Screensavers

Table of Contents

- Overview
- Screensaver Context
- Entry points
- Private screensavers
    - Sharing active data
- Manifest entries
- Usable Components
- Scene Graph Screensavers
    - Example

Overview

A screensaver is a channel which is run automatically when the system has been idle for a period of time. The length of the period is settable by the user. If more than one screensaver is installed, one is selected by the user as the "current" screensaver. If the channel which was running when the system became idle has a private screensaver (see below), then that screensaver is run. Otherwise, the currently selected public screensaver is run.

The main operational difference between a screensaver and a normal channel is that screensavers do not accept user input (remote button presses, etc). When a screensaver is running, any user input will simply terminate the screensaver and return to the channel which was previously running.

Screensaver Context

When a screensaver is run, a new Brightscript context is created in which the screensaver executes. In other words, the screensaver does not share Brightscript data objects with the currently running channel. (However see **Sharing active data** below.)

A screensaver's BrightScript Debugger uses port 8087 rather than port 8085 as normal channels do. This keeps the screensaver console separate from the console for the running channel.

Entry points

A screensaver is just a channel which has a function named RunScreenSaver. It may or may not also have a RunUserInterface (or Main) function like normal channels. If it does not, it is a "pure" screensaver and appears only in the list of screensavers and not on the Roku Home screen (however see **Private screensavers** below). If it has a RunUserInterface (or Main) function in addition to RunScreenSaver, then it will appear both in the list of screensavers and also on the Home screen. When the Home screen icon is selected, the RunUserInterface function is called (or Main if no RunUserInterface function exists), whereas when the channel is run as a screensaver, the RunScreenSaver function is called.

If a screensaver channel also has a function named RunScreenSaverSettings, a "Custom Settings" button will appear when the user selects the screensaver in the Roku Settings menu. When the user clicks the Custom Settings button, the RunScreenSaverSettings function is called. This function should display a user interface to modify settings for the screensaver. Normally RunScreenSaverSettings would store data in the Registry to reflect the settings. RunScreenSaver can then read this data and modify its behavior according to the settings.

When RunScreenSaverSettings is called, the channel is run as a normal channel, not as a screensaver. In particular, it may (and should) accept user input, it may use any Brightscript Components, and the Brightscript console is on port 8085.

Private screensavers

A screensaver may be tagged as "private" by having an entry named "screensaver_private" in its manifest file. A private screensaver will be run as a screensaver only when the same channel is already running as a normal channel. Private screensavers do not appear in the list of screensavers in the Roku Settings menu and therefore cannot be selected by the user as the current screensaver. If a channel has a private screensaver, that screensaver is always used when the channel is running; the current screensaver is not used.

> Note: The use of the words "public" and "private" regarding screensavers is not related to whether the channel is a public or non-certified channel in the Roku Channel Store.

Sharing active data

A public screensaver normally has no relation to the channel which is running when the screensaver runs. However a private screensaver may wish to share information with the running channel. One way to do this is via shared files in the tmp: filesystem. A channel and its private screensaver share the same tmp: directory.

A common case is when the screensaver merely wishes to read data that is written by the main channel. The main channel may periodically write its current status to a file. A simple (but incorrect) implementation might look like this:

Read-only screensaver (INCORRECT)

```
' In the main channel
WriteAsciiFile("tmp:/status", my_status)

' In the screensaver
channel_status = ReadAsciiFile("tmp:/status")
```

Although this would usually work, it would not work correctly if WriteAsciiFile in the main channel wrote only part of the string to the status file, and then ReadAsciiFile in the screensaver read and returned that partial string before WriteAsciiFile completed writing the rest of the string. A more correct implementation might look like this:

Read-only screensaver (correct)

```
' In the main channel
WriteAsciiFile("tmp:/status.temp", my_status)  ' write to a temp file
MoveFile("tmp:/status.temp", "tmp:/status")    ' rename the temp file atomically

 ' In the screensaver
channel_status = ReadAsciiFile("tmp:/status")
```

Since Brightscript does not have mutual exclusion APIs or other concurrency support, simultaneous access to shared data must be handled by careful management and ordering of file operations. Only MoveFile, DeleteFile, and CreateDirectory (and the similar roFileSystem methods Rename, Delete and CreateDirectory) should be assumed to be atomic. In particular, ReadAsciiFile and WriteAsciiFile are not atomic.

Manifest entries

A screensaver should have an entry in its manifest file named "screensaver_title" whose value is the title of the screensaver (usually the same as the title of the channel).

If it is a private screensaver, it should have an entry named "screensaver_private" with a value of 1.

Usable Components

Normally a screensaver uses roScreen or roImageCanvas to display images on the screen. Components which accept user input are not supported in a screensaver. This is a list of components which **cannot** be used in a screensaver.

- roAudioPlayer
- roChannelStore
- roCodeRegistrationScreen
- roGridScreen
- roKeyboardScreen
- roListScreen
- roMessageDialog
- roOneLineDialog
- roParagraphScreen
- roPinEntryDialog
- roPosterScreen
- roSearchHistory
- roSearchScreen
- roSpringboardScreen
- roTextScreen
- roVideoPlayer
- roVideoScreen

Scene Graph Screensavers

Available since firmware version 7.2

You can also create screensavers in Roku Scene Graph. You should *not* create Scene Graph screen savers that require any type of user input (such as a keyboard), or have video playback.

Example

SGSimpleScreensaver.zip demonstrates a simple screen saver developed in Roku Scene Graph.