

Contents

- [1 Work Principles of AV & Custom Systems \(HTTP\)](#)
- [2 Creation and Setting up of the AV & Custom Systems \(HTTP\) Driver](#)
- [3 Structure of the AV & Custom Systems Driver](#)
 - [3.1 Driver Tokens](#)
 - [3.2 Commands and Feedbacks](#)
- [4 Sending Commands to AV Equipment](#)
 - [4.1 GET](#)
 - [4.2 POST and PUT](#)
- [5 Working with AV & Custom Systems \(HTTP\) Commands](#)
- [6 Creation of Your Own Base of AV & Custom Systems \(HTTP\) Commands](#)
 - [6.1 Creating Data Bases](#)
 - [6.1.1 Switching Between the Local and Internet Connection](#)
- [7 Downloads](#)

iRidium for AV & Custom Systems (HTTP) allows you to control any equipment with Ethernet-interfaces or Wi-Fi modules. The control is performed via TCP/IP, locally or via the Internet using the HTTP (HTTPs) protocol.

Conditions for working with the driver:

- possibility to control the selected equipment via HTTP (HTTPs)
- commands for controlling equipment via HTTP (HTTPs) and the protocol
- possibility to create scripts if the protocol suggests authorization and requires data exchange with control panels

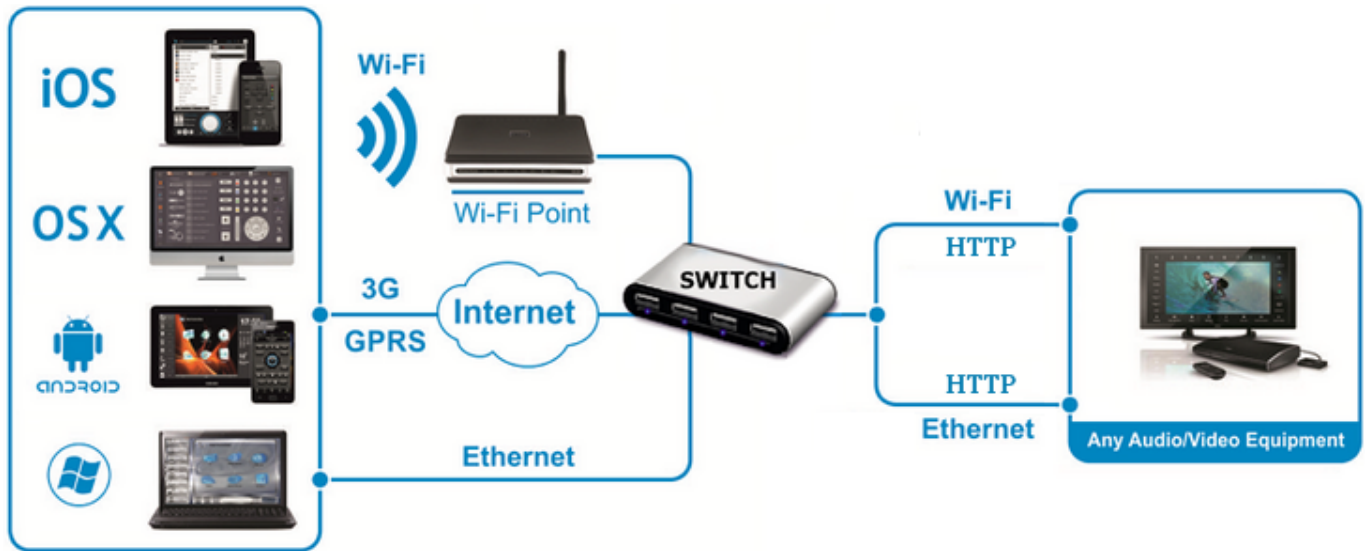
Description of driver work:

The "AV & Custom Systems (HTTP)" driver opens and supports connection session with equipment for sending commands. The list of commands is formed by the user in the process of creating a project. Feedback from the equipment is received by processing data by the script created in the iRidium project iRidium (see [iRidium DDK](#) and [iRidium Script API](#)).

⚠ These instructions describe creation of scripts for processing data received from equipment. Receipt of feedback from equipment is described in the [iRidium DDK](#) section.

Work Principles of AV & Custom Systems (HTTP)

In iRidium you can use the universal "AV & Custom Systems" (HTTP) driver to connect to your Audio-Video and other equipment. The scheme of communication between iRidium and controlled equipment:



Protocol for data sending: HTTP, HTTPs.

Supported methods: GET, POST, PUT.

Supported command format: ASCII.

