



# Provider API VIs

---

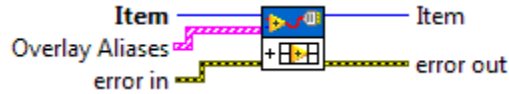
1) **mxLvGetAssociatedItems.vi**

Returns the items that belong to this project and are associated with one of the file paths.



2) **mxLvAddIconOverlays.vi**

Registers a set of Icon Overlay Aliases and tells the framework where to find the .ico files matching these Aliases.



3) **mxLvDeleteAllProperties.vi**

Removes all properties from the property bag for the given Item.



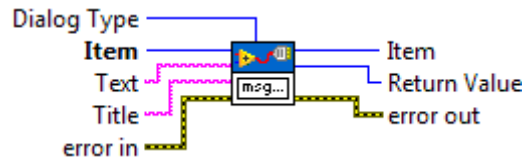
4) **mxLvDeleteProperty.vi**

Removes a property from the property bag for the given Item.



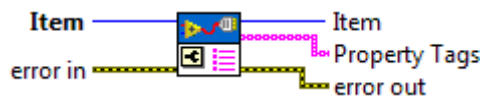
5) **mxLvDisplayMessage.vi**

Displays a message box for the Provider.



6) **mxLvEnumProperties.vi**

Enumerates all the properties of the specified item.



7) **mxLvErrorHandler.vi**

Converts an error or warning code to an error cluster. This VI is useful when you receive a return value from a shared library call or when you return user-defined error codes



8) **mxLvFilesExist.vi**

Checks whether the given files exist or not.



9) **mxLvGenerateGuid.vi**

Generates a GUID (a unique identifier).



10) **mxLvGetAllDescendentsSFP.vi**

Gets all descendents for the given item. It should return the same results as the ProjectItem:Get All Descendents method with the default values.

NOTE: This VI only works during the Provider\_OnSaveForPreviousEx call.



11) **mxLvGetBinaryProperty.vi**

Retrieves a binary property from the property bag for the given Item.



12) **mxLvGetBooleanProperty.vi**

Retrieves a Boolean property from the property bag for the given Item.



**13) mxLvGetFilePath.vi**

Gets the file path for the current Item.



**14) mxLvGetFilePathBatch.vi**

Gets the file paths for the specified Items.



**15) mxLvGetIntProperty.vi**

Retrieves an Integer property from the property bag for the given Item.



**16) mxLvGetIntPropertyBatch.vi**

Retrieves the values for the Integer property from the property bag for the given Items.



**17) mxLvGetItem.vi**

Given a NIIM, gets the associated Item. NIIM is a cluster of Item URL, Item ID, Item Index and the Specific Type. Refer mxLvNIIM.cti in the <LabVIEW>/resource/Framework/Providers/API directory.



**18) mxLvGetItemBatch.vi**

Given a list of NIIMs, gets the associated Items.



**19) mxLvGetItemFromItemIDSFP.vi**

Returns the Item for the specified Item ID. This VI only works during the Provider\_OnSaveForPreviousEx call.



**20) mxLvGetItemIDFromItemSFP.vi**

Returns the Item ID for an Item. This VI only works during the Provider\_OnSaveForPreviousEx call.



**21) mxLvGetItemRef.vi**

Given an item, gets the associated item refnum.



**22) mxLvGetNIIM.vi**

Given an item, gets the associated NIIM.



**23) mxLvGetNIIMBatch.vi**

Given a list of items, gets the associated NIIMs.



**24) mxLvGetPathProperty.vi**

Retrieves a path property from the property bag for the given Item.



**25) mxLvGetProjectPath.vi**

Gets the path to the project that the item belongs to.



**26) mxLvGetProjectRef.vi**

Gets a reference to the 'current' Project so the Provider can call VI Server methods on the Project.



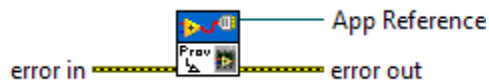
**27) mxLvGetPropertyType.vi**

Retrieves the type of the property for the given Item.



**28) mxLvGetProviderAppRef.vi**

Gets a reference to the current application so the provider can call VI Server methods on the application.



**29) mxLvGetRefProperty.vi**

Retrieves a reference property from the property bag for the given Item.



