**QDMH Main.vit** is a template for a basic Queue Driven Message Handler. This template contains many helpful comments on the block diagram, and the QDMH API VIs have helpful VI descriptions shown in the Context Help.

Additional comments about this QDMH template and API:

* QDMH Main Cluster.vit – Defines and initializes the data for the QDMH that will be stored in the shift register
* QDMH Enqueue VI – Use this polymorphic VI to enqueue a state, multiple states, state with data, or multiple states with data. Usually, not all your states will require data, so it can be more convenient to be able to pass the state without to create a cleaner block diagram.
* QDMH Check Error VI – If an error occurs, this VI filters the errors and triggers the “Error” case in the QDMH template. You can modify this VI and the “Error” case for your error handling needs. The QDMH Main.vit template uses this VI to trap errors after the completion of the case structure.
* State data type – This QDMH uses a string data type for the state.
* “Refresh UI” case – Handle all the UI updates in this case. Enqueue this case to update the UI.
* “”, “No Action” case – When the queue is empty, this case handles front panel UI events and enqueues the corresponding state for the event.
* “Developer Error”, Default case – This case helps the developer catch typos, such as a misspelled state name, by launching a dialog to inform the developer that a non-existent case has been called.
* QDMH Exit on Run Time VI – This VI causes the application to exit when finished if the application is running as an EXE.