

# Contents

- [1 Characteristics of iRidium Script](#)
- [2 Ways of Creating and Launching Scripts](#)
- [3 iRidium API](#)
  - [3.1 GUI API](#)
  - [3.2 List API](#)
  - [3.3 Gestures API](#)
  - [3.4 Sound API](#)
  - [3.5 Drivers API](#)
  - [3.6 Tokens API](#)
  - [3.7 Systems API](#)
- [4 Animation Library](#)
- [5 iRidium DDK](#)
- [6 Ways of Script Debugging](#)
- [7 Examples of Using iRidium Script](#)
- [8 Information about JavaScript](#)
  - [8.1 Web Sites on JavaScript](#)
  - [8.2 Books on JavaScript](#)
- [9 Demo Projects](#)
- [10 Use in Your Projects](#)

**iRidium Script** (iRidium Script API и iRidium DDK) are powerful tools for your automation. They can complete any automation task. Now you can control any equipment from any control panel.

**iRidium Script** is based on **Java Script**. To have access to iRidium objects **iRidium API**. To create drivers use **iRidium DDK**.

All these tools are multiplatform. Once created the script or driver operate similarly on any iRidium supported OS (iOS, Android, Mac OS and Windows)

## Characteristics of iRidium Script

[Characteristics of iRidium Script:](#)

- [Script Main Concepts](#)
- [Script Description of Uploading and Performing Scripts](#)
- [Working with Modules](#)
- [Forming Command Chains](#)
- [IntellHelp](#)
- [Debugging](#)
- [Compatibility](#)
- [API change history](#)

## Ways of Creating and Launching Scripts

[Ways of Creating and Launching Scripts:](#)

- [Launching Scripts by Creating Listeners](#)
- [Launching Scripts Using the \*\*Script Call Macros\*\*](#)

## iRidium API

iRidium API is a set of classes, methods, functions and events which enable the access from iRidium JavaScript to the interface items and driver system.

### [GUI API](#)

GUI API is a set of functions and events for working with graphic items (buttons, sliders, popups, etc.)

#### Functions

|                                  |  |
|----------------------------------|--|
| <a href="#">IR.CreateItem</a>    | Creating an item                         |
| <a href="#">IR.DeleteItem</a>    | Deleting an item                         |
| <a href="#">IR.GetItem</a>       | Changing item properties                 |
| <a href="#">IR.PagesCount</a>    | Counting project pages                   |
| <a href="#">IR.PopupsCount</a>   | Counting project popups                  |
| <a href="#">IR.CurrentPage</a>   | Receiving an identifier of the open page |
| <a href="#">IR.GetPage</a>       | Referring to a page                      |
| <a href="#">IR.GetPopup</a>      | Referring to a popup                     |
| <a href="#">IR.ShowPage</a>      | Opening a page                           |
| <a href="#">IR.ShowPopup</a>     | Opening a popup                          |
| <a href="#">IR.HidePage</a>      | Closing a page                           |
| <a href="#">IR.HidePopup</a>     | Closing a popup                          |
| <a href="#">IR.HideAllPopups</a> | Closing all popups                       |
| <a href="#">ItemsCount</a>       | Counting items                           |
| <a href="#">GetState</a>         | Referring to a state                     |

#### Events

|                                    |  |
|------------------------------------|--|
| <a href="#">EVENT_ITEM_PRESS</a>   | Event is activated when pressing on the item   |
| <a href="#">EVENT_ITEM_RELEASE</a> | Event is activated when releasing the item   |
| <a href="#">EVENT_ITEM_CHANGE</a>  | Event is activated when changing the EditText item                                   |
| <a href="#">EVENT_MOUSE_DOWN</a>   | Event is activated when the mouse button is pressed down                             |
| <a href="#">EVENT_MOUSE_UP</a>     | Event is activated when the mouse button is released                                 |
| <a href="#">EVENT_MOUSE_MOVE</a>   | Event is activated when the mouse button is pressed down and moved                   |
| <a href="#">EVENT_TOUCH_DOWN</a>   | Event is activated when the item on the screen is pressed down by a finger           |
| <a href="#">EVENT_TOUCH_UP</a>     | Event is activated when the item on the screen is released by a finger               |
| <a href="#">EVENT_TOUCH_MOVE</a>   | Event is activated when the item on the screen is pressed down by a finger and moved |

