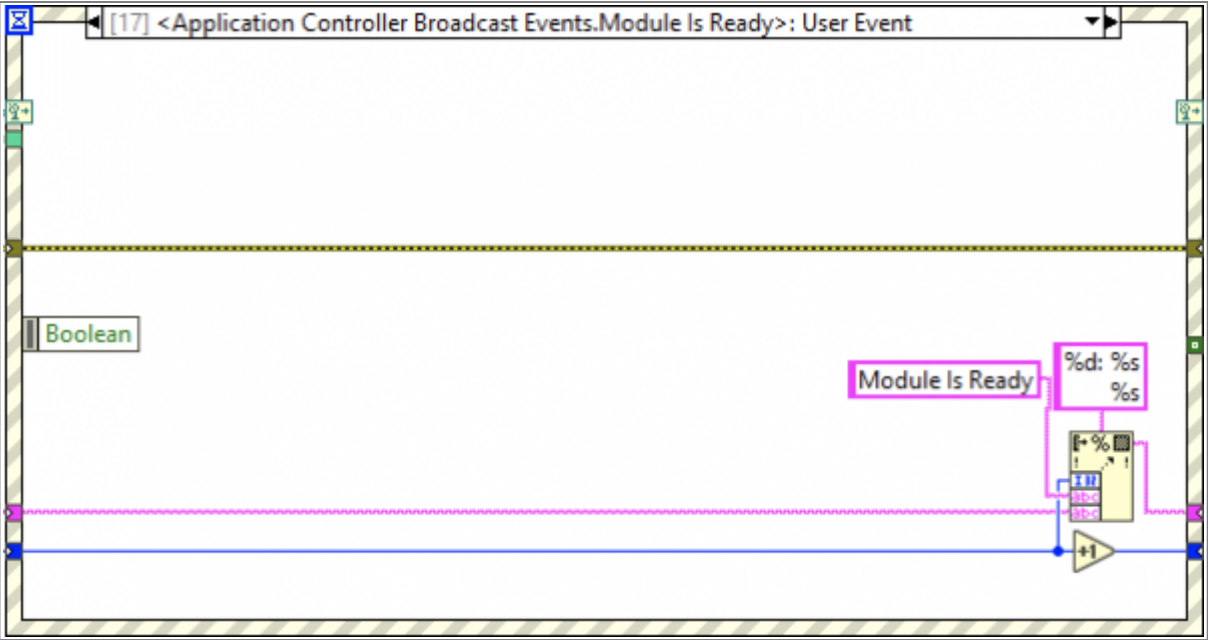
The API Tester will first check if the hse-application object is already in memory (ie the application template is running). If not, then it will try to start an instance itself. In order to do that, it needs to know where the project is stored, so it can deduce other information (app name, config folder location...).



