

# **[Tutorial] LG Web\_Activating or Deactivating Magic Remote Control Unit**

---

Version 1.1 – February 2012

**LGDEV-046**

Home Entertainment Company  
LG Electronics, Inc.

## Copyright

**Copyright © 2011 LG Electronics, Inc. All Rights Reserved.**

Though every care has been taken to ensure the accuracy of this document, LG Electronics, Inc. cannot accept responsibility for any errors or omissions or for any loss occurred to any person, whether legal or natural, from acting, or refraining from action, as a result of the information contained herein. Information in this document is subject to change at any time without obligation to notify any person of such changes.

LG Electronics, Inc. may have patents or patent pending applications, trademarks copyrights or other intellectual property rights covering subject matter in this document. The provision of this document does not give the recipient or reader any license to these patents, trademarks copyrights or other intellectual property rights.

No part of this document may be communicated, distributed, reproduced or transmitted in any form or by any means, electronic or mechanical or otherwise, for any purpose, without the prior written permission of LG Electronics, Inc.

This document is subject to revision without further notice.

All brand names and product names mentioned in this document are trademarks or registered trademarks of their respective owners.

# About This Document

## Revision History

Document Version	Date	Comment
1.1	February 6, 2012	Sample code is updated.
1.0	October 12, 2011	Initial Version

## Purpose

This document describes how to deactivate the Magic Remote Control Unit using Web open API.

## Reference Documents

Refer to the following documents:

- LG Web Application Development Guide
- LG Web Open API Reference Guide

## Conventions

### Codes

Source code and examples are indicated in the `grey Courier New` font.

### Note, Caution

Note and caution are used to emphasize information.  
The following samples describe when each is used.

---

#### Note

Contains information about something that is helpful to you.

---

---

#### Caution

Contains important information about something that you should know.

---

# Abbreviation

The following table defines the abbreviations used in this document.

Abbreviation	Description
RCU	Remote Control Unit

## Contents

<b>1</b>	<b>Introduction.....</b>	<b>6</b>
1.1	Overview .....	7
1.2	Needed APIs .....	8
<b>2</b>	<b>Creating Application.....</b>	<b>9</b>
2.1	Initializing the Page .....	10
2.2	Inputting Keys.....	11
2.3	Handling Events .....	12
2.4	Counting Time .....	14
2.5	Adding Mouse Event Handler .....	15
2.6	Source Code of mouse.html .....	16

## Tables

[Table 1] Description of the Needed APIs.....	8
---	---

## Figures

[Figure 1] Magic-RCU Deactivation Application.....	7
--	---



# 1 Introduction

---

This chapter provides an overview of this application and needed APIs.

1.1 Overview

1.2 Needed APIs

## 1.1 Overview

This application is designed to show how to deactivate the Magic-RCU (Remote Control Unit) using the Web Open API of LG Smart TV.

This application shows which method is used to deactivate the Magic-RCU and how Magic-RCU handles activation/deactivation events.

LG Smart TV SDK | Web Open API Tutorial  
File : /netcast/app/mouse.html

Smart TV

Web Open API List		
Methods	Properties	Events
window.NetCastMouseOff(time) window.NetCastGetMouseOnOff()		mouseon mouseoff

**View** Source

Mouse Status :

Mouse Off

[Figure 1] Magic-RCU Deactivation Application

## 1.2 Needed APIs

This application uses following Web Open API:

[Table 1] Description of the Needed APIs

API Class	Name	Description
Method	Window.NetCastMouseOff(time)	This API can be used by an LG Smart TV application author to deactivate the Magic-RCU and its pointer.
	Window.NetCastGetMouseOnOff()	This API can be used by an LG Smart TV application author to get the on or off status of the Magic-RCU. Its return value is "on" or "off".
Property	N/A	N/A
Event	mouseon	This event is generated when the Magic-RCU is activated.
	mouseoff	This event is generated when the Magic-RCU is deactivated.

For more information on these functions, refer to "LG Web Open API Reference Guide".

---

### Note

Log is used for checking the sequence of Web Open API; this will not be covered in the sample code description.

---





## 2 Creating Application

---

This chapter describes how to deactivate the Magic-RCU using Web open API.

2.1 Initializing the Page

2.2 Inputting Keys

2.3 Handling Events

2.4 Counting Time

2.5 Adding Mouse Event Handler

2.6 Source Code of mouse.html

## 2.1 Initializing the Page

Use the **initPage** function to set the basic functions of the application.

