

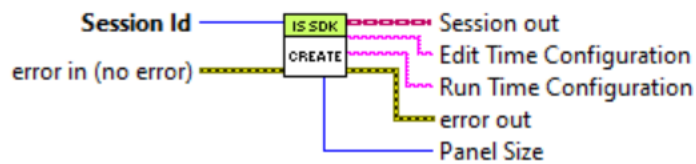
InstrumentStudio LV Plugin SDK Reference

Table of Contents

Create Session.vi	2
Inputs	2
Outputs	2
Get Edit Time Configuration.vi	2
Inputs	3
Outputs	3
Get Run Time Configuration.vi	3
Inputs	3
Outputs	3
Set Edit Time Configuration.vi	3
Inputs	3
Outputs	3
Set Run Time Configuration.vi	4
Inputs	4
Outputs	4
Register For Events.vi	4
Inputs	5
Outputs	5
Notify Project Saved Event Complete.vi	5
Inputs	6
Outputs	6
Get Project Path.vi	6
Inputs	6
Outputs	6

Create Session.vi

Create Session.vi



Call this VI with the Session Id as the first SDK call at the beginning of your plugin VI. This VI sets up the events and other data needed for the plugin panel and framework to communicate. If you need to know the panel size for your plugin panel to adjust its front panel, this VI will return the size. Some plugin panels support both large and small panel sizes and need to adjust the visuals on their front panel to better fit into that panel size. LabVIEW plugin panels are uniquely identified by the Plugin Session which is set by the framework when the plugin is run. Plugins must have a control named 'Session Id' which is a U64 numeric control on their plugin VI front panel so the framework can correctly set it. This 'Session Id' is then passed to Create Session.vi to begin the plugin session. This control can be placed on the user's VI by right-clicking on the input of the Create Session.vi and choosing 'Create Control'.

The Plugin Session returned is then used to call other SDK VIs during execution of the plugin VI to get or set properties of the plugin panel or to notify the framework of various events.

Inputs

- Session Id (required) – the Session Id from the plugin panel's front panel. This is filled in by the framework when the plugin VI is run. Each top-level plugin VI needs a 'Session Id' control on the front panel which should be wired into this input

Outputs

- Plugin Session out – the new plugin session instance to be used when calling other downstream SDK VIs
- Edit Time Configuration – the stored edit time configuration for the panel. If this is a new panel this string will be empty.
- Run Time Configuration – the stored run time configuration for the panel. If this is a new panel this string will be empty.
- Panel Size – the size for the plugin panel selected by the user. This is either 'Large' or 'Small'

Read Edit Time Configuration.vi

Read Edit Time Configuration.vi



Call this VI to read the edit-time configuration string for your plugin panel.

Call this VI to read the edit-time configuration string for your plugin panel.

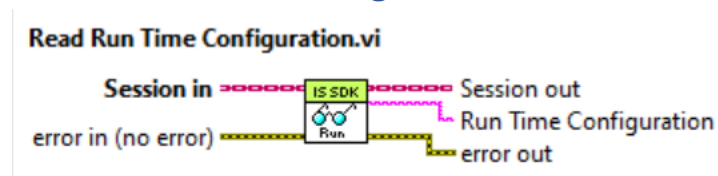
Inputs

- Plugin Session in (required) – the plugin session instance for the plugin panel.

Outputs

- Plugin Session out – the duplicated plugin session instance to be used when calling other downstream SDK VIs
- Edit Time Configuration – the current value of the edit-time configuration for the plugin panel

Read Run Time Configuration.vi



Call this VI to read the run-time configuration string for your plugin panel.

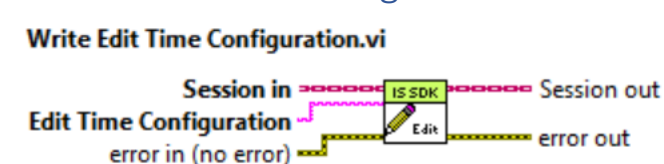
Inputs

- Plugin Session in (required) – the plugin session instance for the plugin panel.

Outputs

- Plugin Session out – the duplicated plugin session instance to be used when calling other downstream SDK VIs
- Run Time Configuration – the current value of the run-time configuration for the plugin panel

Write Edit Time Configuration.vi



Call this VI to update the edit-time configuration string for your plugin panel.

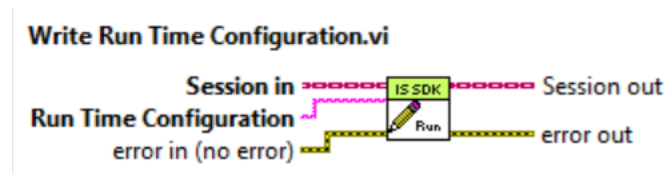
Inputs

- Plugin Session in (required) – the plugin session instance for the plugin panel.
- Edit Time Configuration – the new value for the edit-time configuration for the plugin panel

Outputs

- Plugin Session out – the duplicated plugin session instance to be used when calling other downstream SDK VIs

Write Run Time Configuration.vi



Call this VI to update the run-time configuration string for your plugin panel.

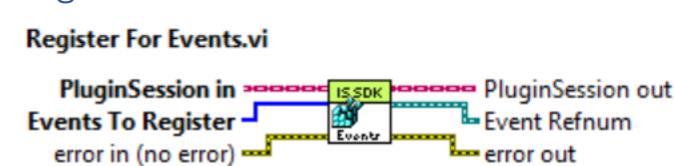
Inputs

- Plugin Session in (required) – the plugin session instance for the plugin panel.
- Run Time Configuration – the new value for the run-time configuration for the plugin panel

Outputs

- Plugin Session out – the duplicated plugin session instance to be used when calling other downstream SDK VIs

Register For Events.vi



Call this VI to let the framework know which events you want to get notifications for. Only pass in events that you are going to handle in your event structure. The following events are available.