### [List API](#)

List API is a set of functions and events for working with lists

#### Functions

|                                    |                      |
|------------------------------------|----------------------|
| <a href="#">IR.CreateItem</a>      | Creating a list      |
| <a href="#">List_Name.Template</a> | Assigning a template |

|                                       |                          |
|---------------------------------------|--------------------------|
| <a href="#">List_Name.CreateItem</a>  | Filling a list           |
| <a href="#">List_Name.Clear()</a>     | Clearing a list          |
| <a href="#">List_Name.DeleteItem</a>  | Deleting list items      |
| <a href="#">List_Name.SetPosition</a> | Setting up list position |

## Events

|                                      |                           |
|--------------------------------------|---------------------------|
| <a href="#">IR.EVENT_ITEM_SELECT</a> | Pressing on the list item |
|--------------------------------------|---------------------------|

## Gestures API

### Functions

|                                     |                    |
|-------------------------------------|--------------------|
| <a href="#">IR.AddRecognizer</a>    | Adding a gesture   |
| <a href="#">IR.RemoveRecognizer</a> | Deleting a gesture |

### Events

|  |                        |
|--|------------------------|
| <a href="#">IR.EVENT_GESTURE_BEGIN</a> | Processing of gestures |
|--|------------------------|

## Sound API

### Functions

|                                  |   |
|----------------------------------|---|
| <a href="#">IR.PlaySound</a>     | Playing a sound file                            |
| <a href="#">IR.StopSound</a>     | Stop playing a sound file                       |
| <a href="#">IR.StopSlot</a>      | Stop playing a sound file in the indicated slot |
| <a href="#">IR.StopAllSounds</a> | Stop playing all sounds                         |

## Drivers API

### Functions

|                                       |  |
|---------------------------------------|--|
| <a href="#">IR.CreateDevice</a>       | Creating a driver  |
| <a href="#">Connect</a>               | Connection to a device   |
| <a href="#">Disconnect</a>            | Disconnection from a device  |
| <a href="#">IR.GetDevice</a>          | Referring to a device  |
| <a href="#">Set</a>                   | Setting up values in the device channel                                  |
| <a href="#">Send</a>                  | Sending commands to a device   |
| <a href="#">InvokeAction</a>          | Sending commands to a UPNP device  |
| <a href="#">Subscribe</a>             | Subscribing to UPNP events   |
| <a href="#">UnSubscribe</a>           | Unsubscribing from UPNP events   |
| <a href="#">HtmlDecode</a>            | Substituting Html symbols  |
| <a href="#">JSON.Stringify</a>        | Converting a JSON object into a string                                   |
| <a href="#">JSON.Parse</a>            | Converting a string into a JSON object                                   |
| <a href="#">new XML</a>               | Creating XML objects   |
| <a href="#">XML.ToString</a>          | Converting an XML object into a string                                   |
| <a href="#">SetFeedback</a>           | Writing values in the feedback channel                                   |
| <a href="#">GetFeedback</a>           | Receiving values from the feedback channel                               |
| <a href="#">SetParameters</a>         | Changing connection parameters   |
| <a href="#">GetCommandAtName</a>      | Referring to a command by its name or identifier                         |
| <a href="#">GetCommandAtPos</a>       | Referring to a command by its position index                             |
| <a href="#">GetCommandsCount</a>      | Receiving the quantity of commands                                       |
| <a href="#">GetFeedbackAtName</a>     | Receiving values from the feedback channel by their names or identifiers |
| <a href="#">GetFeedbackAtPos</a>      | Receiving values from the feedback channel by their position index       |
| <a href="#">GetFeedbacksCount</a>     | Receiving the quantity of feedback channels                              |
| <a href="#">HexArrayToAsciiString</a> | Converting an array of Hex symbols into an ASCII string                  |
| <a href="#">IR.SendNotification</a>   | Sending a local notification on iOS devices                              |
| <a href="#">IR.ClearNotify</a>        | Clearing the list of notifications on iOS devices                        |