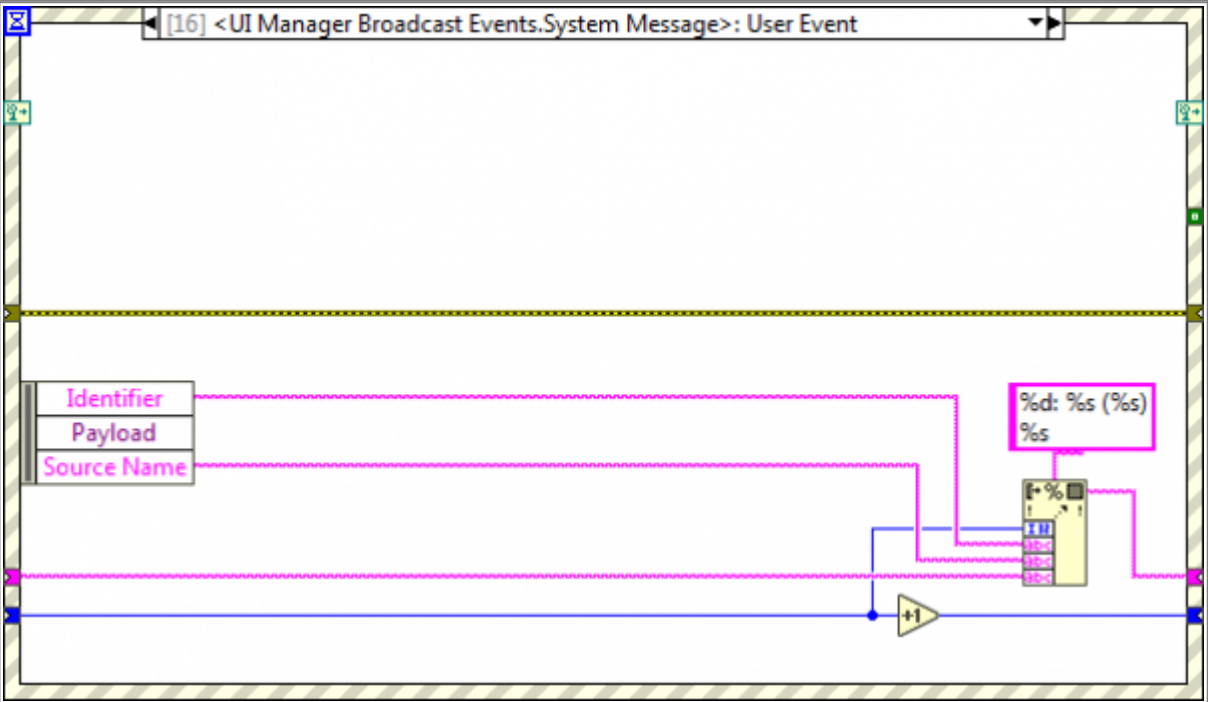
When starting the tester from the development environment, the project location is detected automatically.

When starting from the runtime engine, a dialog will pop up asking the user to “select the folder containing the “hapelsoft-libraries”-based application (i.e. the .lvproj file)”.

