

RedRat Hub – Windows, Linux & Mac

October 2015

For RedRat Hub Version 4.13

R E D R A T
● ● ● ● ● ●

Contents

1	Introduction	3
2	Architecture	3
3	Command Line Arguments and Switches	4
4	RedRat Hub Operation	4
4.1	Communication Protocol	4
4.2	irNetBox Connections	5
4.3	RedRat3 Connections.....	5
4.4	Access control	5
4.5	Client connections.....	5
4.6	Toggle and double signals.....	5
5	RedRat Hub Commands	5
5.1	Hub queries.....	6
5.2	Hardware queries.....	7
5.3	Signal transmission	7
5.4	Command Building Blocks.....	7
5.4.1	Identifying RedRat Hardware [Identifier]	7
5.4.2	Output Descriptor	8
5.4.3	Repeat descriptor.....	8
5.4.4	Using Files	8
5.5	Configuration	8
6	Examples	9
6.1	Getting a list of RedRat hardware.....	9
6.2	Manually adding an irNetBox.....	9
6.3	Saving configuration	9
6.4	Blacklisting an irNetBox	9
6.5	Listing signals for a device.....	9
6.6	Retrieving firmware information	9
6.7	Signal transmission	9

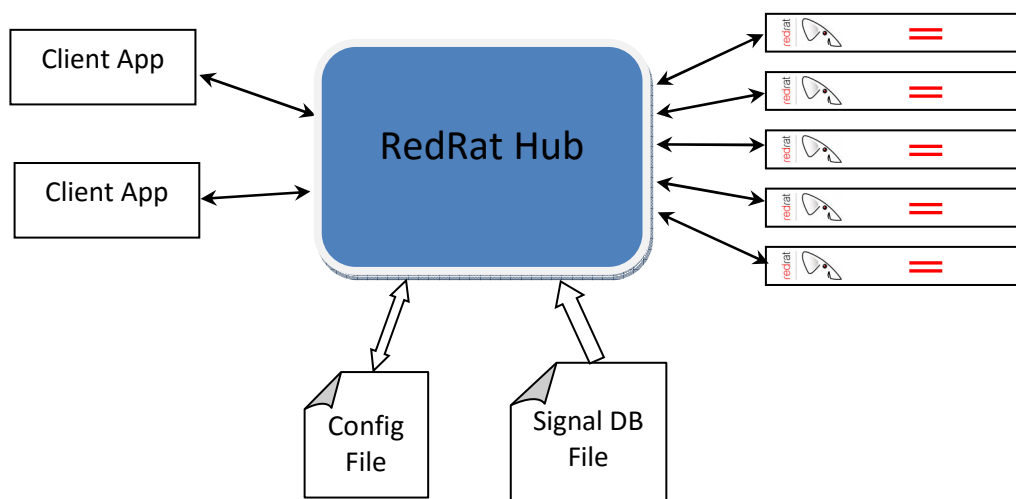
1 Introduction

The RedRat Hub is an application that manages communication with RedRat hardware. Normally RedRat devices are controlled directly via applications provided by RedRat Ltd (e.g. Scheduler or TestManager), third parties or by software developed using the RedRat .NET based SDK. Sometimes these alternatives are not suitable, so RedRat Hub has been developed to provide additional flexibility:

1. If your software or system does not integrate easily or well with .NET, then using RedRat Hub allows control using simple commands sent via a TCP/IP socket.
2. If you need multiple applications to communicate with an irNetBox, the Hub can act as a multiplexer.
3. If you would like to control a USB RedRat3 device from a remote computer, RedRat Hub allows connections via a network.

RedRat Hub is based on the core RedRat SDK .NET code, but will run on both Windows .NET CLR (V4.5) and also on Linux using Mono.

2 Architecture



Client applications connect to RedRat Hub on a socket and send commands (as UTF-8 strings), which are then interpreted and the appropriate instruction sent on to an irNetBox. The Hub can take connections from multiple clients and ensures that they are handled appropriately for the irNetBox hardware, for example with an irNetBox-II, the complete output operation sequence has to be completed before another operation can be initiated whereas with an irNetBox-III, multiple operations can be performed simultaneously where there is no overlap in output port usage.