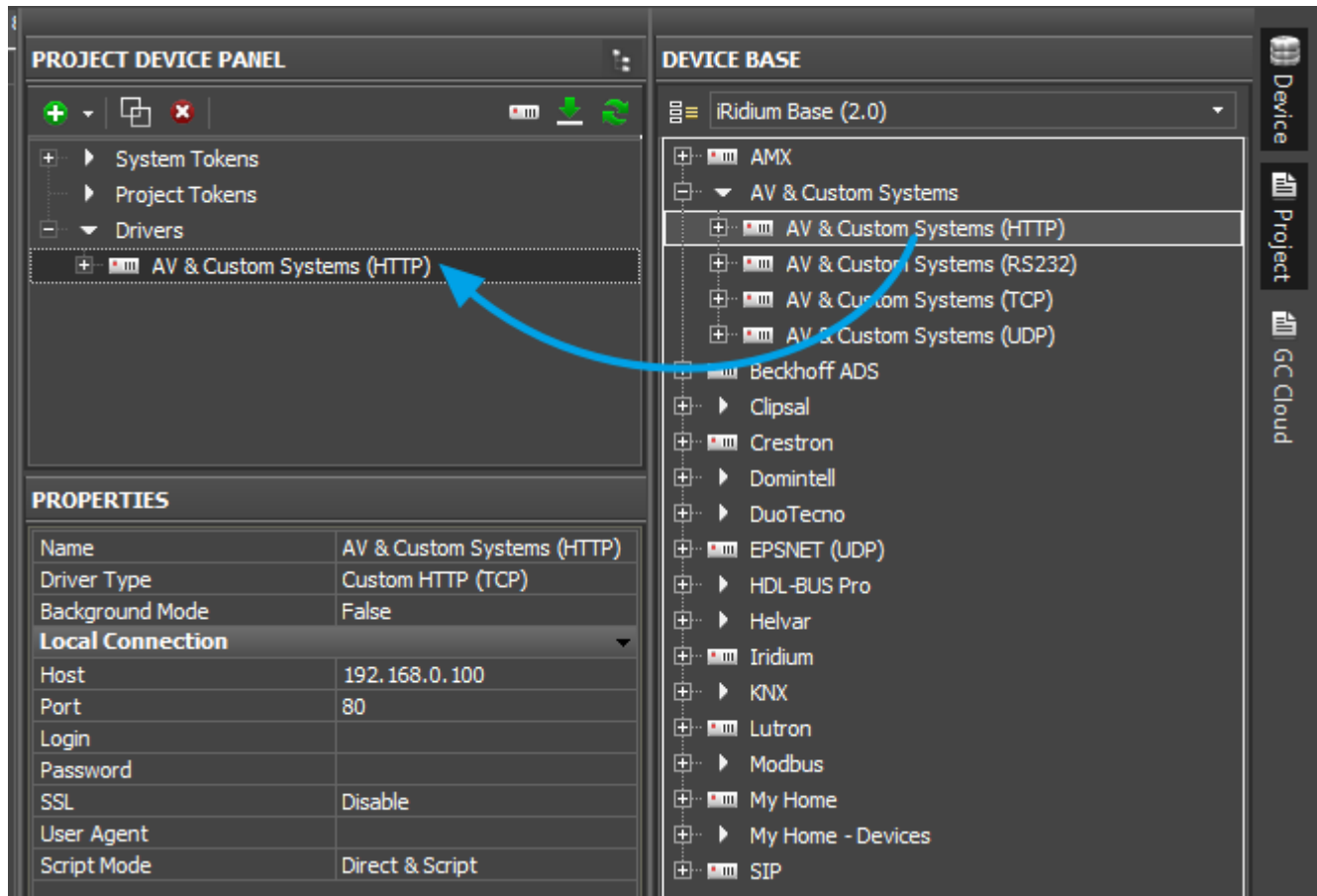
Feedback from equipment: there is none in the standard variant. It is implemented with the help of iRidium Script, see [iRidium DDK](#).

Required licenses: Device License Pro for AV & Custom Systems (see. [Licensing](#)). It works with any equipment controlled with the help of "AV & Custom Systems". It enables use of any scripts in your projects.

[↑ Back](#)

Creation and Setting up of the AV & Custom Systems (HTTP) Driver

Select the AV & Custom Systems tab in the DEVICE BASE window of [iRidium GUI Editor](#). Open it and find "AV & Custom Systems (HTTP)". Drag the driver into the PROJECT DEVICE PANEL window:



Properties of connection to equipment using AV & Custom Systems (HTTP):

- **Host** - the IP-address of the controlled equipment (local, external or the domain name)
- **Port** - the TCP port for connection to equipment
- **Login** - the login for connection to equipment
- **Password** - the password for connection to equipment
- **SSL** - the cryptographic protocol which enables secure connection between the client and the server
- **User Agent** - the part of the HTTP request which includes the name and the version of the application/browser used for accessing the web-interface
- **Script Mode** - the mode of command sending and communication with scripts in iRidium projects
 - Direct & Script - data from the command will be sent directly to the equipment and processes by the script system
 - Script Only - commands created inside the driver are not sent directly to the equipment but are processed by the script system first.

To connect to equipment via the Internet:

1. Indicate the public address of your Internet router in the **Host** field
3. Set up [Port Forwarding](#) on your router to enable remote control of your equipment .

You can learn the public IP-address of your router with the help of external resources, for example [\[1\]](#)

To switch between local and Internet connection:

Use the example presented [below](#).

[↑ Back](#)

Structure of the AV & Custom Systems Driver

The AV & Custom Systems in iRidium projects consists of 3 parts:

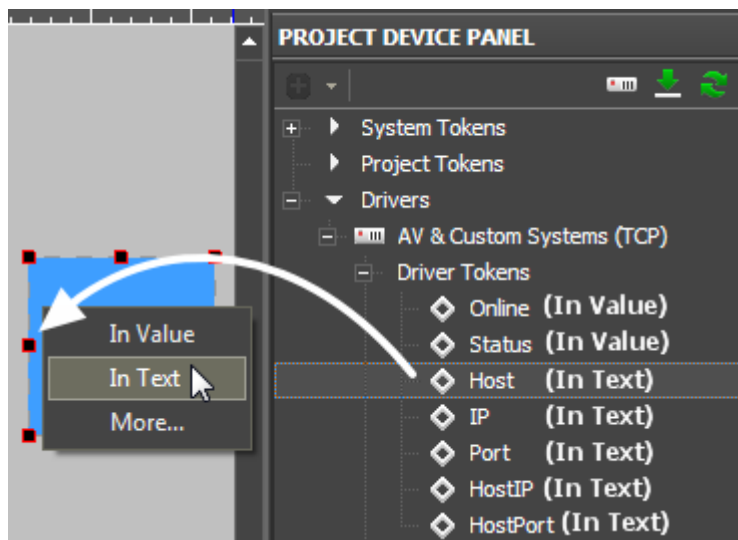
- **Driver Tokens** - the list of variables which store information about connection to equipment
- **Commands** - the list of commands which can be sent to equipment (or script) by pressing interface buttons
- **Feedbacks** - the list of feedback channels where data from the controlled equipment (processed by the iRidium script system) can be written.

Properties and applications of the driver parts:

Driver Tokens

[Driver Tokens](#) - variables storing the status of connection to the controlled equipment. These properties can be read only.

To use a token, drag it on the graphic item (the token value can be output in the item text field or it can be used to change the item state).