- 04: Records the last visited page when running the application.
- 07: Initializes the page.
- 08: Gets the source code of the page using the XMLHttpRequest object.
- 09: Sets the page ID.
- 10: Initializes the Log function.
- 13-17: Adds an event handler which will be executed when the corresponding button is pressed.
- 19: Calls the `updateMouseStatus()` function.

### Sample Code

```
01 : function initPage()
02 : {
03 :     //save page as last visited page
04 :     setLastVisitPage();
05 :
06 :     //common initialize function
07 :     commonInitialize();
08 :     requestSourceCode();
09 :     setPageID("Mouse");
10 :     jsLog.initLG();
11 :
12 :     //add onclick event handler
13 :     addEventHandler(document.getElementById("btn_back"),"click",onClickHandler);
14 :     addEventHandler(document.getElementById("btn_exit"),"click",onClickHandler);
15 :
16 :     //add event handler for test
17 :     addEventHandler(document.getElementById("start"),"click", onClickMouseOff);
18 :
19 :     updateMouseStatus();
20 : }
```

## 2.2 Inputting Keys

The **onUserInput** function is called by the onClickHandler function; it receives a key value as the userInput parameter from onClickHandler and creates the corresponding function for each key value to operate the key.

06: When the Back key is pressed, this code is executed.

### Sample Code

```
01 : //onUserInput function should be implemented
02 : function onUserInput(userInput)
03 : {
04 :     switch(userInput)
05 :     {
06 :         case VK_BACK : window.location.replace("../menu_netcast.html");
            break;
07 :     }
08 : }
```

## 2.3 Handling Events

The following functions are for event handlers added by the `initPage` function.

### **mouseon\_handler**

The event handler called when the mouse is turned on.

04: `updateMouseStatus` is called. It defines functions which will be executed when the corresponding button is pressed.

### **mouseoff\_handler**

The event handler called when the mouse is turned off.

10: `updateMouseStatus` is called. It defines functions which will be executed when the corresponding button is pressed.

### **updateMouseStatus**

Defines the action that will be taken depending on the mouse status.

16: Stores the mouse on or off status in the status variable.

17: Displays the value stored in the status variable.

19-23: If the mouse status is on and device is LG Browser4.0, Mouse off button is enabled and the message is printed.

24-27: If the mouse status is on and device is LG Browser5.0, the message is printed.

28-32: If the mouse status is off, Mouse off button is disabled and the message is printed.

### **checkNetCastVersion**

Checks the LG Browser version and decides registration of mouseoff event.

35: Stores version information in the `nBrowserVersion`.

38-42: If the device is LG Browser4.0, mouseoff event is registered and the message is printed.

43-47: If the device is LG Browser5.0, Mouse off button is disabled and the message is printed.

### **onClickMouseOff**

The event handler called when the Mouse off button is pressed.

53: Calls the `window.NetCastMouseOff(time)` API to turn off the mouse. To have the mouse turned off in 5 seconds, the parameter of 5 is entered.

54: Sets `timeCount` to 5 so it counts the remaining time of 5 seconds until the mouse is turned off.

55: Calls the `showLeftTime` function every 1 second to display the remaining time until the mouse is turned off. Set the timer.