When a command is sent to RedRat Hub, that connection will effectively wait until the operation has been completed before returning either success or failure. This allows the client application to step through a set of operations at the rate at which they are being executed. If a client application needs to send multiple commands at the same time, it can then open multiple connections to issue these commands.

On start-up, the Hub loads the IR signal data file which is then used as the IR data source. It can also optionally load a configuration file, which is can be saved and re-run at runtime by issuing the appropriate commands.

3 Command Line Arguments and Switches

```
Usage: RedRatHubCmd <signal data file 1> [<file 2> ...] [options]
```

```
<signal data file 1> : Name of XML or TXT signal data filename.
```

```
At least one file must be given.
```

```
Options:
```

```
--quiet      : Prints out very few details of operation.
--verbose    : Prints out verbose details of operation.
--debug      : Prints out even more information.
--port       : The port to use for client connections (default = 40000).
--conns      : The number of required client connections (default = 20).
--noscan     : Do not search for RedRat hardware once started.
--config=file : Load configuration from file.
```

At least one signal data file must be specified, either in device database format (xml) or irNetBox format (txt). When loading irNetBox signal files the dataset name is taken to be the file name without its extension, so signals within *stb.txt* would be referred to with the dataset *stb*.

The *--quiet*, *--verbose* and *--debug* switches may optionally be provided to select more or less program output respectively.

If many client applications are to be used, for example one client per IR output port on many irNetBoxes, then the *--conns* switch is used to increase the number of simultaneous client connections that can be handled. If the number of client connections exceeds this value, then client connections will be queued and serviced once a connection becomes available.

By default the hub will scan for RedRat devices once it has started. This can be disabled using the *--noscan* switch.

Configuration may be loaded from a file once the hub has started using the *--config="file"* switch.

4 RedRat Hub Operation

4.1 Communication Protocol

The hub uses a TCP socket interface to accept commands on port 40000 by default. Commands are sent to the hub in plain text as UTF-8 strings. The general format is a list of one or more *key="value"* statements separated by spaces and terminated with a line feed (\n) character. Statements may be in any order and are not case sensitive. Multiple commands may be sent through a single connection or through multiple connections from different clients.

The hub responds to each command it receives and clients should always wait for this response to avoid sending commands faster than the hub can process them. Short responses are simply text terminated with line feed, for example `OK\n`. Responses that span multiple lines are enclosed by braces. For example:

```
{\n  line 1\n  line 2\n}
```

4.2 irNetBox Connections

The hub makes a connection to an irNetBox only when needed, typically once the first signal transmission command has been received. Once a connection has been established it is held open until the hub closes or a disconnection command is received. See commands section for more details.

4.3 RedRat3 Connections

RedRat3s are USB devices and can be used by multiple client applications simultaneously, so the Hub may share control with any other applications on the Windows computer.

Control of USB RedRat3 devices is currently only supported on Windows and not on Linux.

4.4 Access control

By default the hub may connect to any RedRat device that it discovers. A RedRat device can be blacklisted to prevent the hub from connecting to it, which might be useful if the device is under the dedicated control of some other program that must not be interrupted. Blacklisting and whitelisting some or all RedRat devices can be done using hub queries, detailed below.

4.5 Client connections

Clients connect to the hub via a socket interface on TCP port 40000 (by default) and are free to disconnect any time thereafter. The hub will only disconnect clients when it closes.

4.6 Toggle and double signals

The hub automatically alternates transmission of toggle signals and double signals. Toggle state is managed by the hub on a per-signal and per-output basis so that each AV device will see the correct alternation of signals. In this context, however, per-signal refers to the complete signal data rather than just the signal name. Thus, a signal with a repeat descriptor is a *different* signal to the original. Equivalently, a signal with a repeat count of 2 is different to one with a repeat count of 3 and so on.

5 RedRat Hub Commands

There are three types of command sent to RedRat Hub:

1. Hub queries, which allow interrogation of the hub's state or instruct it to perform some action.
2. RedRat hardware queries, which are used to return information about a particular RedRat device.
3. Instructions to output IR signals via RedRat hardware.

