



Project Provider Interface VIs

Global_Interface

Defines global-level events that are not tied into a specific object type.

Global_Init

Description: Called every time the provider is loaded. This happens when LabVIEW.exe runs and whenever a project explorer window is created (open, new, etc.). It should be used to construct menus and do any other initialization tasks. If licensing is not integrated, the check should be performed here (in addition to any licensing-based initialization).

Inputs:

`ItemRef` *Object* Reference to the global item

Outputs: *None*

Global_OnCommand

Description: Called when executing global menu and toolbar commands. Should be used to define or call button/menu behaviors.

Inputs:

`ItemRef` *Item* Reference to the global item

`String` *CommandID* Menu tag of the command that was selected (defined when registered)

Outputs: *None*

Global_OnUpdateCommand

Description: Called during framework update operations. Building menus calls this for each customized menu item. When in the project window, this is called in a loop to make sure all the menus are in the most recent state.

Inputs:

`ItemRef` *Item* Reference to the global item

`mxLvMenuItem` *Menu Item In* The menu item being updated

Outputs: *None*

Global_Exit

Description: Not currently called.

Inputs:

`ItemRef` *Item* Reference to the global item

Outputs: *None*

CreateNewWizard_Interface

Defines events that occur while adding items to the project window.

CreateNewWizard_Finalize

Description: Called after the new item has been created.

Inputs:

[ItemRef](#) *Wizard* Reference to the wizard

[ItemRef](#) *Item* Reference to the newly created item

Outputs: *None*

CreateNewWizard_GetCreateNewWeight

Description: Called when the appropriate *New* menu is being constructed.

Inputs:

[ItemRef](#) *Wizard* Reference to the wizard

Outputs:

[I32](#) *Weight* Controls the relative position of the entry in the *New* menu

CreateNewWizard_GetNewItemInfo

Description: Called whenever right-clicked on an item that supports this *New* menu.

Inputs:

[ItemRef](#) *Wizard* Reference to the wizard

Outputs:

[CreateNewInfo](#) *NewInfo* Unknown

CreateNewWizard_IncludeItem

Description: Called when the appropriate *New* menu is being constructed.

Inputs:

[ItemRef](#) *Wizard* Reference to the wizard

Outputs:

[Boolean](#) *IncludeItem* Controls inclusion of the item in the *New* menu based on some logic

CreateNewWizard_Init

Description: Called to initialize the wizard.

Inputs:

[ItemRef](#) *Wizard* Reference to the wizard

Outputs:

[Boolean](#) *Enable Next* Controls whether the `CreateNewWizard_Invoke` is called or not on creating a new item.

[Boolean](#) *Enable Finish* Unknown

[String](#) *Label* The text that shows up in the *New* pop-up menu

CreateNewWizard_Invoke

Description: Called when an entry added to a *New* menu by this provider is selected.

Inputs:

ItemRef *Wizard* Reference to the wizard

mxLvNIIM *New Item Parent* Item Moniker representing the parent of the item to be created

Outputs:

Boolean *Canceled* Controls whether the framework should create the new item (false if done through VI Server)

Item_Interface

Defines events that occur to individual items in the project explorer window's item tree.

Item_CanDelete

Description: Determines if the project item can be deleted

Inputs:

[ItemRef](#) *Object Reference to the project item*

Outputs:

[Boolean](#) *CanDelete* Controls whether the item can be deleted (true) or not (false)

Item_CanDoHelp

Description: Determines if the project item supports help.

Inputs:

[ItemRef](#) *Object Reference to the project item*

Outputs:

[Boolean](#) *CanDoHelp* Controls if help is supported (true) or not (false)

Item_CanDoProperties

Description: Determines if the project item supports properties.

Inputs:

[ItemRef](#) *Object Reference to the project item*

Outputs:

[Boolean](#) *CanDoProperties* Controls whether properties are supported (true) or not (false)

Item_CanDragToExternalWindow

Description: Called when the project item is dragged to an external window.

Inputs:

[ItemRef](#) *Object Reference to the project item*

Outputs:

[Boolean](#) *CanDrag* Controls whether the item can be dragged to an external window.

Item_CanDragToProjectWindow

Description: Called when an item is dragged into the project window.

Inputs:

[ItemRef](#) *Object Reference to the project item*

[ItemRef](#) *DropTarget* Reference to the item being dropped on

Outputs:

[Boolean](#) *CanDrag* Controls whether DropTarget can be dragged to project window

Item_CanDropExternalData

Description: Called when an item from an external window is dragged on to this item.

