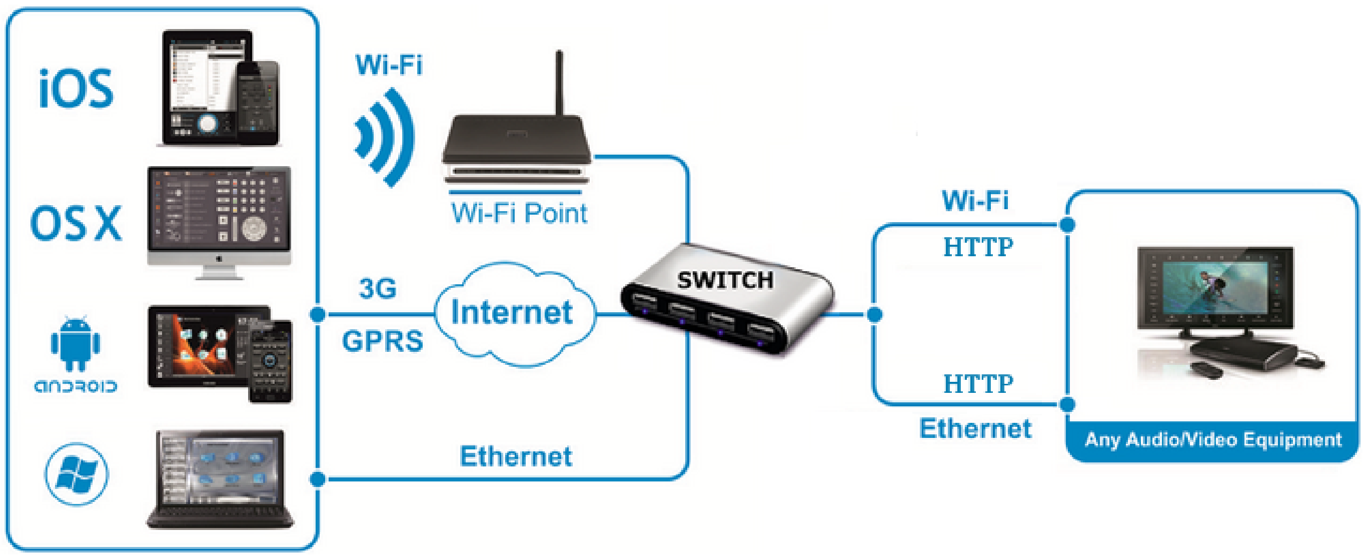
<< back (AV & Custom Systems: Working with Equipment via HTTP)

This article presents control of AV equipment or other systems via HTTP through the example of the DuneHD device.

Contents

Principles of Controlling Equipment via HTTP

To control Smart Home systems via HTTP commands (Ethernet or Wi-Fi) iRidium uses the freely customizable "AV & Custom Systems" (HTTP) which enables iRidium to send commands via HTTP (the commands are sent in the GET and POST formats). Thus the scheme of communication between an iRidium control panel and controlled equipment looks as follows:



Commands can be sent to HTTP devices in the ASCII format. When sending commands use special iRidium syntax to define the data format.

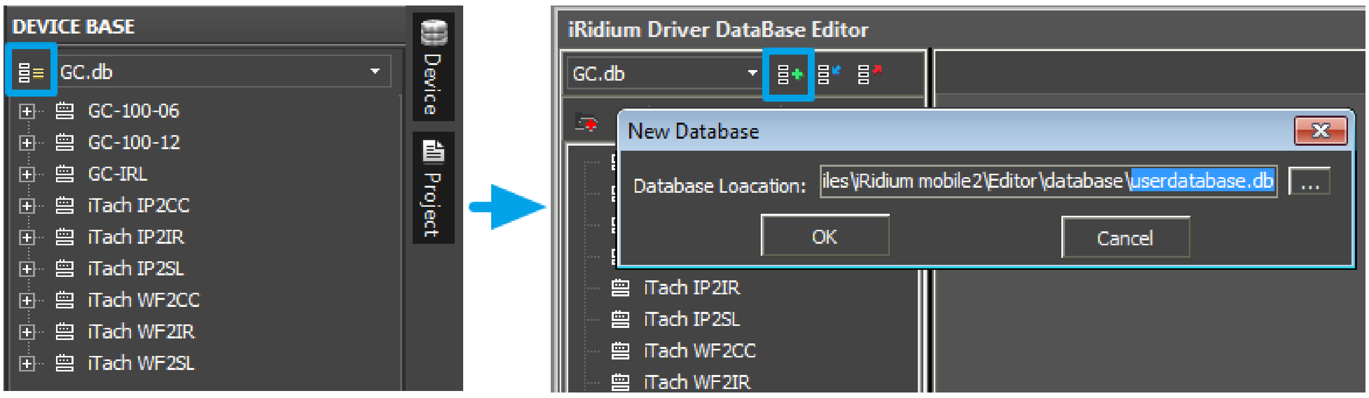
↑ Back

Adding HTTP Commands to iRidium Data Base

To store HTTP controlled devices with commands it is required to create their data base in iRidium GUI Editor. It will enable you to use the devices in any iRidium project.

DB Editor is used for adding devices controlled via TCP and their commands to iRidium base. New devices can be added only to custom data bases as standard bases contain ready preconfigured modules of devices which cannot be edited (you can only take modules from the ready bases).