**30) mxLvGetStringProperty.vi**

Retrieves a String property from the property bag for the given Item.



**31) mxLvGetTarget.vi**

Returns the target for the given item.



**32) mxLvGetURLSFP.vi**

Returns the project URL for the specified item.

NOTE: This VI only works during the Provider\_OnSaveForPreviousEx call.



**33) mxLvGetVariantProperty.vi**

Retrieves a property from the property bag for the given Item as variant data.



**34) mxLvGetVIREfProperty.vi**

Gets an Integer property from the property bag for the given Item and casts it to a VI Refnum.



**35) mxLvHasProperty.vi**

Returns whether the given Item has the specified property in its property bag.



**36) mxLvHasPropertyBatch.vi**

Returns whether the given Items have the specified property in its property bag.



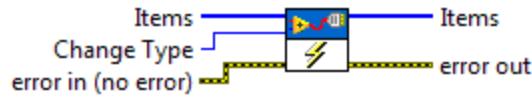
**37) mxLvIsPersistent.vi**

Returns whether the given property of the item is saved with the project or not.



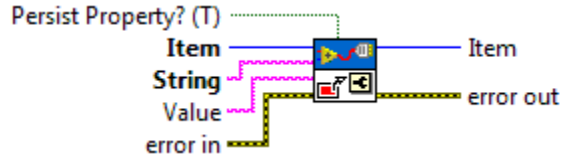
**38) mxLvNotifyChanged.vi**

Notifies the providers for the specified items of the change. Calls the NotifyChanged VI of the item interface of the associated providers if “Items” has only one item and of the provider interface if “Items” has more than one item.



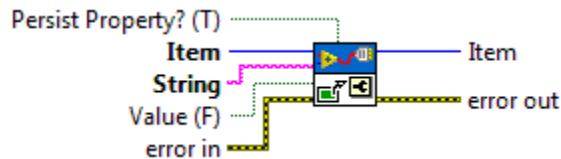
**39) mxLvSetBinaryProperty.vi**

Writes a binary property to the property bag for the given Item.



**40) mxLvSetBooleanProperty.vi**

Writes a Boolean property to the property bag for the given Item.



**41) mxLvSetCursorBusy.vi**

Sets the cursor to busy state.



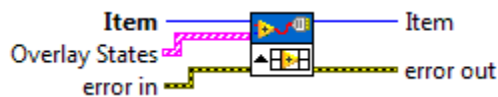
**42) mxLvSetIcon.vi**

Sets the icon to be displayed next to the item in the tree.



**43) mxLvSetIconOverlays.vi**

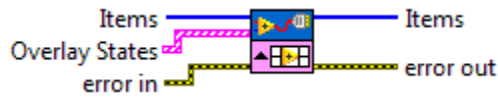
Turns on or off Icon overlays for the given Item, specified by previously registered Aliases. See mxLvAddIconOverlays.vi





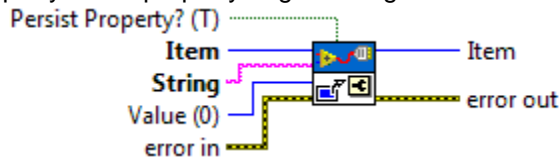
**44) mxLvSetIconOverlaysBatch.vi**

Turns on or off Icon overlays for the given Items, specified by previously registered Aliases. See mxLvAddIconOverlays.vi



**45) mxLvSetIntProperty.vi**

Writes an Integer property to the property bag for the given Item.



**46) mxLvSetIntPropertyBatch.vi**

Writes an Integer property to the property bag for the given Items.



**47) mxLvSetMenu.vi**

Specifies menu items. The order in which menu items appear is determined by their order in the "Menu Items" array. For each menu item the text display, description, icon and event handling command tag can be specified, please refer to the help for the "Menu Items" cluster elements for details.



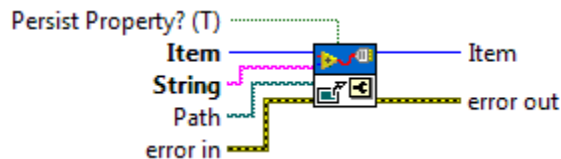
**48) mxLvSetName.vi**

Sets the name (the text that appears in the tree) for an item.