Inputs:

[ItemRef](#) *Object* Reference to the project item

Outputs:

[Boolean](#) *CanDrop* Controls if the project item is available for dropping an external item. (Setting it to true does not drop the external item on the project item though)

Item_CanDropItem

Description: Called when another project item is dragged to the item.

Inputs:

[ItemRef](#) *Object* Reference to the project item

[ItemRef](#) *DropItem* Reference to item being dragged

Outputs:

[Boolean](#) *CanDrop* Controls whether DropItem can be dropped on Object

Item_CanRename

Description: Determines if the project item can be renamed.

Inputs:

[ItemRef](#) *Object* Reference to the project item

Outputs:

[Boolean](#) *CanRename* Controls whether rename is supported (true) or not (false)

Item_Exit

Description: Not currently called.

Inputs:

[ItemRef](#) *Object* Reference to the project item

Outputs: *None*

Item_GetAddCategories

Description: TBD

Inputs: *None*

Outputs: *None*

Item_GetCreateNewCategories

Description: TBD

Inputs: *None*

Outputs: *None*

Item_Init

Description: Called when initializing a single item in the project.

Inputs:

[ItemRef](#) *Object* Reference to the project item

Outputs: *None*

Item_NotifyChanged

Description: Called when the item is modified (item created or modified, file created or modified, item added or renamed, item added to or modified for SCC).

Inputs:

ItemRef *Object Reference to the project item*

mxLvChangeType *Change Type Type of change*

Outputs: *None*

Item_OnCommand

Description: Called when a menu entry installed by the provider is selected on a single item.

Inputs:

ItemRef *Object Reference to the project item*

String *CommandID Menu tag defined when command was registered*

Outputs: *None*

Item_OnDbClick

Description: Called when the user double-clicks on an item in the project tree.

Inputs:

ItemRef *Object Reference to the project item*

Outputs: *None*

Item_OnDelete

Description: Called when the project item is deleted.

Inputs:

ItemRef *Object Reference to the project item*

Boolean *Silently Controls whether the providers delete method should display any UI (false) or not (true)*

Outputs: *None*

Item_OnDoHelp

Description: Called when *Help...* is selected from the popup menu for the project item.

Inputs:

ItemRef *Object Reference to the project item*

Outputs: *None*

Item_OnDoProperties

Description: Called when *Properties* is selected from the popup menu for an item.

Inputs:

ItemRef *Object Reference to the project item*

Outputs: *None*

Item_OnDropFiles

Description: Called when one or more files are dragged from the explorer and dropped on a project item.

Inputs:

[ItemRef](#) *Object* Reference to the project item

[Path\[\]](#) *DropFilesList* List of files

Outputs: *None*

Item_OnDropItem

Description: Called when another project item is dropped on the item.

Inputs:

[ItemRef](#) *Object* Reference to the project item

[ItemRef](#) *DroppedItem* Reference to item being dropped

Outputs: *None*

Item_OnPopupMenu

Description: Called right before the project item's popup menu is displayed—menu can be modified before display.

Inputs:

[ItemRef](#) *Object* Reference to the project item

Outputs: *None*

Item_OnRename

Description: Called when the item is renamed.

Inputs:

[ItemRef](#) *Object* Reference to the project item

[String](#) *New Name* New name of the item.

Outputs: *None*

Item_OnSelect

Description: Called when the project item is selected in the project tree.

Inputs:

[ItemRef](#) *Object* Reference to the project item

Outputs: *None*

Item_OnUnselect

Description: Called when the project item is unselected in the project tree.

Inputs:

[ItemRef](#) *Object* Reference to the project item

Outputs: *None*

Item_OnUpdateCommand

Description: Called to update the specified command right before it is about to be displayed in a menu or popup.

Inputs:

ItemRef *Object Reference to the project item*

String *Command ID* Menu tag of the command being updated (defined when registered)

Outputs:

mxLvMenuItem *Menu Item Out*

Item_OnWizardComplete

Description: Notifies the project item that the Create New Wizard process is finished.

Inputs:

ItemRef *Object Reference to the project item*

Outputs: *None*

Item_PrefersFPHeap

Description: Determines if item wants the VI Front Panel or Block Diagram reference when WasDroppedOnVI is called.

Inputs:

ItemRef *Object Reference to the project item*

Outputs:

Boolean *Prefers FP* True if Front Panel reference, false if Block Diagram reference

Item_ValidateRename

Description: Determines if the new name is valid for the project item.