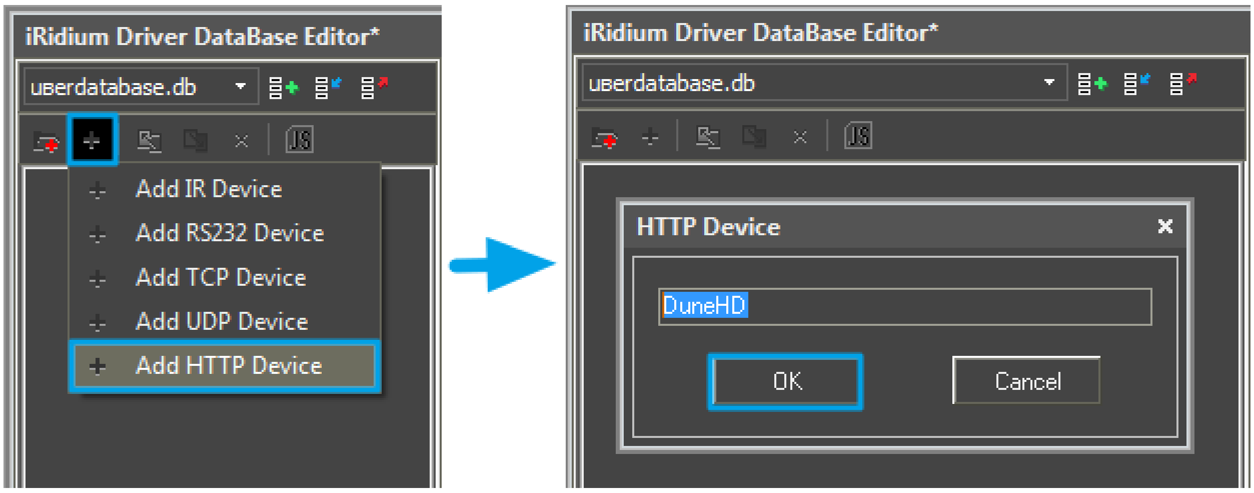
Creating New Files of iRidium Data Base



Database Location: iles\iRidium mobile2\Editor\database\userdatabase.db

Use your own data bases (created manually) for storing your devices not standard data bases which can be updated with releases of new versions of iRidium GUI Editor. The names of the base file and folder for storing are indicated when creating the file.

