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

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

Events

EVENT_ITEM_PRESS	Event is activated when pressing on the item
EVENT_ITEM_RELEASE	Event is activated when releasing the item
EVENT_ITEM_CHANGE	Event is activated when changing the EditBox item
EVENT_MOUSE_DOWN	Event is activated when the mouse button is pressed down
EVENT_MOUSE_UP	Event is activated when the mouse button is released
EVENT_MOUSE_MOVE	Event is activated when the mouse button is pressed down and moved
EVENT_TOUCH_DOWN	Event is activated when the item on the screen is pressed down by a finger
EVENT_TOUCH_UP	Event is activated when the item on the screen is released by a finger
EVENT_TOUCH_MOVE	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

IR.CreateItem	Creating a list
List_Name.Template	Assigning a template

List_Name.CreateItem	Filling a list
List_Name.Clear()	Clearing a list
List_Name.DeleteItem	Deleting list items
List_Name.SetPosition	Setting up list position

Events

IR.EVENT_ITEM_SELECT	Pressing on the list item

Gestures API

Functions

IR.AddRecognizer	Adding a gesture
IR.RemoveRecognizer	Deleting a gesture

Events

IR.EVENT_GESTURE_BEGIN	Processing of gestures

Sound API

Functions

IR.PlaySound	Playing a sound file
IR.StopSound	Stop playing a sound file
IR.StopSlot	Stop playing a sound file in the indicated slot
IR.StopAllSounds	Stop playing all sounds

Drivers API

Functions

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

Events

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

Tokens API

Reading functions: IR.GetVariable

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

Writing functions: IR.SetVariable

IR.SetVariable	Writing a value in the token
System.Update	Sets up the update mode: 0 - off, 1 - on

↑ Back

Systems API

Functions

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

Events

IR.EVENT_START	Preset commands are performed at the application launch
IR.EVENT_WORK	Commands are performed while the application is running
IR.EVENT_EXIT	Commands are performed when exiting the application
IR.EVENT_ORIENTATION	Commands are performed when the device changes its orientation
IR.EVENT_KEYBOARD_SHOW	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

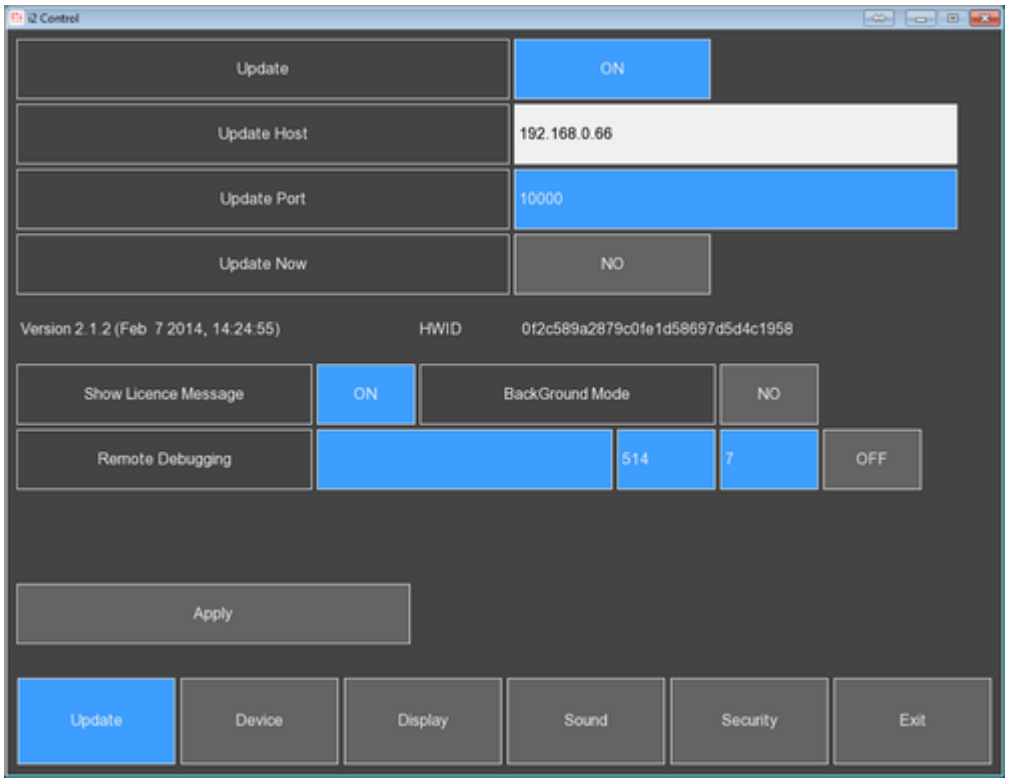
Windows

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'

