

Timer

Table of Contents

- [Description](#)
- [Fields](#)
- [Example](#)

Extends: [Node](#)

Description

The **Timer** node class generates an observable event after a specified amount of time has elapsed.

Fields

Field	Type	Default	Use								
control	string	"none"	Used to control the operation of the Timer node. Recognized values include: <table border="1"><thead><tr><th>Value</th><th>Effect</th></tr></thead><tbody><tr><td>none</td><td>No effect</td></tr><tr><td>start</td><td>Starts the Timer node operation</td></tr><tr><td>stop</td><td>Stops a running Timer node</td></tr></tbody></table>	Value	Effect	none	No effect	start	Starts the Timer node operation	stop	Stops a running Timer node
Value	Effect										
none	No effect										
start	Starts the Timer node operation										
stop	Stops a running Timer node										
repeat	Boolean	false	If set to <code>true</code> , the Timer node fires repeatedly, each time the specified <code>duration</code> field value elapses. If set to <code>false</code> , the Timer node only fires once until restarted.								
duration	time	1	Specifies the time in seconds before the Timer node fires after the <code>control</code> field value is set to <code>start</code> . To specify time values down to millisecond granularity, use a float type (0.001 equals one millisecond).								
fire	Event	N/A	Observe-Only Triggers observer callback functions when the Timer node fires. Please note that the timer observer callback executes on the render thread.								

Example

The following changes the text string on the display screen every five seconds as the **Timer** node generates a fire field observable event.

Timer Node Class Example

```
<?xml version="1.0" encoding="utf-8" ?>

<!--***** Copyright 2015 Roku Corp. All Rights Reserved. *****-->

<component name="timertest" extends="Group" >

<script type="text/brightscript" >
```

```
<![CDATA[

sub init()
  m.testtimer = m.top.findNode("testTimer")
  m.testtimer.control = "start"
  m.defaulttext = "Wait for it, wait for it..."
  m.alternatetext = "Timer fired!!!"

  m.testtimerlabel = m.top.FindNode("testTimerLabel")
  m.testtimerlabel.text = m.defaulttext
  m.textchange = false
  m.testtimer.ObserveField("fire", "changetext")
  m.top.setFocus(true)
end sub

sub changetext()
  if (m.textchange = false) then
    m.testtimerlabel.text = m.alternatetext
    m.textchange = true
  else
    m.testtimerlabel.text = m.defaulttext
    m.textchange = false
  end if
end sub

]]>
</script>

<children>

<Label
  id="testTimerLabel"
  width = "1280"
  translation = "[0,500]"
  horizAlign = "center"
  vertAlign = "center"
/>

<Timer
  id="testTimer"
  repeat="true"
  duration="5"
/>
```

```
</children>  
  
</component>
```