

ComponentLibrary

Table of Contents

- Description
 - Loading Component Libraries
 - Using Library Components
 - Unsigned Component Libraries
 - Fields
- Example

Extends: [Node](#)

Description

The **ComponentLibrary** node class downloads a library of custom SceneGraph components to be used in an application. The **ComponentLibrary** node should be used in a **Scene** node, such as **Scene** or **OverhangPanelSetScene**. One way to ensure that the library downloads before the SceneGraph application begins to compile the components for the application, is to begin the download in the `main.brs` file that creates the **Scene** node, by adding an **<interface>** field to the Scene node that can be used to monitor the download, and starts the application when the download is complete.

Loading Component Libraries

In addition to the custom components you create specifically for your application, you can also create custom component libraries that you can download as part of your application.

To do this, define a **ComponentLibrary** node in the **Scene** node for your application. Set the `id` field of the node to a unique library ID for your application, and specify the URL to download the library from in the `uri` field of the node. As the SceneGraph scene is constructed, the library will be downloaded from the specified URL, and component instances will be created as needed for the application.

Component libraries can be packaged and signed the same way that channels are packaged and signed. You side-load your library, package it using the packaging tool, and then host it on the server of your choosing.

Component libraries cannot be published to the Roku Channel Store for distribution.

Using Library Components

Components declared in a component library can be used inside the app just like custom components. Make sure to specify it in format `<libraryName>:<componentName>`. So say if a component named `LoadingIndicator` is defined in the `SampleComponentLib` library, then it may be used like this:

```
<SampleComponentLib>LoadingIndicator imageUri="pkg:/images/loader.png"
translation="[700, 200]"/>
```

Unsigned Component Libraries

Available since firmware version 7.5

Component libraries no longer need to be packaged or signed with the same devid as the channel starting with firmware version 7.5. However, unsigned component libraries are required to be served over HTTPS before they can be accessed by published channels. See additional cases below:

- **Sideloaded channels** can access **unsigned** component libraries over HTTP or HTTPS.
- **Published channels** can only access **unsigned** component libraries over HTTPS.
- **Published channels** can access **signed** component libraries over HTTP if the channel and the library share the same devid.

Fields

Field	Type	Default	Use										
id	string	no default	Set to a unique ID for the library for the application										
uri	uri	no default	The URL of the library to be downloaded										
loadStatus	value string	"none"	<p>Read-Only Indicates the progress of the library download. The possible values are:</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Meaning</th> </tr> </thead> <tbody> <tr> <td>none</td> <td>The default if the library is not being downloaded</td> </tr> <tr> <td>loading</td> <td>Library is downloading</td> </tr> <tr> <td>ready</td> <td>Library has downloaded successfully</td> </tr> <tr> <td>failed</td> <td>Download of the library has failed</td> </tr> </tbody> </table>	Value	Meaning	none	The default if the library is not being downloaded	loading	Library is downloading	ready	Library has downloaded successfully	failed	Download of the library has failed
Value	Meaning												
none	The default if the library is not being downloaded												
loading	Library is downloading												
ready	Library has downloaded successfully												
failed	Download of the library has failed												

Example

[ComponentLibraryTestChannel.zip](#) is a test channel that downloads the component library implemented from the source code in [ComponentLibrary.zip](#). The test channel then creates an instance of the <LoadingIndicator> component implemented in the library.