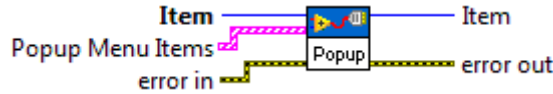
**49) mxLvSetPathProperty.vi**

Writes a Path property to the property bag for the given Item, taking care to make the path delimiters cross-platform.



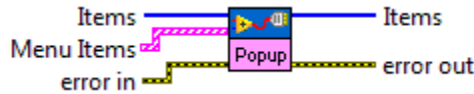
**50) mxLvSetPopupMenu.vi**

Specifies pop-up menu items for specified project item. The order in which menu items appear is determined by their order in the "Popup Menu Items" array. For each menu item the text display, description, icon and event handling command tag can be specified, please refer to the help for the "Popup Menu Items" cluster elements for details.



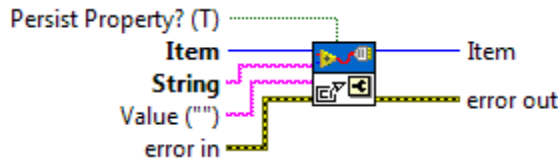
**51) mxLvSetPopupMenuBatch.vi**

Specifies pop-up menu items for specified project items. The order in which menu items appear is determined by their order in the "Menu Items" array. For each menu item the text display, description, icon and event handling command tag can be specified, please refer to the help for the "Menu Items" cluster elements for details.



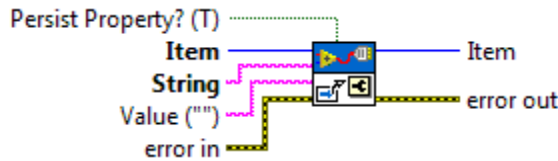
**52) mxLvSetProperty.vi**

Writes a property to the property bag for the given item. This is a polymorphic VI.



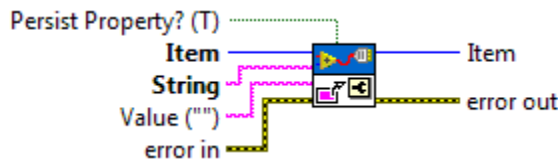
**53) mxLvSetRefProperty.vi**

Writes a reference property to the property bag for the given Item.



**54) mxLvSetStringProperty.vi**

Writes a String property to the property bag for the given Item.



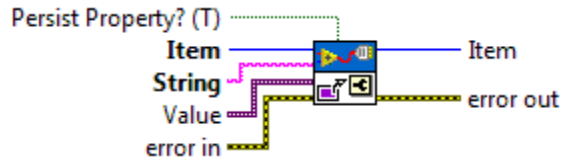
**55) mxLvSetToolbar.vi**

Specifies toolbar items. The order in which toolbar items appear is determined by their order in the "Toolbar Items" array. For each toolbar item the text display, description, icon and event handling command tag can be specified, please refer to the help for the "Toolbar Items" cluster elements for details.



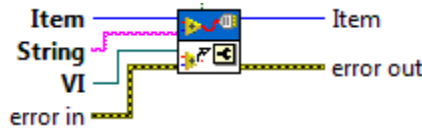
**56) mxLvSetVariantProperty.vi**

Writes variant property after converting to appropriate type to the property bag for the given Item.



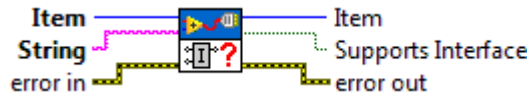
**57) mxLvSetVIRefProperty.vi**

Casts the given VI Reference to an Integer and writes the Integer to the property bag for the given Item.



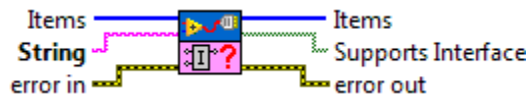
**58) mxLvSupportsInterface.vi**

Returns whether the given Item supports the specified interface.



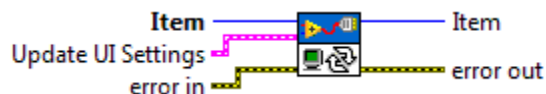
**59) mxLvSupportsInterfaceBatch.vi**

Returns whether the given Items support the specified interface.



**60) mxLvUpdateUI.vi**

Forces the UI to update the item based on the specified flags.



**61) mxLvUpdateUIBatch.vi**

Forces the UI to update the items based on the specified flags.