Creating New HTTP Devices

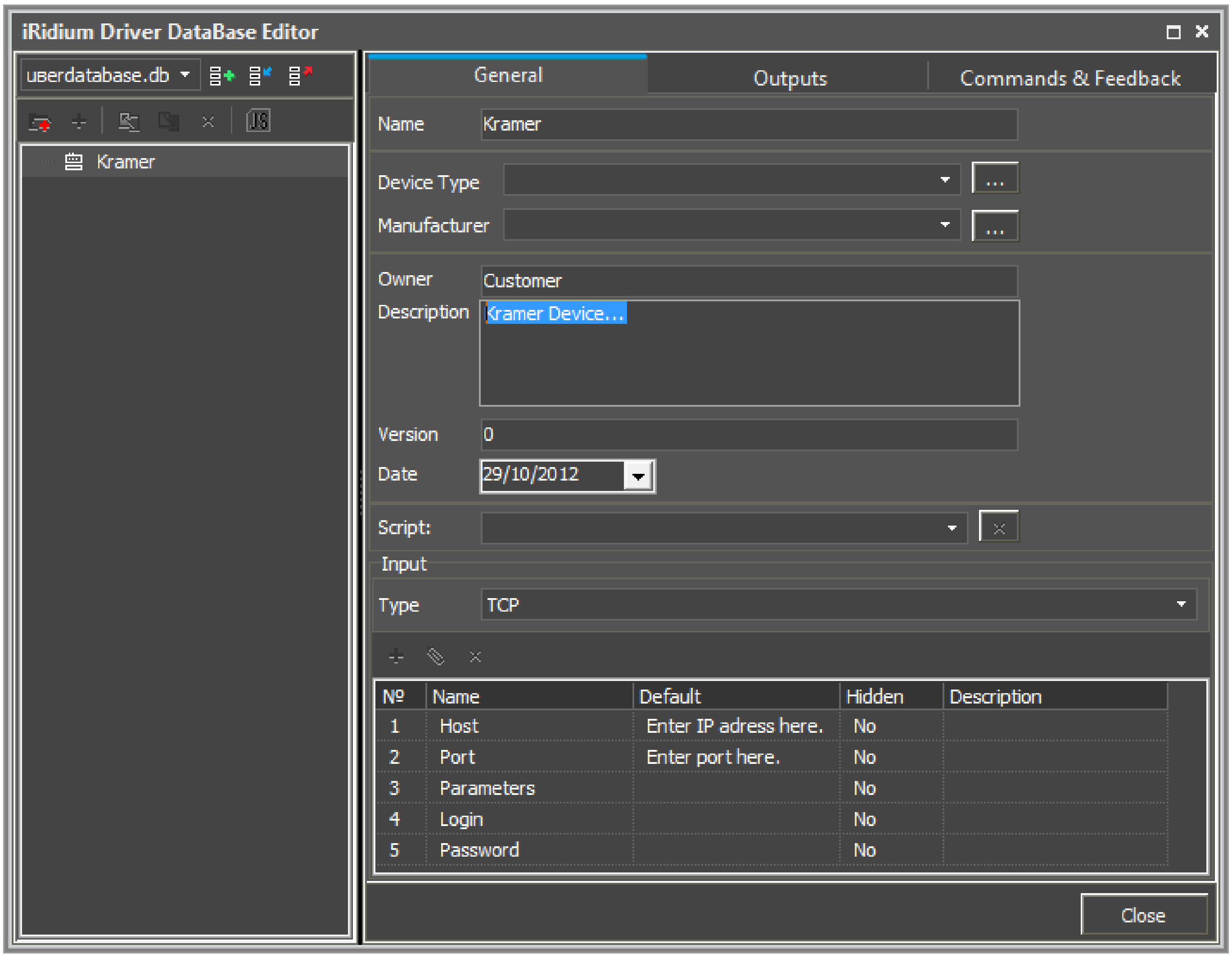


DuneHD

Tools for creating new devices in iRidium data base are available in the right-click menu. The type of the created device - HTTP (HTTP Device) – defines its characteristics. For HTTP devices you can indicate information about the manufacturer, type of the controlled equipment, comments and commands.

Setting Up HTTP Devices

Creation of description for new devices is not compulsory but it can help with their further identification



Name	Kramer
Device Type	
Manufacturer	
Owner	Customer
Description	Kramer Device...
Version	0
Date	29/10/2012
Script:	

Input

Type	TCP

№	Name	Default	Hidden	Description
1	Host	Enter IP adress here.	No	
2	Port	Enter port here.	No	
3	Parameters		No	
4	Login		No	
5	Password		No	

- **Device Type** – a 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

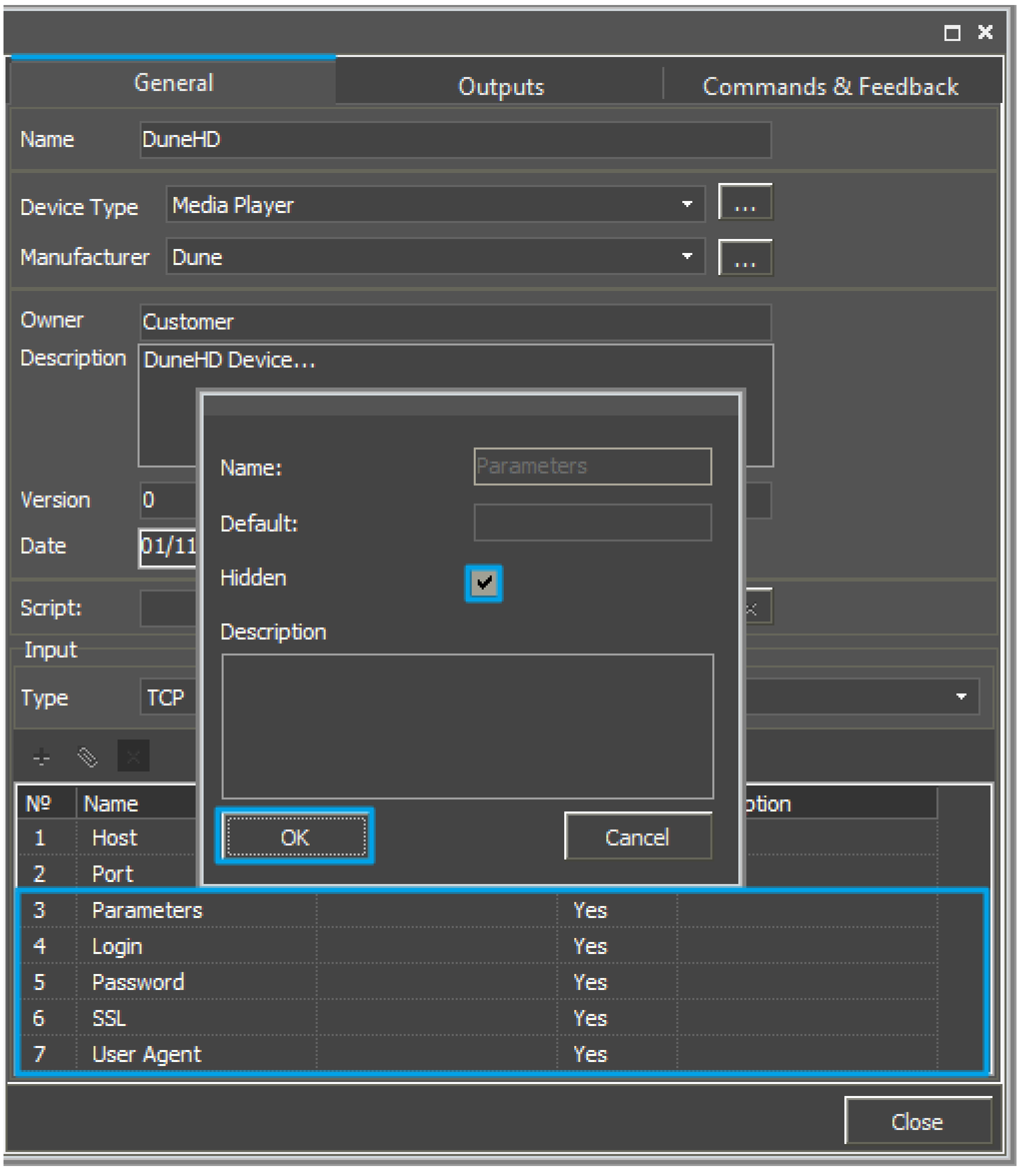
HTTP devices have the following properties for connection:

- **Host** – an IP-address of the controlled Kramer device in your network.
- **Port** – a port the device works through.