## Events

|  |   |
|--|---|
| <a href="#">EVENT_RECEIVE_DATA</a>         | Receiving data from a device in the byte format |
| <a href="#">EVENT_RECEIVE_TEXT</a>         | Receiving strings from a device                 |
| <a href="#">EVENT_RECEIVE_EVENT</a>        | Receiving events from a device (UPnP Event)     |
| <a href="#">EVENT_ONLINE</a>               | Connection to a device is established           |
| <a href="#">EVENT_OFFLINE</a>              | Connection to a device is lost                  |
| <a href="#">EVENT_TAG_CHANGE</a>           | Changing a tag value                            |
| <a href="#">EVENT_DEVICE_FOUND</a>         | Finding UPnP devices                            |
| <a href="#">EVENT_CHANNEL_SET</a>          | Activating of commands                          |
| <a href="#">EVENT_APP_ENTER_BACKGROUND</a> | Application goes into the background mode       |
| <a href="#">EVENT_APP_ENTER_FOREGROUND</a> | Application goes into the foreground mode       |
| <a href="#">EVENT_APP_WILL_TERMINATE</a>   | Application finishes its work                   |
| <a href="#">EVENT_RECIEVE_NOTIFY</a>       | Receiving local notifications                   |

## [Tokens API](#)

### Reading functions: [IR.GetVariable](#)

|                                |   |
|--------------------------------|---|
| <a href="#">IR.GetVariable</a> | <a href="#">Get time</a>  |
| <a href="#">IR.GetVariable</a> | <a href="#">Get date</a>  |
| <a href="#">IR.GetVariable</a> | <a href="#">Receive GPS coordinates and parameters</a>                |
| <a href="#">IR.GetVariable</a> | <a href="#">Receive data from the compass</a>                         |
| <a href="#">IR.GetVariable</a> | <a href="#">Receive data from the accelerometer</a>                   |
| <a href="#">IR.GetVariable</a> | <a href="#">Receive data from the gyroscope</a>                       |
| <a href="#">IR.GetVariable</a> | <a href="#">Receive data about the device display</a>                 |
| <a href="#">IR.GetVariable</a> | <a href="#">Receive information about the network</a>                 |
| <a href="#">IR.GetVariable</a> | <a href="#">Receive information about the battery</a>                 |
| <a href="#">IR.GetVariable</a> | <a href="#">Receive the current volume on the device</a>              |
| <a href="#">IR.GetVariable</a> | <a href="#">Receive the state of vibration</a>                        |
| <a href="#">System.Update</a>  | <a href="#">Returns the state of the update mode: 0 - off, 1 - on</a> |

### Writing functions: [IR.SetVariable](#)

|                                |  |
|--------------------------------|--|
| <a href="#">IR.SetVariable</a> | <a href="#">Writing a value in the token</a>             |
| <a href="#">System.Update</a>  | <a href="#">Sets up the update mode: 0 - off, 1 - on</a> |

[↑ Back](#)

## [Systems API](#)

### Functions

|                                   |  |
|-----------------------------------|--|
| <a href="#">IR.Log</a>            | Outputting data into a console               |
| <a href="#">IR.AddListener</a>    | Activating Listener                          |
| <a href="#">IR.RemoveListener</a> | Remove Listener                              |
| <a href="#">IR.ClearInterval</a>  | Remove delay function                        |
| <a href="#">IR.GetSystemInfo</a>  | Receiving the volume of the allocated memory |
| <a href="#">IR.Exit</a>           | Exiting the application                      |
| <a href="#">IR.SetInterval</a>    | Run a function when the time delay           |
| <a href="#">IR.SetTimeout</a>     | Cycle with a given recurrence interval       |
| <a href="#">IR.Execute</a>        | Start of programs                            |
| <a href="#">IR.StartAction</a>    | Starting a macros                            |
| <a href="#">IR.HWID</a>           | Get HWID(UDID) device                        |

### Events

|  |  |
|--|--|
| <a href="#">IR.EVENT_START</a>         | Preset commands are performed at the application launch        |
| <a href="#">IR.EVENT_WORK</a>          | Commands are performed while the application is running        |
| <a href="#">IR.EVENT_EXIT</a>          | Commands are performed when exiting the application            |
| <a href="#">IR.EVENT_ORIENTATION</a>   | Commands are performed when the device changes its orientation |
| <a href="#">IR.EVENT_KEYBOARD_SHOW</a> | Commands are performed when a keyboard is opened               |

# [Animation Library](#)

## iRidium DDK

[iRidium DDK](#) - detailed instructions for creating a **driver** on the example of the **Marantz SR7007** device.