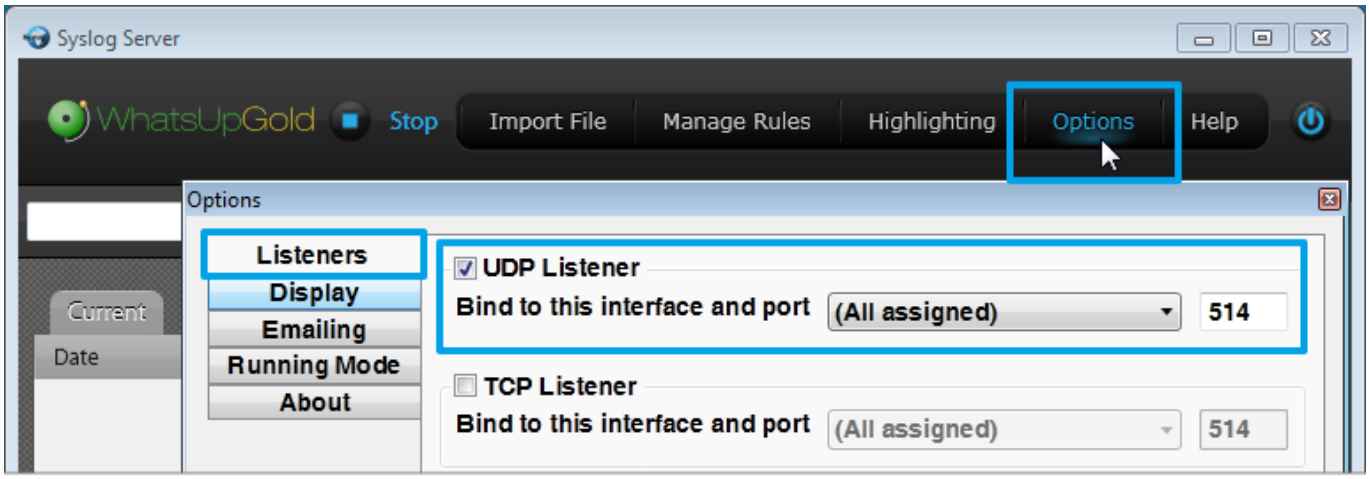
> *Indicate the local IP-address of the PC (Host) where the server is launched and the port of the listener (Port) - **514** in the App settings on the control panel. Press the Apply button and*

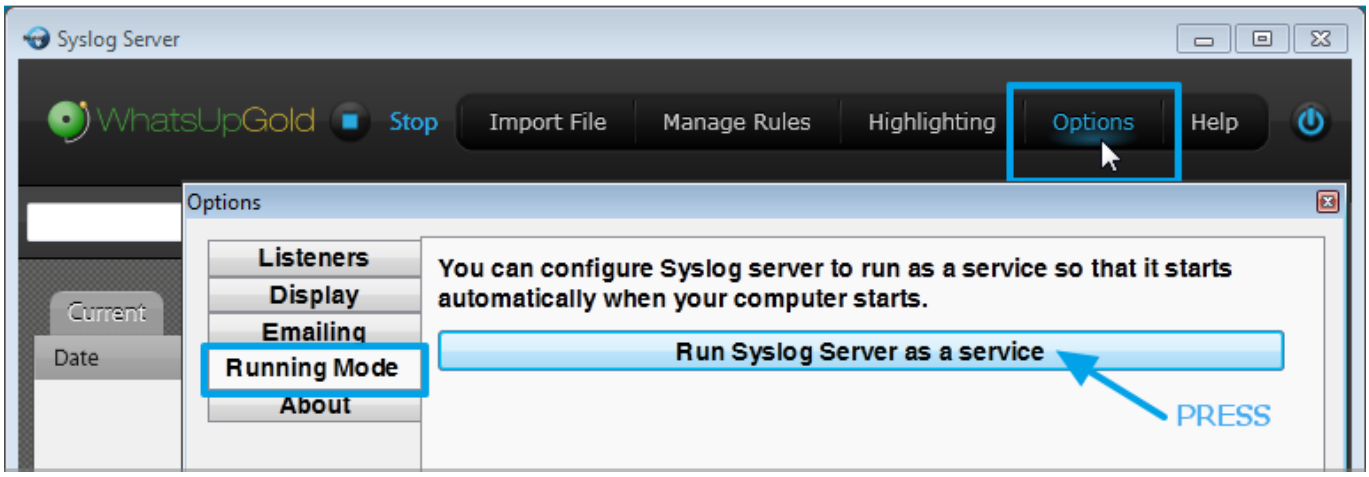
restart iRidium App on the panel.

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/ljsdaoxj.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 CoffeScript 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