## Sample Code

```

01 : function mouseon_handler()
02 : {
03 :     jsLog.lgevent('mouseon');
04 :     updateMouseStatus();
05 : }
06 :
07 : function mouseoff_handler()
08 : {
09 :     jsLog.lgevent('mouseoff');
10 :     updateMouseStatus();
11 : }
12 :
13 : function updateMouseStatus()
14 : {
15 :     jsLog.lgmethod('window.NetCastGetMouseOnOff()');
16 :     status = window.NetCastGetMouseOnOff();
17 :     setInnerTextById("mouse_status", status);
18 :
19 :     if(status == 'on' && nBrowserVersion == 4)
20 :     {
21 :         document.getElementById("mouse_off").style.visibility = "visible";
22 :         setInnerTextById("description", "");
23 :     }
24 :     else if(status == 'on' && nBrowserVersion >= 5)
25 :     {
26 :         setInnerTextById("description", "");
27 :     }
28 :     else
29 :     {
30 :         document.getElementById("mouse_off").style.visibility = "hidden";
31 :         setInnerTextById("description", "Press the OK button of mouse to
use mouse");
32 :     }
33 : }
34 :
35 : var nBrowserVersion = getBrowserVersion();
36 : function checkNetCastVersion()
37 : {
38 :     if(nBrowserVersion == 4) // NetCast 2.0
39 :     {
40 :         addEventHandler(document.getElementById("mouse_off"), "click",
onClickMouseOff);
41 :         setInnerTextById("APIIdescription", "In the LG DTV Emulator 2011,
mouseoff event is not supported.");
42 :     }
43 :     else if(nBrowserVersion >= 5) // NetCast 3.0
44 :     {
45 :         document.getElementById('mouse_off').style.visibility = "hidden";
46 :         setInnerTextById("APIIdescription", "In NetCast 3.0,
NetCastMouseOff(time) API is not supported and the mouse gets deactivated
when the halt of the mouse movement continues for 3 seconds only on TV
set.");
47 :     }
48 : }
49 : // "Mouse Off" click handler
50 : function onClickMouseOff()
51 : {
52 :     jsLog.lgmethod('window.NetCastMouseOff(time)');
53 :     window.NetCastMouseOff(5);
54 :     timeCount = 5;
55 :     timer = setInterval(showLeftTime, 1000);
56 :     setInnerTextById("description", "Mouse will be off after 5 second");
57 : }

```

## 2.4 Counting Time

The **showLeftTime** function counts the remaining time until the mouse is turned off. The following sample code counts and displays the remaining time until the mouse is set to off by the `onClickMouseOff` function.

- 07:     Decreases `timeCount` by 1.
- 08-09: If `timeCount` is greater than 0, the message "Mouse will be off after +  
          "`timeCount`" + second" will be displayed on the id = "description" area.
- 10-11: If `timeCount` is 0, the `clearInterval` is called the `setInterval` function stops the specified timer.

### Sample Code

```
01 : //show left time
02 : var timer;
03 : var timeCount = 5;
04 :
05 : function showLeftTime()
06 : {
07 :   timeCount--;
08 :   if(timeCount > 0)
09 :     setInnerTextById("description", "Mouse will be off after " +
      timeCount + " second");
10 :   else
11 :     clearInterval(timer);
12 : }
```

## 2.5 Adding Mouse Event Handler

The following sample code adds the event handlers of mouseon and mouseoff.

### Sample Code

```
01 : window.onmouseon = mouseon_handler;  
02 : window.onmouseleave = mouseoff_handler;
```

## 2.6 Source Code of mouse.html

Source code of mouse.html is as follows:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>App Template of API Unit Sample App</title>
<link rel="stylesheet" href="../../css/style.css">
<script language="javascript" src="../../js/keycode.js"></script>
<script language="javascript" src="../../js/common.js"></script>
<script language="javascript" src="../../js/menu.js"></script>

<script type="text/javascript" src="../../blackbirdjs/blackbird.js"></script>
<link type="text/css" rel="Stylesheet" href="../../blackbirdjs/blackbird.css" />

<script>

    //initialize page
    function initPage()
    {
        //save page as last visited page
        setLastVisitPage();

        //common initialize function
        commonInitialize();
        requestSourceCode();
        setPageID("Mouse");
        jsLog.initLG();

        //add onclick event handler
        addEventHandler(document.getElementById("btn_back"), "click",
onClickHandler);
        addEventHandler(document.getElementById("btn_exit"), "click",
onClickHandler);

        //add event handler for test
        checkNetCastVersion();
        // addEventHandler(document.getElementById("mouse_off"), "click",
onClickMouseOff);

        updateMouseStatus();
    }

    //onUserInput function should be implemented
    function onUserInput(userInput)
    {
        switch(userInput)
        {
            case VK_BACK :
window.location.replace("../menu_netcast.html"); break;
        }
    }

    function mouseon_handler()
    {
        jsLog.lgevent('mouseon');
        updateMouseStatus();
    }

    function mouseoff_handler()
    {
        jsLog.lgevent('mouseoff');
        updateMouseStatus();
    }

    function updateMouseStatus()
```



```

{
    jsLog.lgmethod('window.NetCastGetMouseOnOff()');
    status = window.NetCastGetMouseOnOff();
    setInnerTextById("mouse_status", status);

    if(status == 'on' && nBrowserVersion == 4)
    {
        document.getElementById("mouse_off").style.visibility =
"visible";
        setInnerTextById("description", "");
    }
    else if(status == 'on' && nBrowserVersion >= 5)
    {
        setInnerTextById("description", "");
    }
    else
    {
        document.getElementById("mouse_off").style.visibility =
"hidden";
        setInnerTextById("description", "Press the OK button of
mouse to use mouse");
    }
}