## Ways of Script Debugging

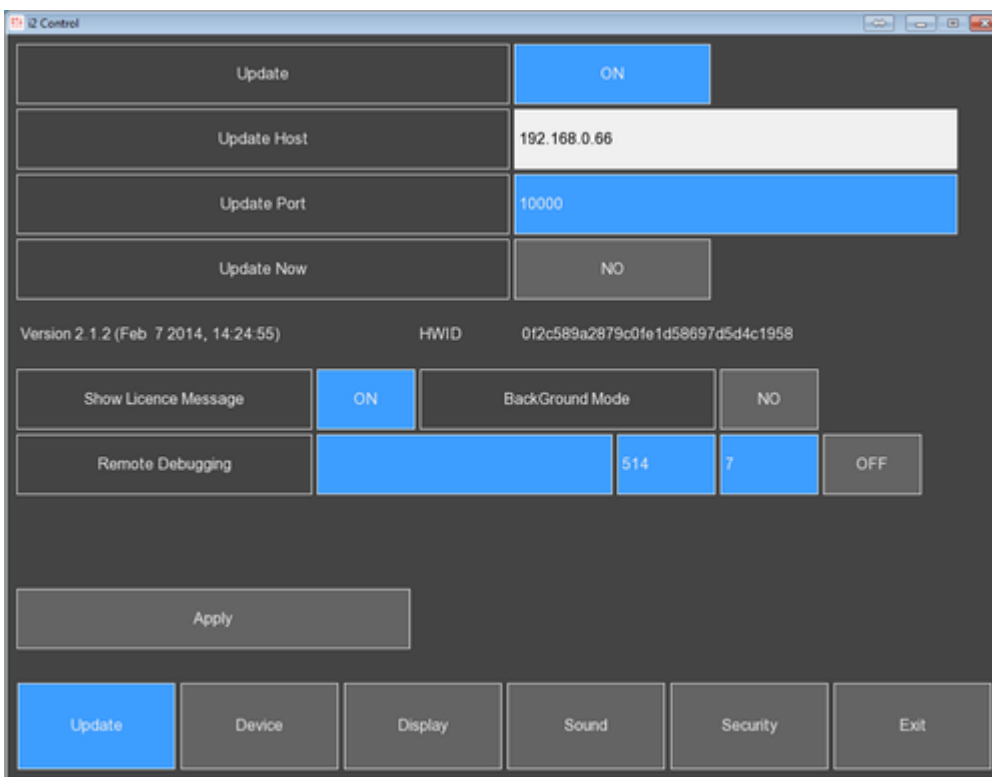
### Windows

For debugging scripts created on Windows use Emulator of GUI Editor, which has its own logging system. To open Emulator logging, press on **F4** when emulator is running.

### iOS, OS X, Android

It is not possible to launch logging on devices with installed iRidium App but there is a method of Remote Debugging which can be activated through the settings of [iRidium App](#).

To launch Remote Debugging launch iRidium App on your control panel and [go to its settings](#). The launch of Remote Debugging is made in the **Update** section of the application settings.