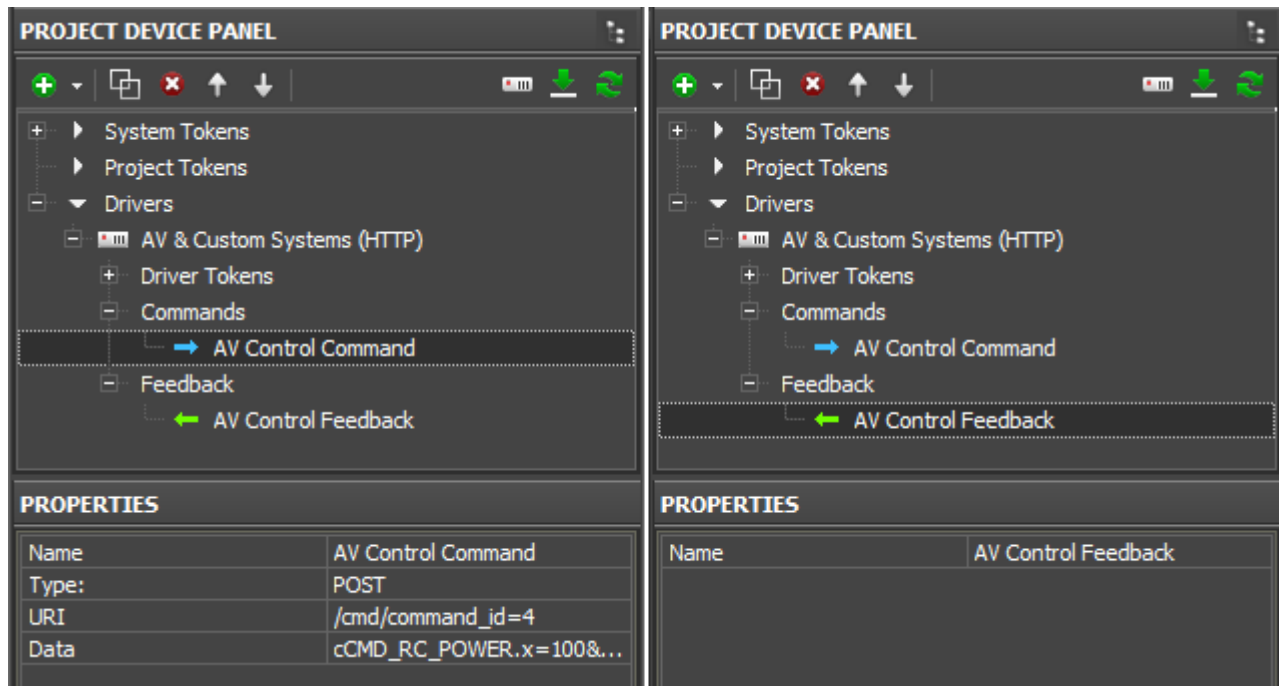


Online	a state of connection to the controlled system (Online/Offline = 1/0)
Status	the status of connection to the system (Offline/Connecting/Online/Waiting = 0...3)
Host	the domain name of the remote system
HostPort	the port of the remote system to which iRidium App is connected
IP	the IP-address of the control panel
HostIP	an IP-address of the remote system to which iRidium is connected
Port	the local port of the client through which the connection to the remote device is established

Commands and Feedbacks

Commands - the list of commands which can be sent to the equipment (or scripts) by pressing interface buttons. It is filled at random. The number of commands is not limited.

Feedbacks - the list of feedback channels where data from the controlled equipment (processed by the iRidium script system) can be written. The data written in the Data filed of the Feedback channel can be read using the script.



Commands

- **Name** - the command name, at random
- **Type** - the method defining actions with data
 - **GET** - to get (forward the request string - URI)
 - **POST** - to add, change, remove (forward the set of data)
 - **PUT** - to add, substitute (forward the set of data)
- **URI** - the unique resource identifier
- **Data** - the data in the request body

Feedback

- **Name** - the feedback channel name. You can refer to the channel from scripts for writing data.

To assign a command to a graphic item:

drag it on the graphic item and select the event on which it should be sent:

- **Press** - pressing on the item
- **Release** - releasing the item
- **Hold** - holding the item
- **Move** - moving on the item (it is mostly used for Level)

To assign a feedback channel to a graphic item:

drag it on the graphic item and select what it should affect:

- **In Value** - when receiving any non-zero value the item will change its state (State 1/State 2). It is used for visualizing changes by changing the item appearance.
- **In Text** - when receiving any data they will be written in the item text field
- **More...** - the received data can affect any other item properties (see the available properties in the dialog window)

[↑ Back](#)

Sending Commands to AV Equipment

Select the commands you need to add in the list of the AV & Custom Systems (HTTP) driver. They can be stored in one of the following formats:

- **GET** - to get (forward the request string - URI)
- **POST** - to add, change, remove (forward the set of data)
- **PUT** - to add, substitute (forward the set of data)

You can see how to form requests further.

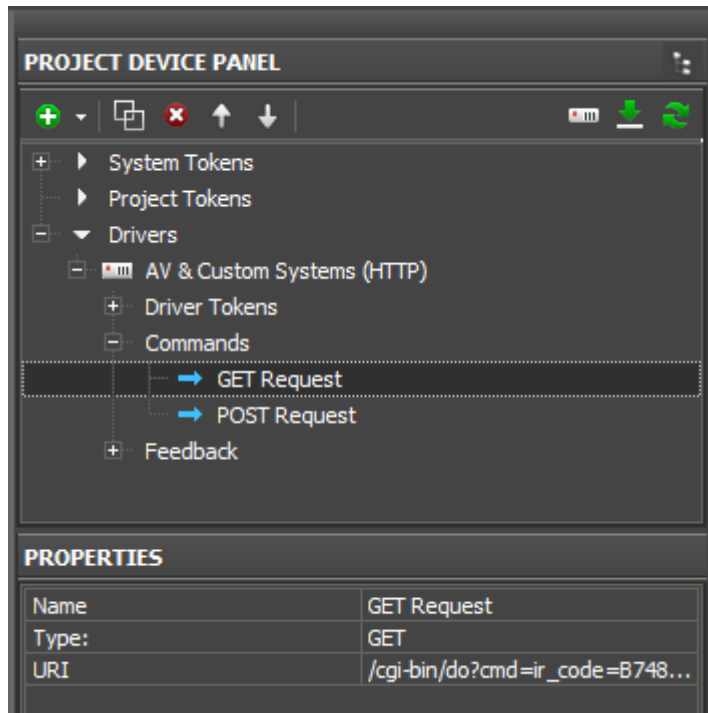
[↑ Back](#)

GET

Add a GET request in the "URI" window:

- * remove the part of the request where the IP-address and the port of referring to the resource are stored
- * add the request body in URI including the back slash before it
- **Red** indicates the part of the GET-request which should be written in the URI window
- **IP-address** and **port** of connection to the device controlled via the web-interface are indicated in the driver properties

Initial command (in documentation):	http://192.168.0.100:81/cgi-bin/do?cmd=ir_code=B748BF00
In iRidium (the URI field):	/cgi-bin/do?cmd=ir_code=B748BF00
Initial command (in documentation):	http://<ip address>:<port>/<command_body>
In iRidium (the URI field):	/<command_body>



[↑ Back](#)

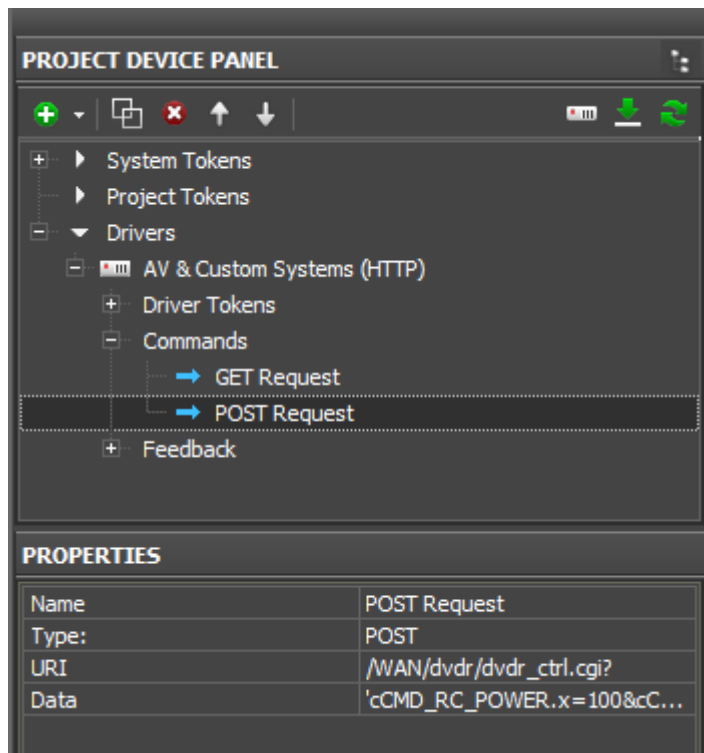
POST and PUT

Внесите части запроса в окна URI и Data:

- * separate the resource identifier and add it in the URI window
- * separate the request body and add it in the Data window
- * **include the request body in single quotes in the Data window**

- **Green** indicates the part of the POST-request which should be written in the URI window
- **Red** indicates the part of the POST-request which should be written in the Data window **including it in single quotes**.

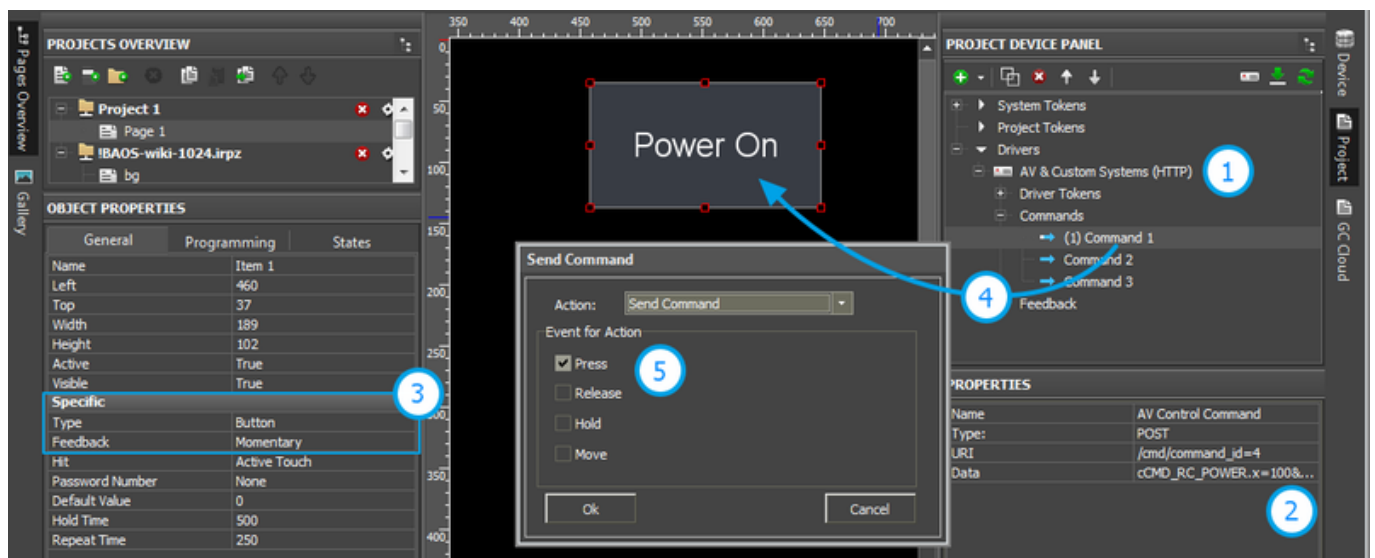
POST-запрос:	
<pre>POST /YamahaRemoteControl/ctrl HTTP/1.1 Accept: */* Accept-Language: ru Referer: http://192.168.10.102/ Content-Type: text/xml; charset=UTF-8 Accept-Encoding: gzip, deflate User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; BRI/2) Host: 192.168.10.100 Content-Length: 105 Connection: Keep-Alive Cache-Control: no-cache Cookie: mstnc=1 <YAMAHA_AV cmd="PUT"><Power_Control><Power>Standby</Power></Power_Control></System></YAMAHA_AV></pre>	<pre>URI: /YamahaRemoteControl/ctrl Data: '<YAMAHA_AV cmd="PUT"><System><Power_Control><Power>Standby</Power></Power_Control></System></YAMAHA_AV>'</pre>
POST-запрос:	
<pre>POST /WAN/dvdr/dvdr_ctrl.cgi? HTTP/1.1 User-Agent: MEI-LAN-REMOTE-CALL Host: 192.168.1.105 Content-Length: 39 cCMD_RC_POWER.x=100&cCMD_RC_POWER.y=100</pre>	<pre>URI: /WAN/dvdr/dvdr_ctrl.cgi? Data: 'cCMD_RC_POWER.x=100&cCMD_RC_POWER.y=100'</pre>



[↑ Back](#)

Working with AV & Custom Systems (HTTP) Commands

Example of using commands for AV & Custom Systems (HTTP):



1. Add the AV & Custom Systems (HTTP) driver in your iRidium project. Set up properties of connection to the equipment.
2. Create a command.
3. Set up the button which will be responsible for sending the command.