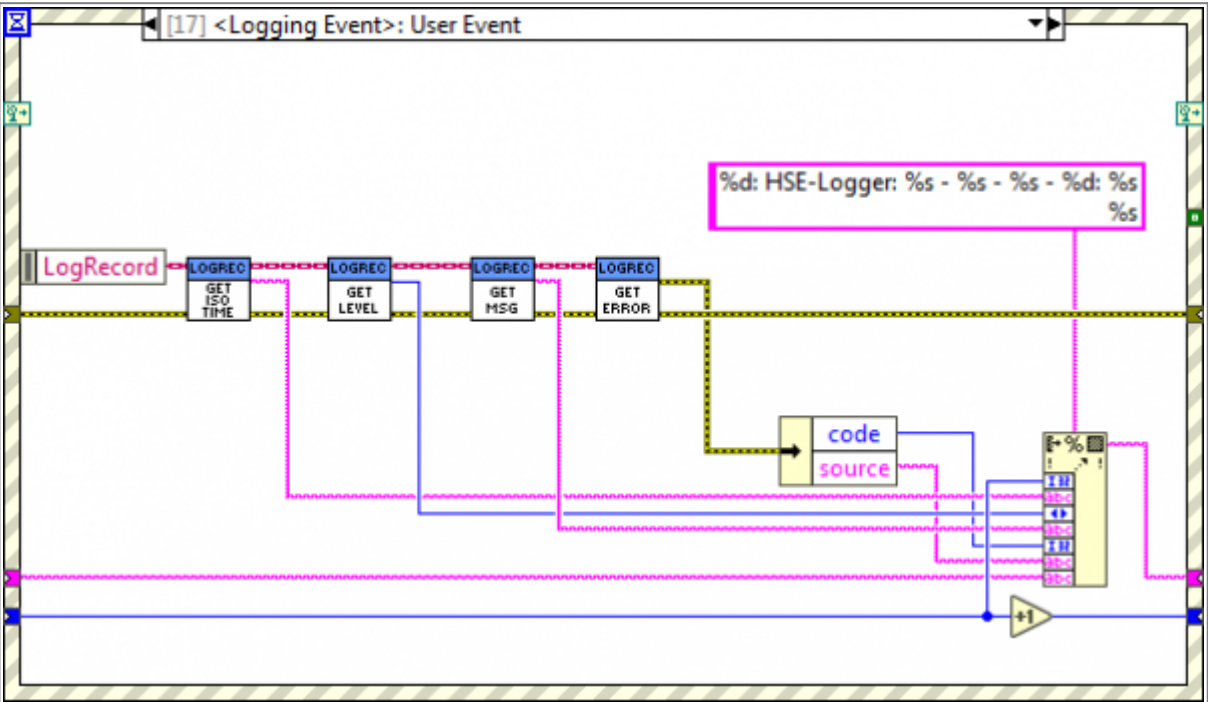
Broadcasts



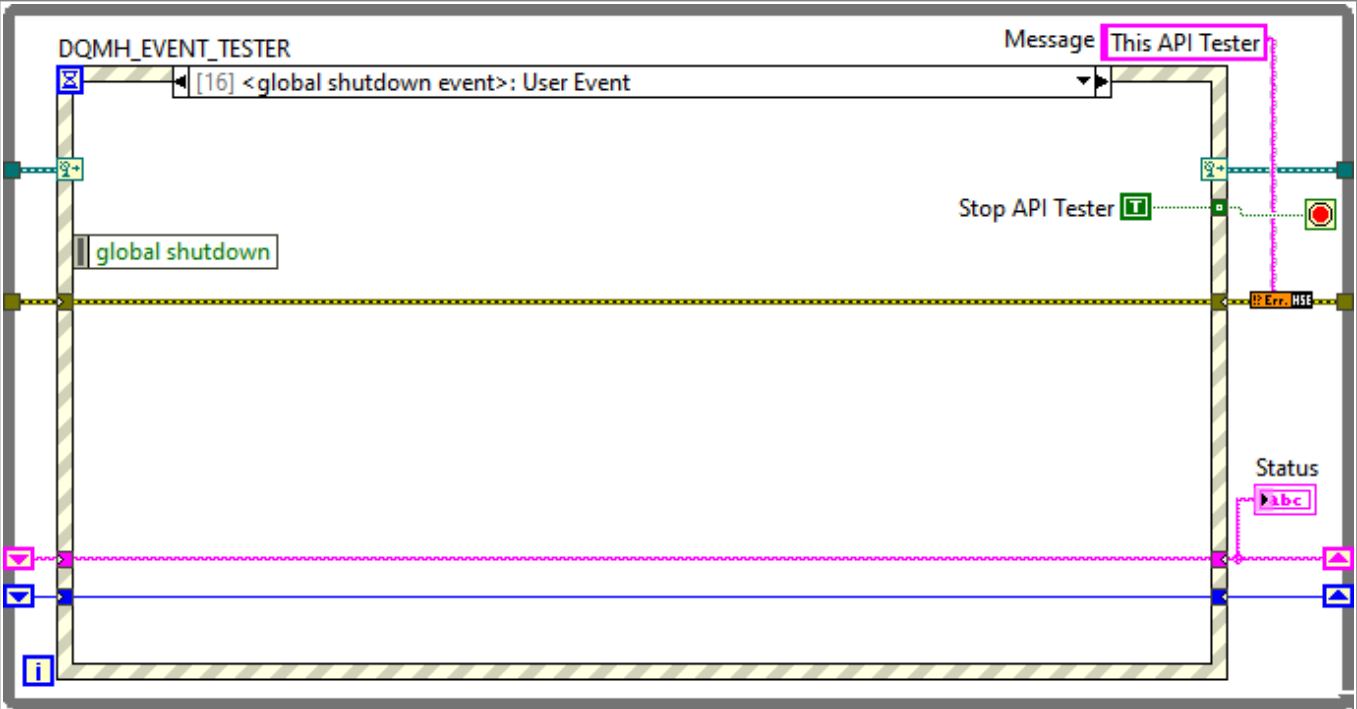
“Module Is Ready” broadcast



“System Message” broadcast



“hse-logger logging event” broadcast



“Global Shutdown Event” broadcast

Requests

- Buttons for calling the “Configure” and “Prepare” events

From:

<https://dokuwiki.hampel-soft.com/> - **HAMPEL SOFTWARE ENGINEERING**

Permanent link:

<https://dokuwiki.hampel-soft.com/kb/labview-frameworks/dqmh/hse-flavour/api-tester>

Last update: **2026/08/19 08:29**