Optional properties:

- **Parameters** – additional properties for connection
- **Login** – a login for connection
- **Password** - a password for connection
- **SSL** - a cryptographic protocol which enables safe connection between the device and the server.
- **User Agent** – a part of the HTTP request usually including such information as application name and version, PC OS and language.

These optional properties are used seldom and if you are not going to use them it is recommended to hide them: click two times on each property to open the window for settings and select **Hidden**.


General | Outputs | Commands & Feedback

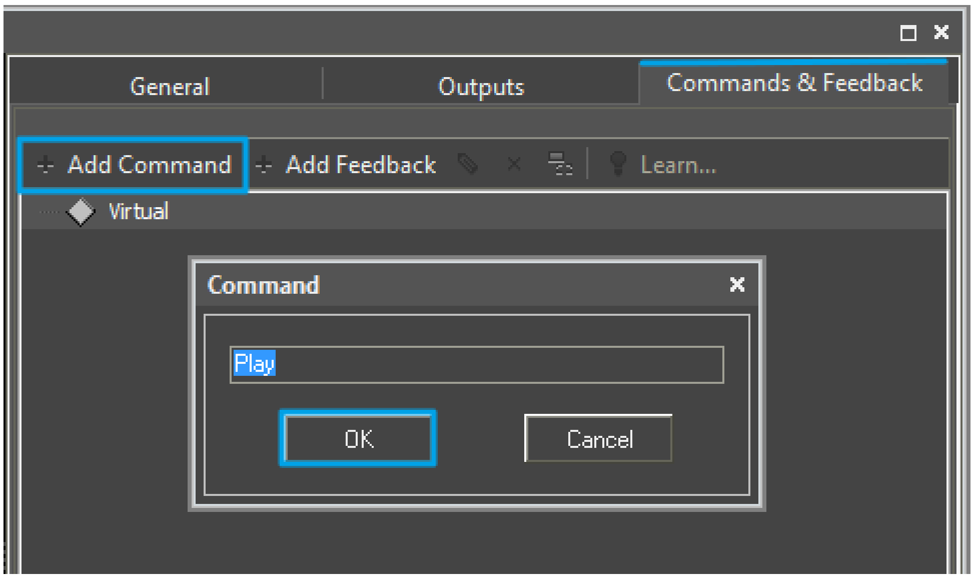
Name: DuneHD
Device Type: Media Player
Manufacturer: Dune
Owner: Customer
Description: DuneHD Device...
Version: 0
Date: 01/11
Script:
Input
Type: TCP

Name: Parameters
Default:
Hidden: ✓
Description:
OK | Cancel

Nº	Name		
1	Host		
2	Port		
3	Parameters	Yes	
4	Login	Yes	
5	Password	Yes	
6	SSL	Yes	
7	User Agent	Yes	

Close

To add the command in the list of the created device go to the Commands & Feedback tab and use tools of the Editor base:

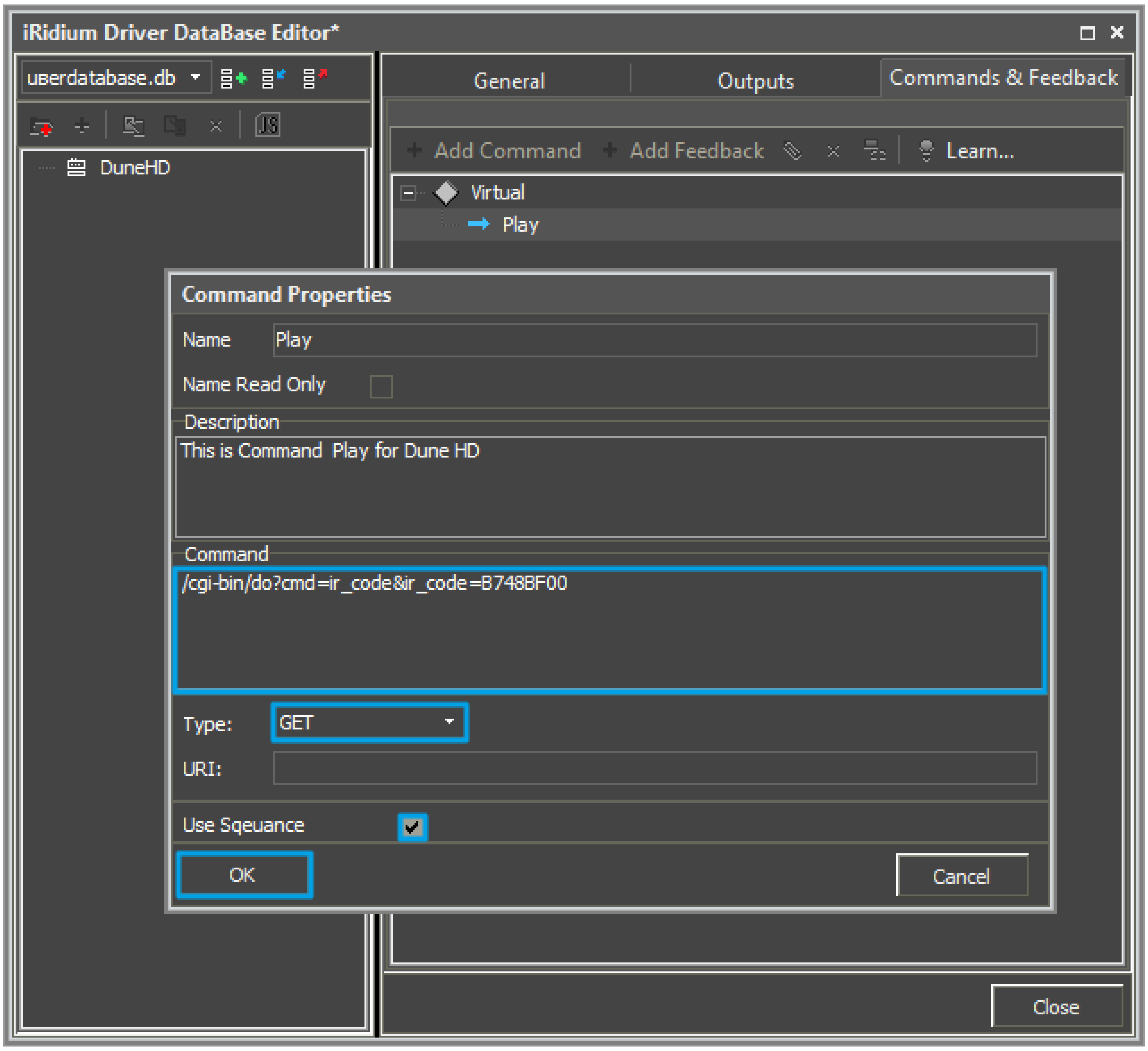

General | Outputs | Commands & Feedback

Add Command | Add Feedback | Learn...

Virtual

Command
Play
OK | Cancel

The command is created empty by default. The instruction the command should send is indicated in the **Command Properties** windows. Click two times on the created command to open the **Command Properties** window.



Command Properties

Name: Play

Name Read Only: ☐

Description: This is Command Play for Dune HD

Command: /cgi-bin/do?cmd=ir_code&ir_code=B748BF00

Type: GET

URI:

Use Sqeuance: ✓

OK   Cancel

- 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, where:

- **/cgi-bin/do?cmd=ir_code&ir_code** – reference to the device
- **B748BF00** - NEC command code (the command is sent in the ASCII format).

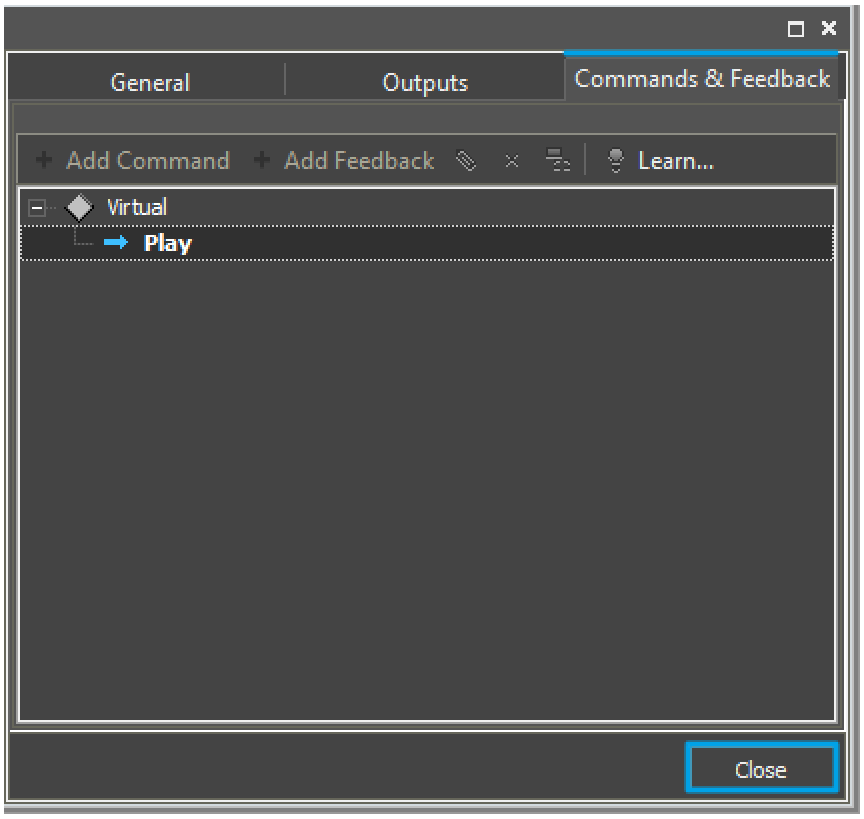
At that the command can be checked through the browser window, as a GET-request it looks as follows:

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

Nec codes can be found here</a>

- Select the **GET** method in the Type field.
  - GET – a request in one string
  - POST – sending a block of data

- 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.



- Now you can close the **DB Editor** window.

↑ Back

Syntax of HTTP Commands

Commands for sending via the HTTP protocol cannot be added to DB Editor without checking them for compliance with the syntax accepted in iRidium. It allows you to identify the command data format and form the command correctly so it could be read by the controlled equipment.

The syntax of data sending via HTTP (GET and POST) defines the rules for sending data in the ASCII format.

