ifChannelStore

The query methods (GetCatalog, GetStoreCatalog, GetPurchases and GetUpgrade) return information about products in the Channel Store. These are asynchronous methods. The method call returns immediately, and the requested information is returned later in an roChannelStoreEvent.

The order modifiers (SetOrder, ClearOrder and DeltaOrder) change the list of items (commonly called a "shopping cart") that will be the initial list used by the DoOrder purchase method.

Important Note: Because DoOrder() (and the other order modifiers) need to read the product type returned by GetCatalog(), only one instance of roChannelStore should ever be used for a complete purchase flow - for example, you should never create a separate roChannelStore object to complete a purchase that was initiated by a different instance of roChannelStore.

Implemented By

- roChannelStore

Supported Methods

- GetIdentity() as Integer
- GetCatalog() as Void
- GetStoreCatalog() as Void
- GetPurchases() as Void
- SetOrder(order as Object)
- ClearOrder() as Void
- DeltaOrder(code as Object, qty as Integer) as Integer
- GetOrder() as Object
- DoOrder() as Boolean
- FakeServer(enable as Boolean) as Void
- GetUserData() as Object
- GetPartialUserData(properties as String) as Object
- StoreChannelCredData(data as String) as Object
- GetChannelCred() as Object
- RequestPartnerOrder(orderInfo as roAssociativeArray, productID as String) as Object
- ConfirmPartnerOrder(confirmOrderInfo as roAssociativeArray, productID as String) as Object

Description of Methods

GetIdentity() as Integer

Returns a unique number for this object that can be used to identify whether a roChannelStoreEvent event originated from this object, by comparing with the roChannelStoreEvent object's GetSourceIdentity() value.

Note that the value can be any arbitrary value as assigned by the firmware, and should only be used for comparison purposes.
For example, the value should not be used as an array index.
For use as a look-up key, one option would be to use GetIdentity().ToStr() as an associative array key.

This function is available in firmware 7.5 or later.

GetCatalog() as Void

Requests the list of In-Channel products which are linked to the running channel.

If successful, a later roChannelStoreEvent will be received which contains an roList of roAssociativeArray items where each item contains the following parameter names with specified value type:

- String code
- String name
- String description
- String SDPosterUrl
- String HDPosterUrl
- String cost (Localized cost with local currency symbol)

GetStoreCatalog() as Void

Requests the list of globally available In-Channel products, which are available to all channels.

If successful, a later roChannelStoreEvent will be received which contains an roList of roAssociativeArray items, where each item contains the following parameter names with specified value type:

- String code
- String name
- String description
- String SDPosterUrl
- String HDPosterUrl
- String cost (Localized cost with local currency symbol)

GetPurchases() as Void

Requests the list of purchases associated with the current user account.

If successful, a later roChannelStoreEvent will be received which contains an roList of roAssociativeArray items, where each item contains the following parameter names with specified value type:

Parameter	Type	Description
code	string	The product identifier
cost	string	Localized cost of the item with local currency symbol
expirationDate	string	The subscription expiration date (ISO 8601 format)
freeTrialQuantity	integer	The free trial amount associated with the freeTrialType
freeTrialType	string	The free trial type ("Days" or "Months")
name	string	The item name
productType	string	The product type (ex. "MonthlySub")
purchaseDate	string	The purchase date (ISO 8601 format)
purchaseId	string	The transaction ID
qty	integer	The quantity purchased
renewalDate	string	The subscription renewal date (ISO 8601 format)

SetOrder(order as Object)

Sets the current Order (shopping cart) to the elements specified in the parameter, which must be an roList of roAssociativeArray items, where each item contains the following parameter names with specified value type:

- String code
- Integer qty

Passing an empty roList clears the Order, like calling ClearOrder().

ClearOrder() as Void

Clears the current Order (shopping cart). After this call, the Order is empty.

DeltaOrder(code as Object, qty as Integer) as Integer

Applies a change in quantity to one item in the current Order (shopping cart). If the item identified by code is not in the Order, it is added with the specified quantity. If the item already exists in the Order, qty is added to the quantity of this item in the Order. qty may be negative. The returned value is the quantity of the item remaining in the Order after applying the change. If this number is zero or negative, the item is deleted from the Order.

GetOrder() as Object

Retrieves the current Order. The returned object is an roList of roAssociativeArray items, where each item contains the following parameter names with specified value type:

- String code
- Integer qty

DoOrder() as Boolean

Displays the Roku Channel Store Product Purchase Screen populated with information from the current Order. The user can then either approve and complete the purchase, or cancel the purchase. If the user approves the order, this function returns true. Otherwise it returns false. In the case that the user approves, the channel should wait for and respond to the roChannelStoreEvent.isRequestSucceeded event to get the details of the completed transaction.

FakeServer(enable as Boolean) as Void

If enable is true, enables a test mode for the roChannelStore component.

This test mode short circuits communication to the Roku Channel store. It makes other methods get their responses to async queries and operations from configuration files, rather than actual server communication.

This should never be called in a production channel.

Note: Developer Blog: Supporting In App Purchases in Your Roku BrightScript Channels has more information regarding test mode.

GetUserData() as Object

The GetUserData() function provides a way to request user authorization to share the user's account information with the calling channel. The primary use case of this method is to facilitate partner account creation/updating within channels that have a customer billing relationship with Roku.

For example, a developer may have a Roku channel that offers a VOD subscription to users. This subscription may require an account with the content provider. The GetUserData() method could be called to read the user's account information in order to prepopulate an account registration screen.

