

## Device.Send()

the method can be used only with [AV & Custom Systems](#) drivers. It enables sending of random data (string, number, array)

### Syntax

```
Module.GetDevice(name).Send([data]) Module.GetDevice(name).Send(array)
```

input	sample	description
name	"Driver 1"	type: string the driver name in the iRidium project
data	a, b, c	type: string the list of variables or strings: instructions for sending
array	[a, b, c]	type: array the array of variables or strings: instructions for sending
<b>output</b>		
successful or not		type: Boolean successful

### Example:

```
// AV & Custom Systems (TCP, UDP, RS232)
Module.GetDevice("AV Device").Send(['instruction', '\r\n']); // send
instruction
// AV & Custom Systems (HTTP)
// <Type or request>,<URI>,<Data> divided by "," so u cannot use "," inside
of <URI> and <Data>
Module.GetDevice("AV Device").Send(['GET,URI']); // GET request
Module.GetDevice("AV Device").Send(['POST,URI,Data']); // POST request
Module.GetDevice("AV Device").Send(['PUT,URI,Data']); // PUT request
/* Global Cache:
Module.GetDevice("Global Cache").Send(['DATA'], <TRANSPORT ID>)
TRANSPORT ID:
0 = port 4998 (IR commands, system data, relays)
1 = port 4999 (Serial 1)
2 = port 5000 (Serial 2)
*/
Module.GetDevice("Global Cache").Send(['setstate,1:1,1', '\r\n'], )
```

**HDL, Domintell:** use the network name "HDL-BUS Pro Network (UDP)" / "Domintell Network (UDP)", not the name of a particular device at referring.

**HTTP:** when sending an HTTP command to the driver, indicate the method of sending first: GET, POST, PUT, then after the comma indicate the command URI and Data for POST and PUT. Inside URI and Data you cannot use commas. If your protocol uses commas in HTTP commands, send them via AV & Custom Systems (TCP), forming requests all over.