Type: Button

Feedback: Momentary

4. Drag the command on the button.

5. Select the event the on which the command will be sent:

Press – pressing on the item

Release – releasing the item

Hold – holding the item

Move – moving on the item (it is mainly used for Level)

⚠ You cannot change the command in the Data field when the project is already uploaded on your control panel. It means that if your command, for example, is responsible for setting the volume to 50%, it cannot be assigned to Level for gradual volume regulation. It will work only with Button and set the volume to 50% at each pressing.

⚠ To send values (e.g. volume) from Level, you need to from the command with the help of scripts (see [iRidium Script API](#) and [iRidium DDK](#)), not in the project tree.

[↑ Back](#)

Creation of Your Own Base of AV & Custom Systems (HTTP) Commands

To use AV & Custom Systems (HTTP) in any iRidium project you need to save it in the iRidium data base.

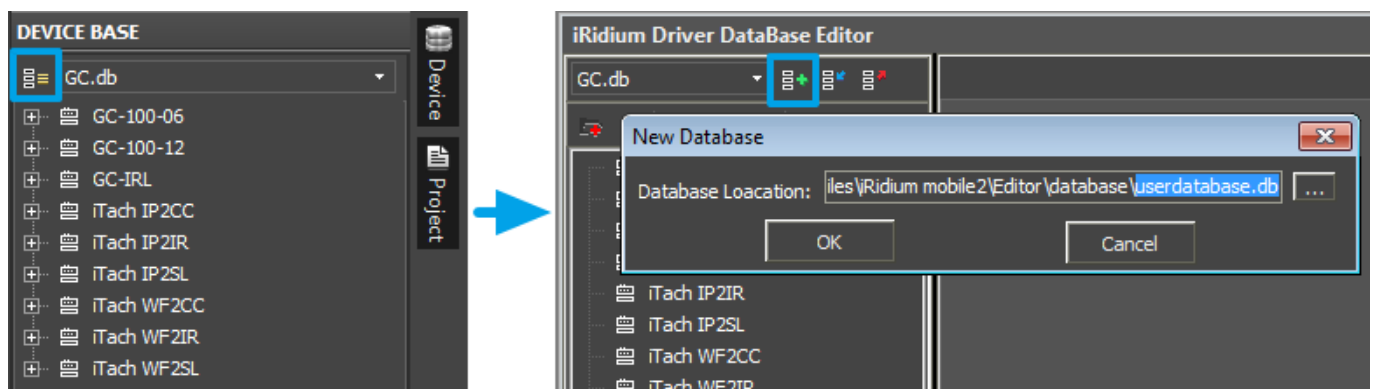
Use **DB Editor** for adding devices. New devices can be added only to custom data bases as standard data bases cannot be edited.

⚠ Creation of data bases is not the only way of transferring commands from one iRidium project to another. If you created a device with commands in your iRidium project, you can import it to another project using **File > Import > *.irpz**

[↑ Back](#)

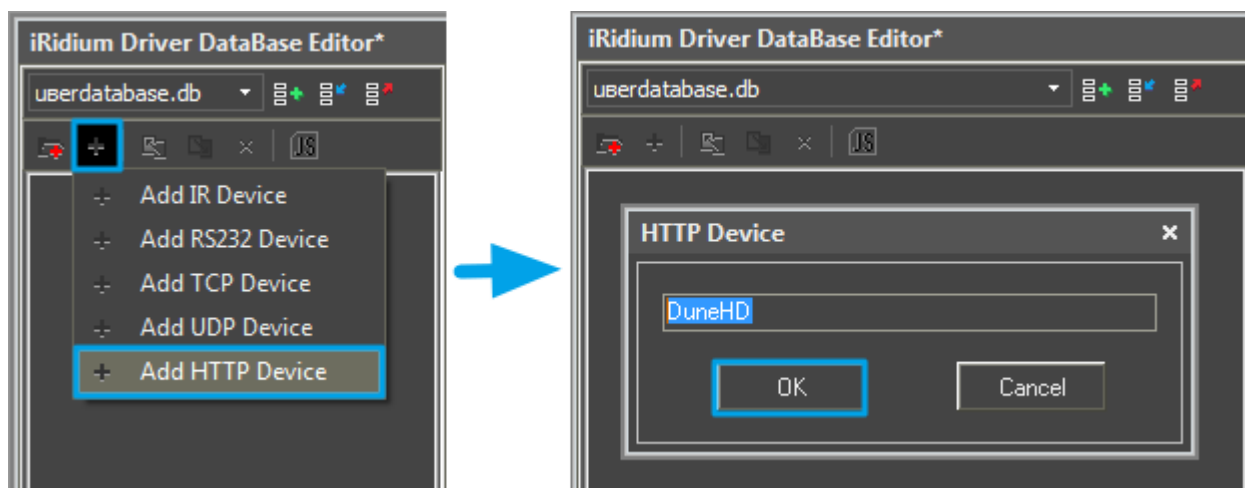
Creating Data Bases

Use custom data bases for storing your devices. They are created in DB Editor. When creating a data base, indicate its name and directory for saving.