When called, the method presents a dialog screen containing the user's account information, along with two buttons labeled Share and Don't Share. If the user presses the Don't Share button, GetUserData() returns invalid. If the user presses the Share button, GetUserData() returns an roAssociativeArray containing the following Roku account information for the channel user. All values are Strings.

- firstname
- lastname
- email
- street1
- street2
- city
- state
- zip
- country
- phone

> In order to call this function, the roChannelStore object needs to have a valid roMessagePort assigned to it by calling the ifChannelStore.SetMessagePort function. Also, since this method displays a dialog, the application must display a UI screen or canvas prior to calling the method. Otherwise, the behavior is undefined.

GetPartialUserData(properties as String) as Object

This function works like GetUserData(), but allows the caller to specify which user data elements to return. The specified values are also displayed in the user data dialog screen. To tell the function which properties to return, pass a string with a comma separated list of the attribute names. For example, to return only the email address and first name of the user's account, you would call GetPartialUserData("email, firstname"). The full set of user account properties that can be queried with the function is:

- firstname
- lastname
- email
- street
- city
- state
- zip
- country
- phone

StoreChannelCredData(data as String) as Object

Available since firmware version 8.1

This method can be used to store custom data (such as an OAuth token or a custom token) that can be retrieved by calling `GetChannelCred`. This data is stored securely in the cloud and can be retrieved by other devices linked to the same Roku account. Your channel can use the Roku `StoreChannelCredData` method to store an authentication artifact with Roku for a signed in user, associating that user with a particular Roku account. For more information, see Universal Authentication Protocol for Single Sign-On.

If the transaction is successful, an roAssociativeArray will be returned containing the following value:

Key	Type	Value
status	integer	An integer representing the request status. A successful request will return a status of 0.

If the transaction failed, the roAssociativeArray will contain the following value:

Key	Type	Value
errorCode	string	An error code representing why the transaction failed.

GetChannelCred() as Object

Available since firmware version 7.2

This function can be used to retrieve a Roku Partner Unique Customer Identifier (`roku_pucid`). The `PUCID` can be used in place of requiring the user to enter their email address or username again (ex. when setting up a new device on the same Roku account).

It returns an roAssociativeArray containing the following values:

Key	Type	Description
channelID	string	A string representing the channel ID (ex. "2213" for Roku Media Player)
json	string	A string in JSON format, with the following key-value pairs: **Key** / **Type** / **Description** error / string / A string containing an error message (if any). This value will be `null` (uninitialized) for a successful request. roku_pucid / string / This is an agnostic ID (in UUID format) representing the user. This value will be identical when retrieved in the same channel across devices linked to the same Roku account. token_type / string / Type of the returned token, e.g. "`urn:roku:pucid:token_type:pucid_token`" If the request fails, this *json* string will be empty.
publisherDeviceID	string	A unique identifier of the device. See GetPublisherId() for more details.
status	integer	An integer representing the request status. A successful request will return a status of 0.

RequestPartnerOrder(orderInfo as roAssociativeArray, productID as String) as Object

Available since firmware version 7.6

This function checks the user's billing status and is a prerequisite for `ConfirmPartnerOrder()` when doing transactional purchases. This function requires the following parameters:

Key	Type	Description
orderInfo	roAssociativeArray	contains two String values: **Key** / **Type** / **Description** priceDisplay / String / the price displayed price / String / the price to charge
productID	String	the product identifier as entered on the Developer Dashboard when the product was created

This function returns an roAssociativeArray containing the following values:

Key	Type	Description
id	String	This `ID` must be passed in the `confirmOrderInfo` parameter in `ConfirmPartnerOrder()`
status	String	`Success`
tax	String	Cost of tax (if applicable)
total	String	Total cost of transaction

If status is Failure, the roAssociativeArray will contain the following values:

Key	Type	Description
errorCode	String	An error code representing why the transaction failed
errorMessage	String	An error message explaining why the transaction failed
status	String	`Failure`

ConfirmPartnerOrder(confirmOrderInfo as roAssociativeArray, productID as String) as Object

Available since firmware version 7.6

This function is equivalent to doOrder() for transactional purchases. The user's billing status must first be confirmed with RequestPartnerOrder() prior to calling this function. This function requires the following parameters:

Key	Type	Description
confirmOrderInfo	roAssociativeArray	<table><tr><th>Key</th><th>Type</th><th>Required</th><th>Description</th></tr><tr><td>title</td><td>String</td><td>Required</td><td>the name of the content item (shown on user's invoice</td></tr><tr><td>priceDisplay [1]</td><td>String</td><td>Required</td><td>the original price displayed</td></tr><tr><td>price [1]</td><td>String</td><td>Required</td><td>the price to charge</td></tr><tr><td>orderID</td><td>String</td><td>Required</td><td>the ID returned from RequestPartnerOrder()</td></tr></table> [1] The currency symbol must not be included for `price` and `priceDisplay`.
productID	String	the product identifier as entered on the Developer Dashboard when the product was created

If the transaction is successful, an roAssociativeArray will be returned containing the following values:

Key	Type	Value
purchaseID	String	the transaction iD
status	String	`Success`

If the transaction failed, the roAssociativeArray will contain the following values:

Key	Type	Value
errorCode	String	An error code representing why the transaction failed

errorMessage	String	An error message explaining why the transaction failed
status	String	`Failure`