'Update'

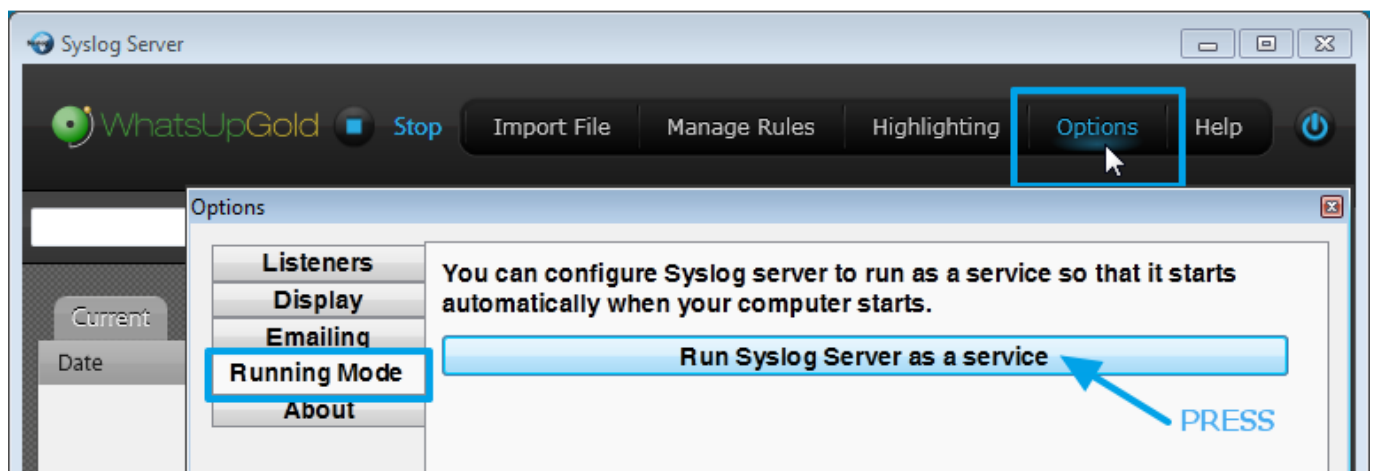
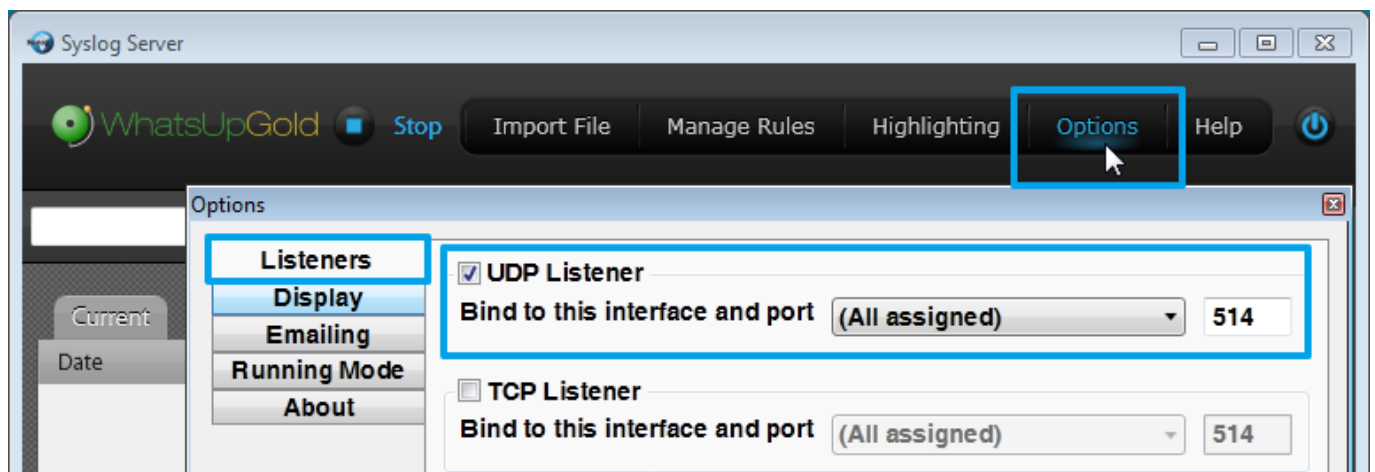
*Indicate the local IP-address of the PC (Host) where the server is launched and the port of the listener (Port) - **514** in the App settings on the control panel. Press the Apply button and*

restart iRidium App on the panel.