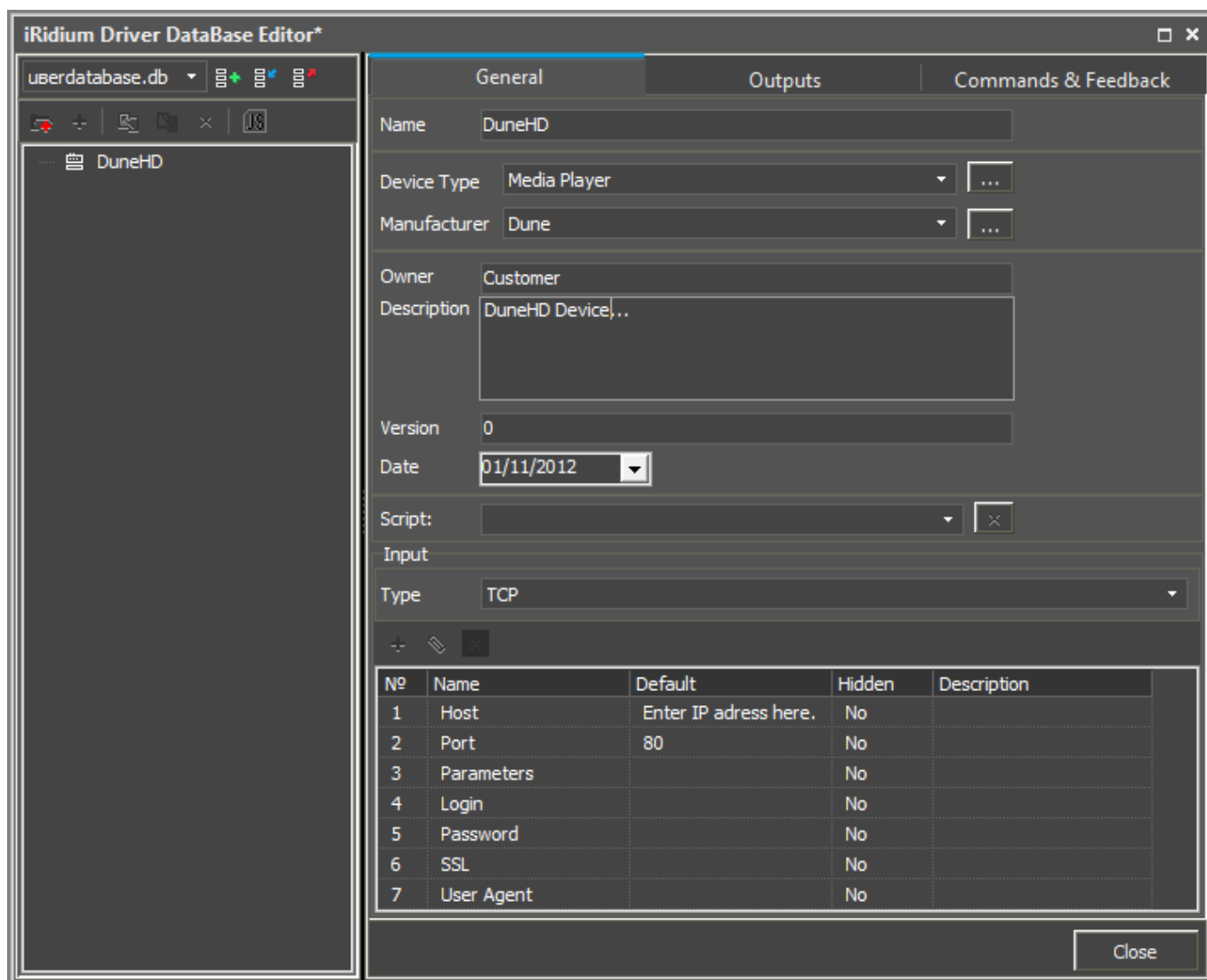
Creating new HTTP devices:

The type of the created device - HTTP (HTTP Device) - defines its transport part and characteristics.



Setting up HTTP devices

Describe new devices to help with their further identification:



- **Device Type** - the type of the controlled device (TV, DVD, Blu Ray, ...)
- **Manufacturer** - the name of the device manufacturer
- **Description** - description of the device, it can include hyper links
- **Date** - the date of creating the driver

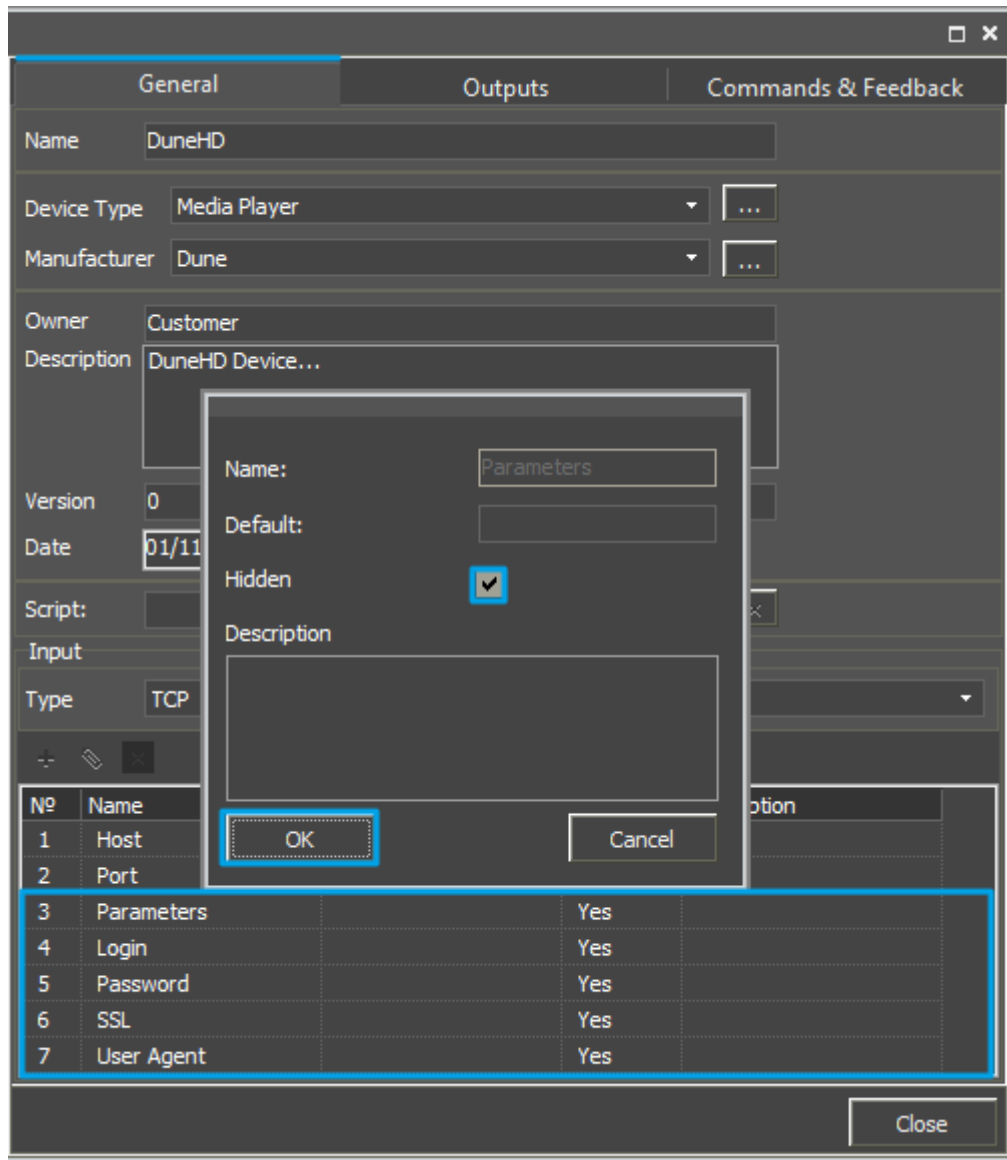
Properties for connection:

- **Host** - the IP-address of the device in your network
- **Port** - the port the device uses

Optional properties:

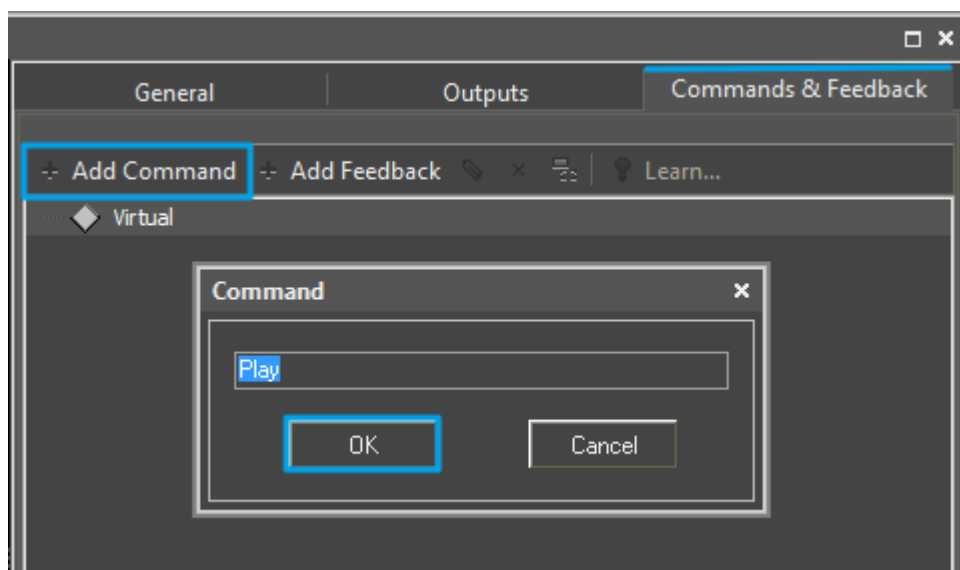
- **Login**
- **Password**
- **SSL** - the cryptographic protocol which enables secure connection between the client and the server
- **User Agent** - the part of the HTTP request which includes the name and the version of the application/browser, computer OS and language used

The optional properties can be hidden: click two times on the property to open the window for settings and select **Hidden**.