Sending GET Requests

When working with the AV & Custom Systems driver via HTTP GET requests are formed with the following view:

```
http://<ip address></span>:<port></span>/<command_body></span>
```

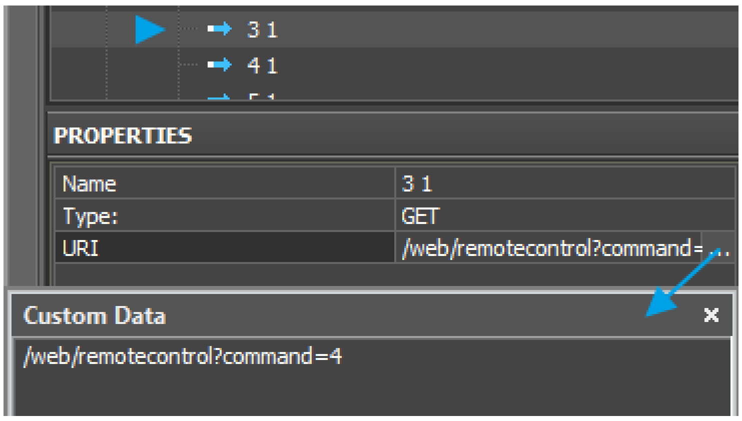
To form a GET request in iRidium indicate the body of the GET request with the first slash in the Data window of the data sending channel. For example:

```
http://192.168.0.100</span>:81</span>/command</span>
```

- Red color indicates the part of the GET request which should be written in the Data window.
- IP-address and port of connection to the device controlled via the web-interface are indicated in the driver properties.

The string in the **Data** window looks as follows:

`/command</b></span>`



Name	3 1
Type:	GET
URI	/web/remotecontrol?command=...

/web/remotecontrol?command=4

↑ Back

Sending POST Requests

A POST request for sending via the AV & Custom Systems (HTTP) driver should be divided in 2 parts: **URI** (head) и **Data** (body).

The head of the request (URI) is the part of the hyper link which defines the directory the request is addressed for. It starts with the back slash and can contain any kind of data. The head is in the upper part of the POST request.

Further there is system information which should not be duplicated in iRidium.

The request body (Data) is separated from it. The separation is made by double combination of two symbols – the symbol of the string end and the symbol of carriage shift: 0D,0A,0D,0A. In the editor it will look as an empty string between request strings. There can be any data in the request head and body. **The request body should be written in 'single quotes' when you add it in the Data field!**

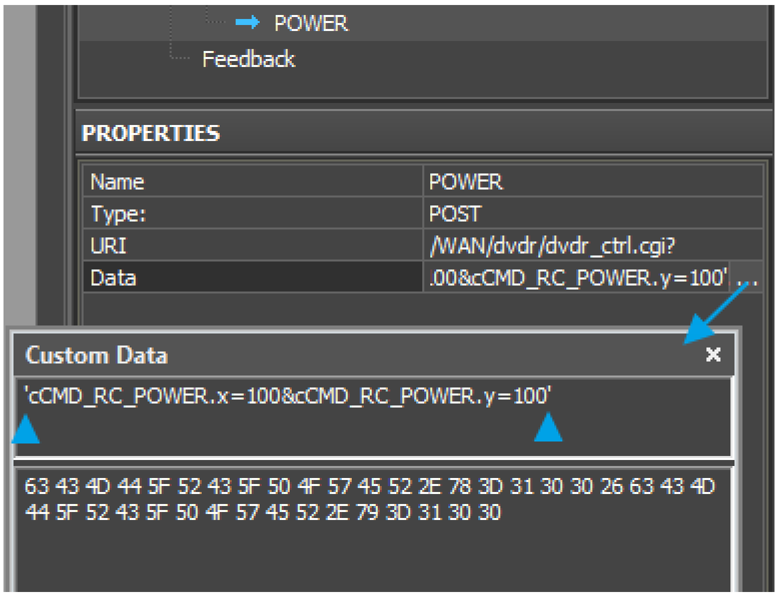
Example 1:

```
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"><System><Power_Control><Power>Standby</Power></Power_Control></Syst
em></YAMAHA_AV>
```

Example 2:

```
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
```

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


PROPERTIES

Name	POWER
Type:	POST
URI	/WAN/dvdr/dvdr_ctrl.cgi?
Data	.00&cCMD_RC_POWER.y=100' ...

Custom Data

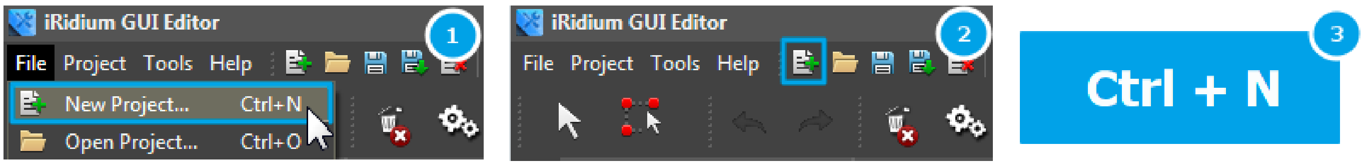
```
'cCMD_RC_POWER.x=100&cCMD_RC_POWER.y=100'
```

```
63 43 4D 44 5F 52 43 5F 50 4F 57 45 52 2E 78 3D 31 30 30 26 63 43 4D
44 5F 52 43 5F 50 4F 57 45 52 2E 79 3D 31 30 30
```

