**Form Server.lvclass**

Form Server.lvclass is the form server actor that manages communication with form controller actors.

The form server currently can open and close multiple forms (defined by an open/close logic) convert a form to a table object (and vice versa), add form data as a row to a database, and update FK fields when multiple forms are open and a form is updated (defined by an update fields logic).

**Note:** The business logic for how the form itself operates is handled by the non-actor form, table, field, and column classes.

**Note:** The UI logic for how the form is displayed is defined by the form controller.

**Note:** An additional layer called the Form Model actor exists to help facilitate opening multiple forms from a single parent form. I originally intended this actor to help with dynamic form UIs, but chose to use tiered panels in MGI panel manager instead. In the future this layer can probably be removed.

**Accessors:**

**Message Implementations:**

**Note:** This class uses the actor framework. Do not call these method directly outside of this actor. Instead, use the send message VIs in Form Server Interfaces.lvlib/Messages for this Actor or their wrappers in Form Server Wrapper.lvclass.

**Open Form.vi**

Opens and registers a new form by launching the form model actor. Forms are registered by unique names based on their caller and panel type. If a form has already been registered, instead of opening it will be focused, displayed, and a new registration entry will be added to include the additional caller.

**Close Form.vi**

Sends a normal stop message to the actor specified by Form ID. When a form actor stops, the form server override of Handle Last Ack Core.vi calls Deregister and Close Children.vi.

**Get Table Object.vi**

Sends a reply message requesting the Form Controller to return its form object converted to a table object, and returns the table object.

**Update Form.vi**

Takes a table object as input and sends it along with a message to the form controller. The Form Controller updates the form objects data.

**Add Row to database.vi**

Sends an add row to database message to the form controller. The Form controller executes the add row method for the form/table class. The Form controller should also send an Update FKs message to update any open forms.

**Deregister FKs.vi**

Removes **Form ID** from the update fields logic object.

**Register Form FKs.vi**

Adds **Form ID** and **FK information** to the update fields logic object.

**Update FKs.vi**

Determines which forms and fields to re-populate based registered FKs in the update fields logic object and the FK filter information. The FK filter information is specified by the update fields logic class.

**Overrides (Protected):**

**Form Server Wrapper.lvclass**

The Form Server Wrapper provides wrappers for launching form actors and sending messages to the actors. These wrappers can be used in actor-framework applications.

**Start Form Server.vi**

Launches the form server as a root actor. By default, the form server is autoconfigured.

Adv. Schema Database object: Any database object that inherits from the Adv. Schema interface. The database connection must be open for the forms to function correctly.

Custom Form Server: A custom form server configuration object to overwrite the autoconfiguration.

**Stop Form Server.vi**

Sends a Normal Stop message to the form server and stops the form server and all actors launched by it.

**Open Form.vi**

Sends an open form message to the form server to open a new form in a subpanel. Use the name of the form to uniquely identify forms when multiple forms are launched by your application. By default, the form model and form controller are autoconfigured to open child forms as MGI tiered panels in the Form Window.lvclass actor.

Form: A default object of a child actor of Panel Form Controller.lvclass that contains the UI that you want to be displayed in the subpanel.

Name: A string that uniquely identifies forms launched by your application. Note: Any child forms launched by the panel form controller are not accessible by this VI.

Subpanel Refnum: The VI server reference for the subpanel that the form will be launched in.

Custom Form Model Config: A custom form model configuration object to overwrite the autoconfiguration.

**Close Form.vi**

Sends a close form message to the form server to close the form identified by Name. By default, closing the parent from will also close any child form windows, unless they are being used by other parent forms.

Name: A string that uniquely identifies forms launched by your application. Note: Any child forms launched by the panel form controller are not accessible by this VI.

**Get Table.vi**

Sends a reply message to the form server to return the contents of a parent from as a table object. You can use the Basic Table.lvclass methods to perform additional actions on the table and database.

Name: A string that uniquely identifies forms launched by your application. Note: Any child forms launched by the panel form controller are not accessible by this VI.

Table: A Basic Table.lvclass object that contains the data and configurations used by the specified form.

**Update Form.vi**

Sends an update form message that updates the contents of a parent form with the contents of the wired table object.

Name: A string that uniquely identifies forms launched by your application. Note: Any child forms launched by the panel form controller are not accessible by this VI.

Table: A Basic Table.lvclass object that contains the new data to be used in the form. The table name in the table object must match the table name registered to the form in its actor core.vi override.

**Add form to database**.vi

Sends an add form message that adds the contents of the parent from as a new row in the database. Note: The Basic Table.lvclass, which underlies the form, performs basic integrity checks on its data before executing the INSERT SQL statement.

Name: A string that uniquely identifies forms launched by your application. Note: Any child forms launched by the panel form controller are not accessible by this VI.