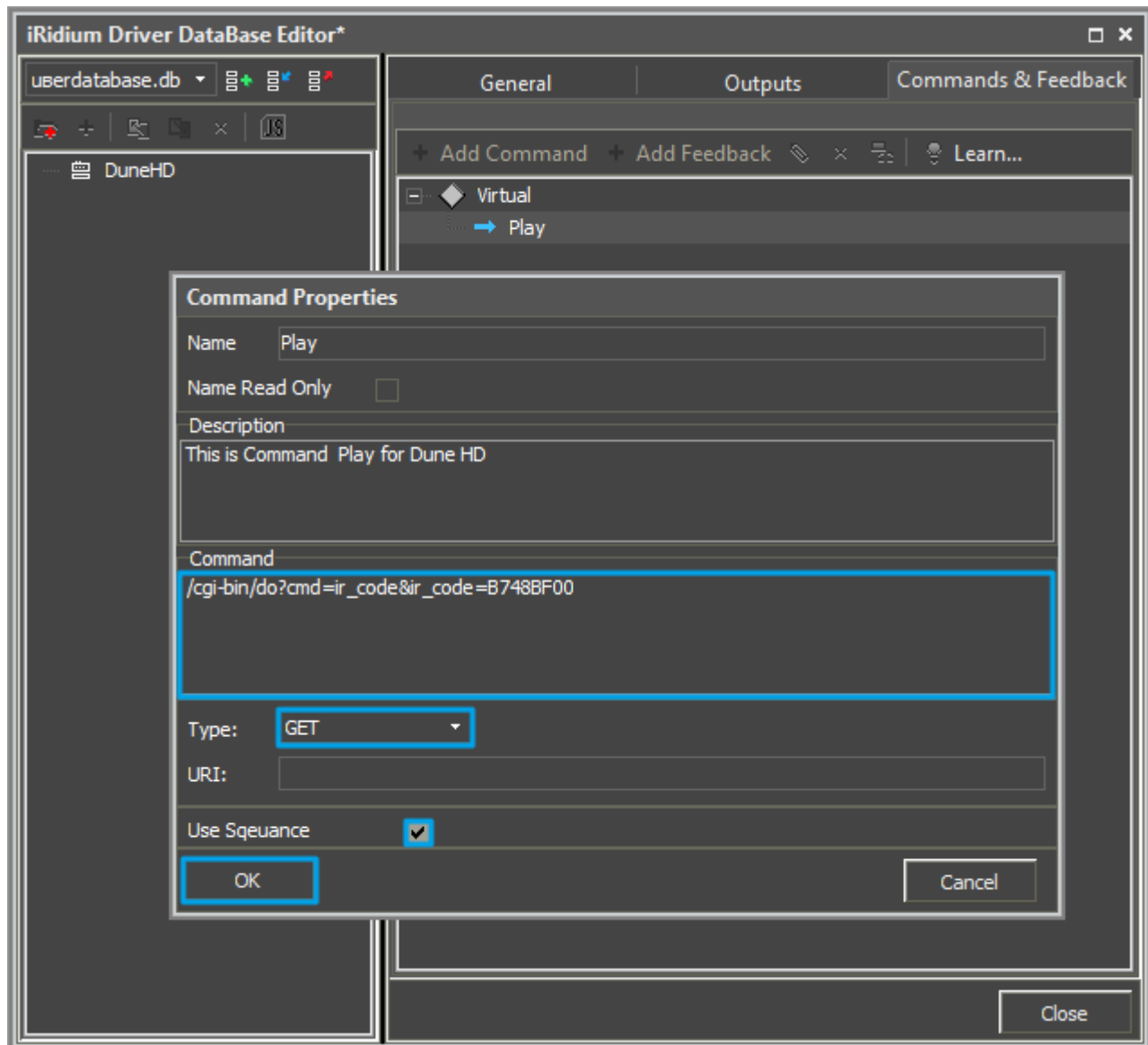
Adding commands in the list of devices

Go to the Commands & Feedback tab and use tools of iRidium GUI Editor:



The command is empty by default. The instruction the command should send is indicated in the

Command Properties window. Click two times on the created command to open the **Command Properties** window.



- Write the instruction for the command in the **Command** field of the opened window. In our case for the **Play** command there is the `'/cgi-bin/do?cmd=ir_code&ir_code=B748BF00'` instruction – referring to the device and the command NEC code .
- Mark the **Name Read Only** field if you do not want to change the command name in the future (it is used when the driver works with scripts which refer to the command identifying it by its name).
- In the **Descriptions** field indicate the command description.

The command can be checked in the browser window, it look as follows as a GET-request:

`http://192.168.0.100/cgi-bin/do?cmd=B748BF00`

When all commands are added, you can close the **DB Editor** window.

Switching Between the Local and Internet Connection

If the control panel should hold the connection with the system when the panel is out of the limits of the Wi-Fi network of the system, you need to set up the switch between the Internet and the local network.

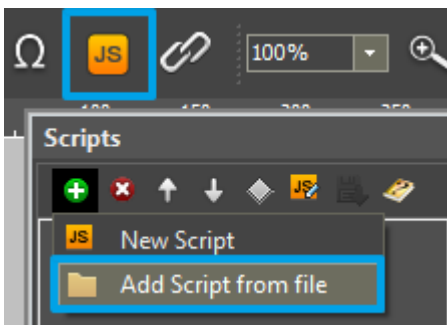
The remote mode suggests connection to the system via the Internet. At that the external IP-address or the domain name of the system to which you need to connect have to be used.

⚠ In iRidium Wi-Fi/3G CANNOT be switched automatically. For switching between the Internet and the local network you need buttons with special settings. See the settings below.

⚠ To control the system remotely you have to open the system for external access - to set up [the Port Forwarding Service](#).

⚠ To secure equipment from unauthorized access we recommend using secure connection with the remote system (VPN).

Setting up of the switch Wi-Fi/3G in iRidium projects:



1. Open the script editor in iRidium GUI Editor.
2. Download and add into your project the template of the Wi-Fi/3G switch (Add Script from file): [download the template of the Wi-Fi/3G switch](#)

The Wi-Fi/3G switch is performed with the help of the script function [SetParameters](#)

Setting up of parameters of the Wi-Fi/3G switch:

```
function Internal_1() // Function name  
  
{
```

```
IR.GetDevice('AV & Custom Systems (HTTP)').SetParameters({Host:
"192.168.0.21",Port: "12004",SSL: "1",Login: "admin",Password: "12345"}); // Driver
Name + Parameters
```

```
}
```

```
function External_1()
```

```
{
```

```
IR.GetDevice('AV & Custom Systems (HTTP)').SetParameters({Host:
"192.168.0.21",Port: "12004",SSL: "1",Login: "admin",Password: "12345"});
```

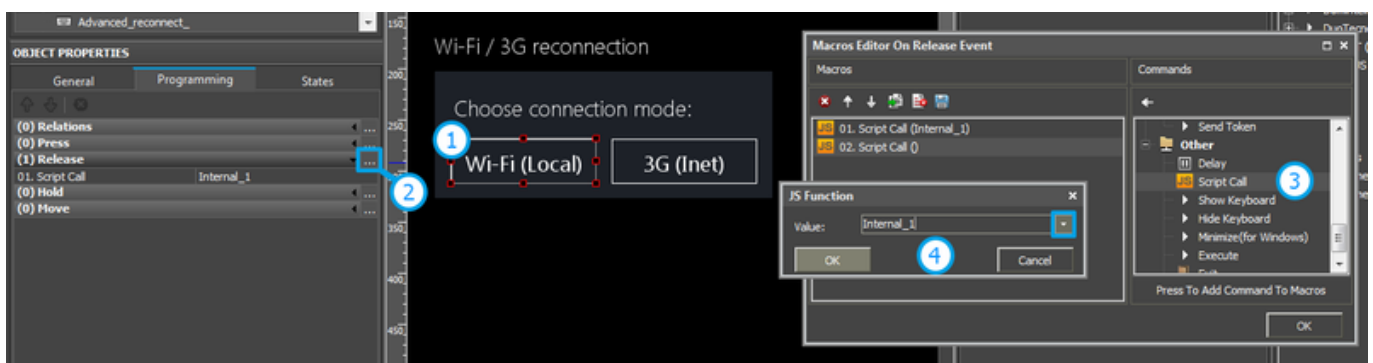
```
}
```

Indicate in the command settings:

- Function name - the name of the switch function (command). Two functions cannot have the same name in a project.
- Driver Name - the name of the driver which parameters are changed
- Parameters - the set of the switch parameters which you need to apply to the driver

Assign commands to buttons:

1. Select the button which will be responsible for the Wi-Fi/3G switch.
Open the properties of the button: Object Properties > Programming
2. Open [Macros Editor](#) of the button for the Press or Release events
3. Select the **Script Call** command and add it by double-clicking on it
4. Select the name of the function you want to activate in the drop-down list. Create the command.



Set up access to the equipment from the Internet:

In order to do that [open the equipment ports for remote access](#).

[Download the example of the Wi-Fi/3G switch \(project\) >>](#)

[↑ Back](#)

Downloads

[Example of switching Wi-Fi/Internet for AV & Custom Systems](#)

[↑ Back](#)