↑ Back

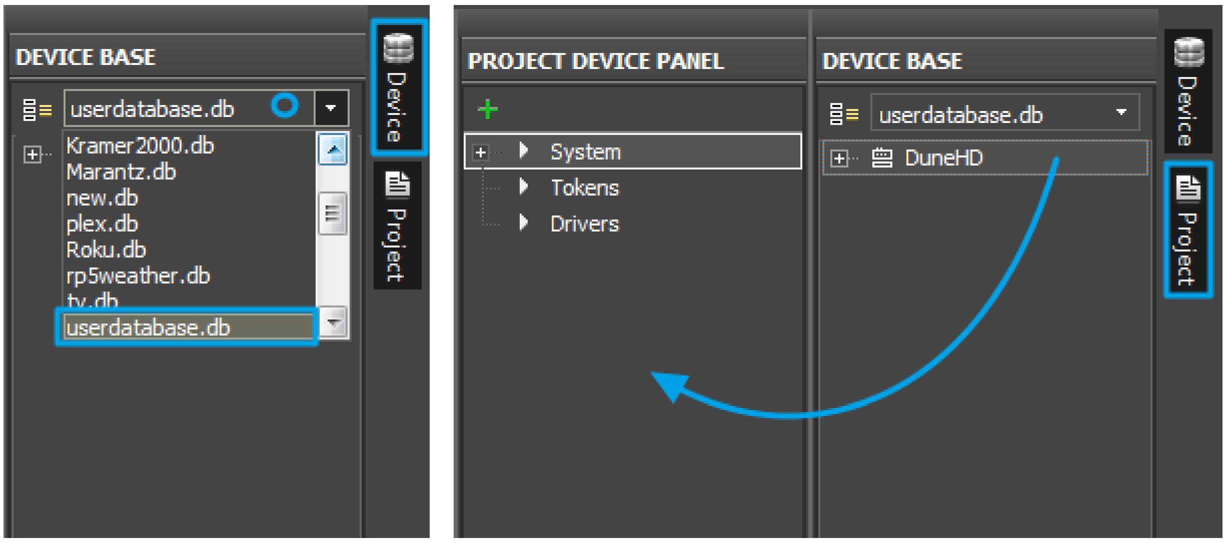
Adding HTTP Devices to iRidium Projects

Create a new project. New projects can be created with the help of the button on Tool Box or through the menu **File > New Project**.

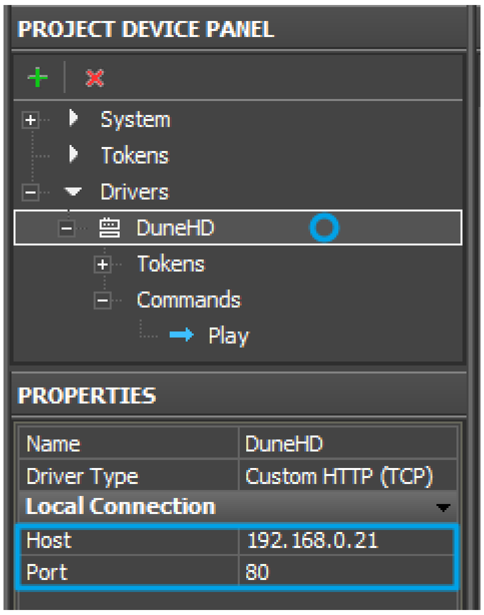


You can find more information about project creating in the section [Creating the Graphic Part of iRidium Projects]

Select the **userdatabase.db** data base in the **Device Base** panel and add the **DuneHD** device you have in **Project Device Panel** of your project using the **Drag&Drop** method.



To connect iRidium to **DuneHD**, indicate the local IP-address in general settings of the device in the PROPERTIES window of the Project Device Panel:



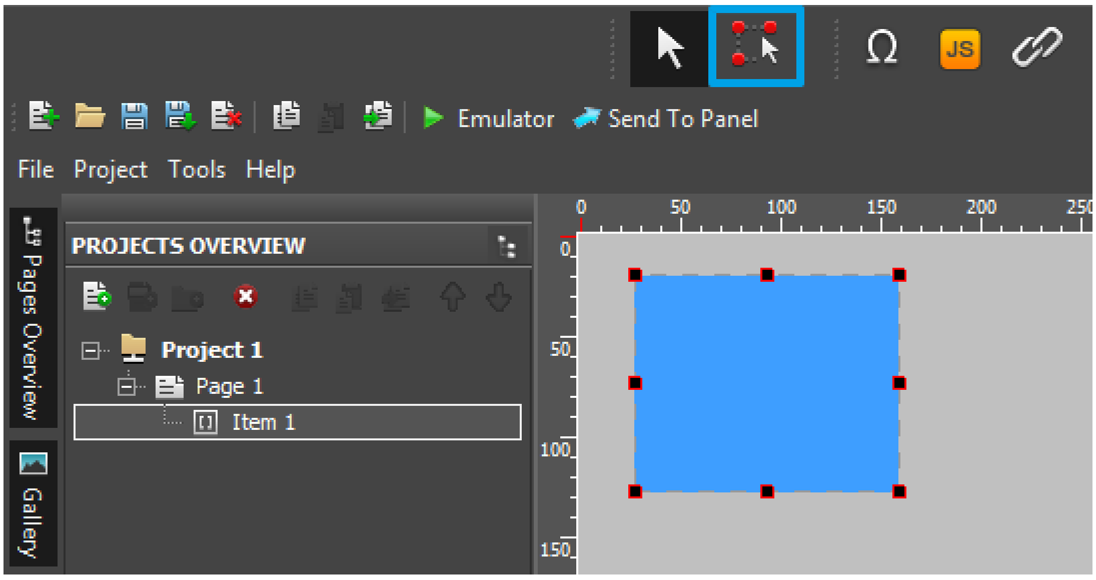
Name	DuneHD
Driver Type	Custom HTTP (TCP)
Local Connection	
Host	192.168.0.21
Port	80

- **Host** – an IP-address of the controlled DuneHD device.
- **Port** – a port the device works through.