Project Saved

This event is fired each time the project is saved and passes event data for the project path. You must call Notify Event Complete with this Completion Id to let the framework know processing is complete. Failure to do this will result in a hang.

Stop All Outputs

The Stop All Outputs event is fired when the 'Stop all outputs' button is pressed on the document toolbar. It notifies the plugin that it should cease sending output data to any external devices.

Shutdown

The Shutdown event is fired when the document containing the plugin is closed, the project containing the plugin is closed, or the plugin panel is removed from the document using the Edit Layout dialog.

Failure to stop execution of the plugin in response to this event will prevent the project from closing or a force abort of the plugin. Plugins should always register for this event.

All events except the Shutdown event pass a Completion ID as part of the event data. You must call Notify Event Complete with this Completion Id to let the framework know when processing of the event is complete. Failure to do this may result in a UI hang or freeze.

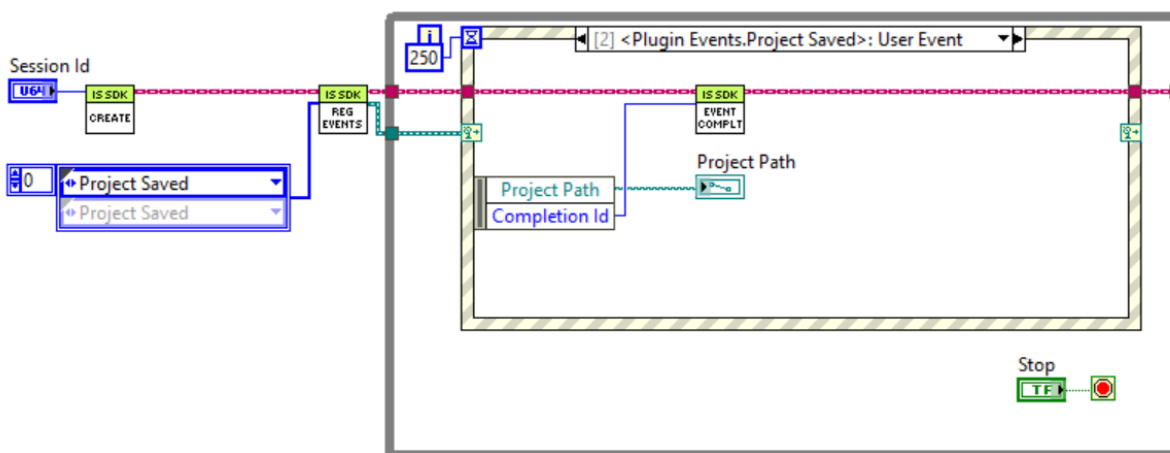
Inputs

- Plugin Session in (required) – the plugin session instance for the plugin panel.
- Events To Register (required) – an array of events which will be registered and which will fire events to the event structure. Pass one array element for each event you wish to observe.

Outputs

- Plugin Session out – the duplicated plugin session instance to be used when calling other downstream SDK VIs
- Event Refnum – an event which can be wired to the event structure’s dynamic events input terminal, adding the framework events as dynamic events within the event structure

Below is an example of registering for and handling the Project Saved event. If the ‘Register For Events.vi’ is called with the Project Saved event, it is critical that the ‘Notify Event Complete.vi’ is called in the Project Saved event, passing in the ‘Completion Id’ received in the event data so that the framework can complete the code for project save and does not hang.



Notify Event Complete.vi

Notify Event Complete.vi



Call this VI to notify the framework that processing of an event is complete. This VI should be called for each SDK event received except the Shutdown event.

Inputs

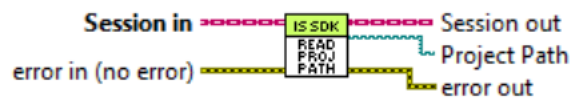
- Plugin Session in (required) – the plugin session instance for the plugin panel.
- Completion Id – the id to identify this event uniquely. It is obtained from the event data. See the example under 'Register For Events.vi'

Outputs

- Plugin Session out – the duplicated plugin session instance to be used when calling other downstream SDK VIs

Read Project Path.vi

Read Project Path.vi



Call this VI to retrieve the absolute path of the InstrumentStudio project file on disk.

Inputs

- Plugin Session in (required) – the plugin session instance for the plugin panel.

Outputs

- Plugin Session out – the duplicated plugin session instance to be used when calling other downstream SDK VIs
- Project Path – the absolute path to the '.instudioproj' file of the current project on disk. Returns an empty string if the project has not yet been saved.

Update Running State.vi

Update Running State.vi



Call this VI to notify the framework when the plugin panel is running or stops running.

Inputs

- Plugin Session in (required) – the plugin session instance for the plugin panel
- Is Running – pass true if the plugin panel is currently running or false if it is stopped

Outputs

- Plugin Session out – the duplicated plugin session instance to be used when calling other downstream SDK VIs

Close Session.vi

Close Session.vi



Call this VI to close out all references created for the session and unregister the plugin session with the framework. It should be called at the end of your plugin VI after the event loop is completed as the last call in the VI. After calling this VI, the plugin session is no longer valid and should not be used to call any other SDK VIs. This VI executes regardless of the value of error in.

Inputs

- Plugin Session in (required) – the plugin session instance for the plugin panel which is being close.