## Settings of the Syslog server:

Remote debugging is performed with the help of the Syslog server which is installed on your PC and receives data from remote iOS or Android devices. With the help of this tool you can see the log of your control panel on your PC.

To start work install and set up any *Syslog server* for Windows on your PC, for example, [Ipswitch Syslog Server](#). In the settings indicate the UDP port the server will receive data from: **514** and set the Syslog Server for the service mode:



At the next launch iRidium App will send information to the Syslog server on your PC.

## Examples of Using iRidium Script

### [Examples of Using iRidium Script](#)

- [Working with Keyboards and EditBox](#)
- [Working with Color Schemes](#)

# Information about JavaScript

## Web Sites on JavaScript

- <http://www.codecademy.com/> RUS/ENG
- <http://javascript.ru/> RUS
- <http://learn.javascript.ru/>RUS
- <http://www.w3schools.com/js/default.asp>ENG
- <http://help.dottoro.com/ljsdao.x.php>ENG
- <http://mas-html.narod.ru/js01.htm>RUS

## Books on JavaScript

- [JavaScript Enlightenment](#), Author: **Cody Lindley**, Language: **ENG**

Knowledge level: **intermediate**

Detailed review of JavaScript outlook through analysis of integrated objects and nuances.

- [Eloquent JavaScript](#), Author: **Marijn Haverbeke**, Language: **ENG**

Knowledge level: **beginner**

Introductory book on JavaScript and programming on the whole.

- [Building A JavaScript Framework](#), Author: **Alex Young**, Language: **ENG**

Knowledge level: **advanced**

Selected articles from the cyclus «Let's Make a Framework».

- [Mastering NodeJS](#), Author: **TJ Holowaychuk**, Language: **ENG**

Knowledge level: **advanced**

This book helps to learn how to write a web server with highly competitive connections.

- [Smooth CoffeeScript](#), Author: **E. Hoigaard**, Language: **ENG**

Knowledge level: **intermediate**

Introduction to CoffeeScript with focus on clarity, abstraction and verification.

- [Node Beginner](#), Author: **Manuel Kiessling**, Language: **ENG**

Knowledge level: **intermediate**

The goal of the book is to give the starting point in development of applications in Node.js and at the same time to teach «advanced» programming on JavaScript.

- [jQuery Fundamentals](#), Author: **Rebecca Murphey**, Language: **ENG**

Knowledge level: **beginner**  
The book on JavaScript and jquery.

- [Learning Javascript Design Patterns](#), Author: **Addy Osmani**, Language: **ENG**

Knowledge level: **beginner**  
This book examines conventional and contemporary patterns of programming in JavaScript.

- [Developing Backbone.js Applications](#), Author: **Addy Osmani**, Language: **ENG**

Knowledge level: **intermediate**  
The book examines the theory of MVC(Model-View-Controller) and shows how to develop applications using models, views, collections and routing in Backbone.js

- [The little Book on CoffeeScript](#), Author: **Alex MacCaw**, Language: **ENG**

Knowledge level: **advanced**  
This book is written to help to study CoffeeScript, understand the best practical aspects and start creating incredible applications.

## Demo Projects

### [Pages and Popups](#)

Creating and deleting pages/popups. Changing properties of pages/popups. Working with device orientation, etc.

### [Graphic Items](#)

Creating and deleting items. Changing item properties. Counting the number of items and item properties.

### [Gestures](#)

Activating, deactivating and processing gestures.

### [Tokens](#)

Data writing and reading from tokens.

### [Sounds](#)

Turning sounds on/off using iRidiumScript

### [System Tokens](#)

Referring to system tokens.

### [Drivers](#)

Creation of drivers, performing data sending.

### [Lists](#)

Creating and deleting lists. Changing list properties, etc.

### [Keyboards and EditBox](#)

Types of keyboards, receiving data from keyboards.

### [Color Schemes](#)

Working with colors in iRidiumScript.

### [Xml-objects](#)

Working with xml-objects in iRidiumScript.

### [Json-objects](#)

Working with json-objects in iRidiumScript.

[↑ Back](#)

## Use in Your Projects

[Learn the Names of Items Pressed](#)  
[Subscribe to Pressings on All Items](#)