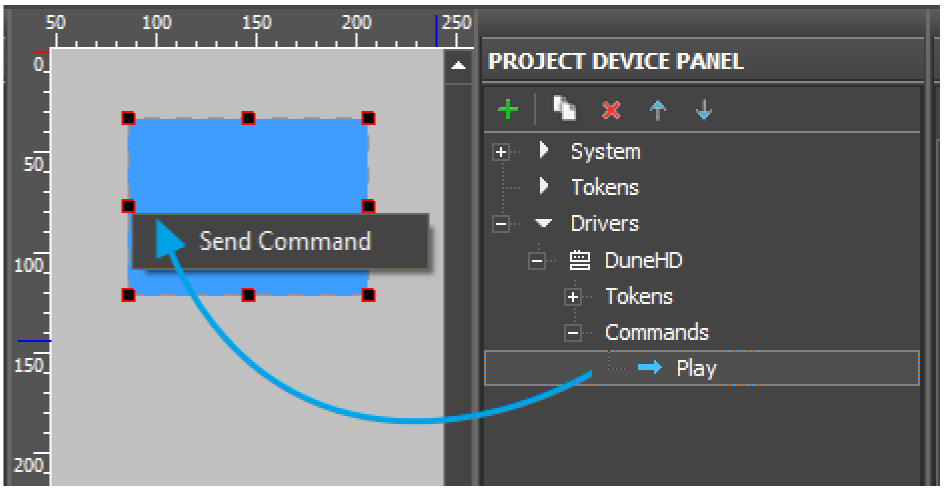
To work with the DuneHD device via the Internet indicate the public IP-address of the router the converter is connected to. Set up the Port Forwarding Service for the router which enables possibility of referring to the local address of the controlled converter from the Internet (remote control).

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

Select the **Draw Item** tool and create a new item.



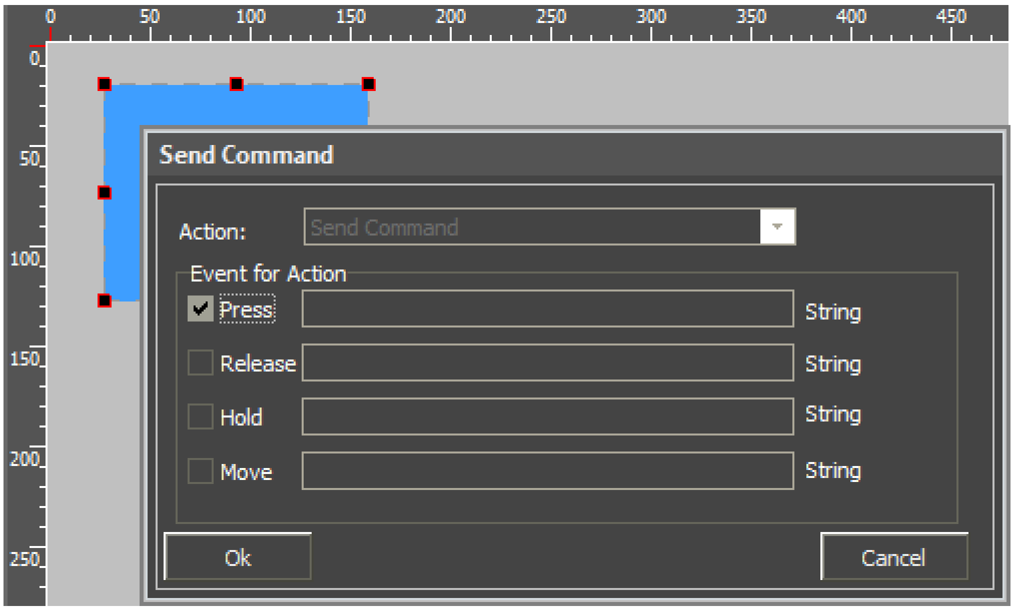
Open commands of the **DuneHD** device and drag the **Play** command on the item. Select **send command** in the appeared window.



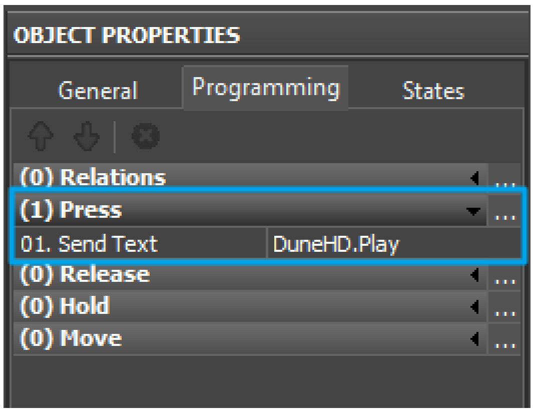
In the **Send Command** dialog window select the type of the event which will activate command sending to the device.

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

The **Press** event is indicated by default as the most commonly used.

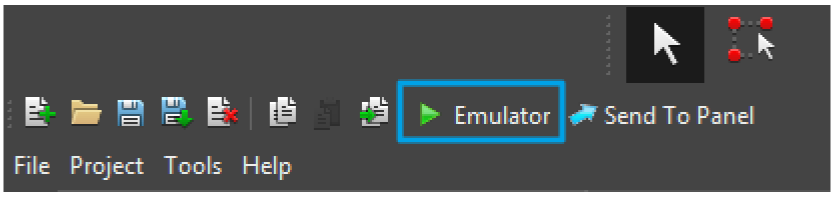


The commands added on the item can be seen in the **Programming** tab of the **Object Properties** panel.



01. Send Text	DuneHD.Play

Launch Emulator to check how the command works.



↑ Back

<< back (AV & Custom Systems: Working with Equipment via HTTP)