Inputs:

ItemRef *Object Reference to the project item*

String *New Name* New name of item

Outputs:

Boolean *Can Rename* Controls whether the new name is valid (true) or not (false)

Item_WasDroppedOnItem

Description: Called when the project item was dropped on another project item in the tree.

Inputs:

ItemRef *Object Reference to the project item*

ItemRef *Drop Item* The item that the project item was dropped on

Outputs: *None*

Item_WasDroppedOnVI

Description: Called when the project item is dropped on an open VI (either front panel or block diagram).

Inputs:

ItemRef *Object* Reference to the project item
I32 X X-coordinate of drop
I32 Y Y-coordinate of drop
VI Refnum *VI* Front panel or Block Diagram reference (determined by PrefersFPHeap method)
Boolean *Front Panel?* Specifies whether item has been dropped on Front Panel (true) or not (false)
I32 Flags **NOTE: REFER TO `_lvProjectDragDropFlags` IN `LvProject.h`**

Outputs:

Boolean *Can Drop* Controls whether object can be dropped or not
lvProjectDropresults *Result* **NOTE: REFER TO `lvProjectDropResults` IN `LvProject.h`**

Provider_Interface

Defines events that occur to multiple items on the project explorer window.

Provider_InitItems

Description: Called when multiple items in the project are bound.

Inputs:

`ItemRef[]` *Items Array* Array of references to project items

Outputs: *None*

Provider_LoadComplete

Description: Called when the project has finished loading.

Inputs:

`ItemRef` *Project Reference* Reference to the project

Outputs: *None*

Provider_LoadCompleteWithWarnings

Description: Same as LoadComplete except save warnings are optionally returned.

Inputs:

`ItemRef` *Project Reference* Reference to the project

Outputs:

`String[]` *warnSummaryStrings* Warning summary messages

`String[]` *warnStrings* Warning messages

Provider_NotifyChanged

Description: Called when changes occur involving multiple items (create, add, etc.)

Inputs:

`ItemRef[]` *item IDs*

`mxLvChangeType` *Change Type*

`Error Cluster` *error in Input* error cluster

Outputs:

`Error Cluster` *error out* Output error cluster

Provider_OnCommand

Description: Called when a command is invoked on multiple items (menu or toolbar).

Inputs:

`ItemRef[]` *Items Array*

`String` *Command ID* Menu tag defined when command was registered

Outputs: *None*

Provider_OnPopupMenu

Description: Called right before the project items' popup menu is displayed—can modify menu.

Inputs:

`ItemRef[]` *objects* References to the selected items

Outputs: *None*

Provider_OnSaveForPrevious

Description: Called when a project is being saved to a previous version—gives provider a chance to modify property bag to match the previous version.

Inputs:

`U32 Version` Version number that is being saved for
`ItemRef[] ObjectsIn` References to items being saved

Outputs: *None*

Provider_OnSaveForPreviousEx

Description: Same as `OnSaveForPrevious` except the provider is also passed to the method.

Inputs:

`U32 Version` Version number that is being saved for
`ItemRef[] ObjectsIn` References to items being saved
`ItemRef Provider Item` Reference to this provider

Outputs: *None*

Provider_OnSaveForPreviousWithWarnings

Description: Same as `OnSaveForPreviousEx` except save warnings are optionally returned.

Inputs:

`U32 Version` Version number being saved for
`ItemRef[] ObjectsIn` References to items being saved
`ItemRef Provider Item` Reference to this provider

Outputs:

`String[] Warning Summaries` Warning summary messages
`String[] Warnings` Warning messages

Provider_OnSaveProject

Description: Called when the project is being saved.

Inputs:

`ItemRef Project` Reference to the project

Outputs: *None*

Provider_OnUpdateCommandBegin

Description: Called before `UpdateCommand` is called for each command in the toolbar or menu.

Inputs:

`ItemRef Project` Reference to the project

Outputs: *None*

Provider_OnUpdateCommandEnd

Description: Called after `UpdateCommand` is called for each command in the toolbar or menu.

Inputs:

`ItemRef Project` Reference to the project

Outputs: *None*

Provider_Shutdown

Description: Called when the project shuts down.

Inputs:

ItemRef *Project* Reference to the project

Boolean *Is LV Exiting?* Specifies whether LabVIEW is exiting (true) or not (false)

Outputs: *None*

Provider_Startup

Description: Called when the project is created.

Inputs:

ItemRef *Project* Reference to the project

Outputs: *None*