var nBrowserVersion = getBrowserVersion();
function checkNetCastVersion()
{
    if(nBrowserVersion == 4) // NetCast 2.0
    {
        addEventHandler(document.getElementById("mouse_off"),
"click", onClickMouseOff);
        setInnerTextById("APIdescription", "In the LG DTV Emulator
2011, mouseoff event is not supported.");
    }
    else if(nBrowserVersion >= 5) // NetCast 3.0
    {
        document.getElementById('mouse_off').style.visibility =
"hidden";
        setInnerTextById("APIdescription", "In NetCast 3.0,
NetCastMouseOff(time) API is not supported and the mouse gets deactivated when
the halt of the mouse movement continues for 3 seconds only on TV set.");
    }
}

// "Mouse Off" click handler
function onClickMouseOff()
{
    jsLog.lgmethod('window.NetCastMouseOff(time)');
    window.NetCastMouseOff(5);
    timeCount = 5;
    timer = setInterval(showLeftTime, 1000);
    setInnerTextById("description", "Mouse will be off after 5
second");
}

// show left time
var timer;
var timeCount = 5;
function showLeftTime()
{
    timeCount--;
    if(timeCount > 0)
        setInnerTextById("description", "Mouse will be off after " +
timeCount + " second");
    else
        clearInterval(timer);
}

window.onmouseoff = mouseoff_handler;
window.onmouseon = mouseon_handler;

```

```

</script>
</head>

<body ondragstart='return false' onselectstart='return false'
onload="javascript:initPage();">

<!-- title -->
<div class='SuiteTitle' >LG Smart TV SDK | Web Open API Tutorial</div>

<!-- navigation -->
<div class='SuiteNavigation'>
    <div style="float:left;">File : netcast/app/mouse.html</div>
</div>

<div class='SuiteTitleLine'> </div>

<!-- test contents -->
<div id='content_body' class='ContentArea'>
    <div class='ApiListTitleArea'>Web Open API List</div>
    <div class='ApiListArea'>
        <div class='MethodTitleArea'>
            Methods
            <div class='MethodListArea'>
                window.NetCastMouseOff(time)<br>
                window.NetCastGetMouseOnOff()
            </div>
        </div>
        <div class='PropertyTitleArea'>
            Properties
            <div class='PropertyListArea'>

        </div>
        </div>
        <div class='EventTitleArea'>
            Events
            <div class='EventListArea'>
                mouseon<br>
                mouseoff<br>
            </div>
        </div>
    </div>

    <div class='ViewTitleArea'>
        <div id='tabViewArea' class='SelectedViewArea' style='float:left;'
onclick="showView();">View</div>
        <div id='tabCodeArea' class='UnselectedViewArea'
style='float:right;' onclick='showCode();'>Source</div>
    </div>
    <div id='view'>
        <div class='ViewArea'>
            <table border="0" cellpadding="0" cellspacing="0">
                <tr height="50px">
                    <td width=200px align=left><div>Mouse
Status :</div></td>
                    <td width=200px align="left"><div
id="mouse_status"></div></td>
                    <td width=100px align="left"></td>
                </tr>
                <tr height="50px">
                    <td width=200px align=left><div id="mouse_off"
class="executeButton">Mouse Off</div></td>
                    <td width=200px align="left"><div></div></td>
                    <td width=100px align="left"></td>
                </tr>
            </table>
            <table border="0" cellpadding="0" cellspacing="0">
                <tr height="50px">
                    <td align=left><div id="description"
class="blueColor"></div></td>
                </tr>
                <tr height="50px">

```

```

        <td align=left><div id="APIdescription" ></div></td>
    </tr>
</table>
</div>
</div>
<div style="visibility: hidden" id='codeview'>
    <textarea class="SourceCodeArea" value=""
id='sourcecode'></textarea>
</div>

</div>

<!-- button and copyright -->
<div class='SuiteButtonArea'>

    <!-- back key description -->
    <div id='btn_back' class='buttonDescription'>BACK</div>

    <!-- exit key description -->
    <div id='btn_exit' class='buttonDescription'>EXIT</div>

    <!-- copyright -->
    <div class='copyright'>Copyright LG Electronics</div>
</div>

</body>
</html>

```