5.1 Hub queries

Information about the state of the hub can be retrieved and modified through hub queries. The format of a hub query is simply:

```
hubQuery="query text"
```

or

```
hq="query text"
```

where the query text is one of the following:

- *hub version* – returns the software version of the hub
- *list redrats* – returns the list of RedRat devices as of the last scan
- *find redrats* – performs a search for RedRat devices and returns the result. Existing connections are preserved.
- *forget redrats* – disconnect from any connected devices and clear list
- *whitelist all irnetboxes* – allow the hub to connect to any irNetBox it knows of
- *blacklist all irnetboxes* – the hub should not connect to any irNetBox. Existing connections are closed
- *whitelist all redrat3s* – allow the hub to connect to any RedRat3 device it can see
- *blacklist all redrats3s* – the hub should not connect to any RedRat3 device
- *whitelist redrat* – allow the hub to connect to the given RedRat device. Requires that the RedRat device is identified with *[identifier]* – see section 5.4.1.
- *blacklist redrat* – prevent the hub from connecting to the given irNetBox. Existing connection, if any, is closed. Requires *[identifier]* – see section 5.4.1.
- *list datasets* – returns the list of IR signal datasets loaded from the signal files.
- *list signals in dataset* – returns the list of signal names for the given dataset. Requires *dataset* (see Signal transmission)
- *add ir-file* – adds a new IR dataset file (XML or TXT) to the IR data used by the application. If there are datasets with the same name as existing datasets, then old data will be overwritten. Requires the *file* parameter. See section Configuration for more information.
- *load ir-file* – Loads a new IR dataset file, replacing (and deleting) any IR dataset information known by the application. Requires the *file* parameter. See section Configuration for more information.
- *add irnetbox* – add an irNetBox to the list of devices known to the hub. Requires that the device is identified with *[identifier]*.
- *remove irnetbox* – remove an irNetBox from the list of devices known to the hub. Requires that the device is identified with *[identifier]*.
- *connect irnetbox* – forces RedRatHub to connect to the given irNetBox. It usually only connects on demand.
- *disconnect irnetbox* – closes the connection to the given irNetBox. The connection will be re-established on demand, or can be forced with the *connect irnetbox* command.
- *save config* – write the current state of the hub to a file. Requires the *file* parameter. See section Configuration for more information.
- *load config* – configure the hub by running commands in a file. Requires the *file* parameter. See section Configuration for more information.

5.2 Hardware queries

The hub can retrieve information about RedRat hardware. The general format is:

```
hardwareQuery="query text" [identifier]
```

where *hardware query* may be one of the following:

- *firmware version* – returns the version of the firmware running on the given irNetBox
- *hardware type* – returns the hardware type of the given RedRat device. For example, an irNetBox-III would return *RedRat4_III_IP*.
- *reset* – resets the main microcontroller, but does not reset the XPort module. This is only supported on irNetBox-III devices.
- *reset xport* – resets the Lantronix XPort network module. The hub then waits for a few seconds to give it time to restart before returning a response. This is only supported on irNetBoxes.

5.3 Signal transmission

Signal transmission commands are hardware agnostic and return *OK\n* as soon as transmission has been completed. This makes it easy to maintain synchronisation between code and hardware. A signal transmission command has the form:

```
[identifier] dataset="dataset name" signal="signal name" output="[output descriptor]"
```

- *[identifier]* – identifies the RedRat device to use for IR signal output – see section 5.4.1
- *dataset name* – the name of the dataset to use for this *signal*
- *signal name* – the signal to transmit
- *[output descriptor]* – which output(s) and power level(s) to transmit the signal on when using an irNetBox – see section 5.4.2
- *[repeat descriptor]* – how to repeat the signal, if at all, described in section 5.4.3

An example of a full output command is:

```
ip="192.168.1.40" dataset="Sky+" signal="play" output="3"
```

5.4 Command Building Blocks

For the three different classes of command given in the previous sections, there are some common building blocks:

5.4.1 Identifying RedRat Hardware [Identifier]

A RedRat device can be identified by its name (RedRat3 & irNetBox), MAC address (irNetBox), IP address (irNetBox) or host name (irNetBox), whichever is most convenient. An *identifier* is exactly one of these properties (specifying more than one is redundant at best and contradictory at worst). Wherever *[identifier]* is required, substitute one of the following:

- *name="my redrat"* e.g. *name="RR3-MKII"*
- *mac="mac address"* e.g. *mac="00-20-4a-11-22-33"* (note the use of dashes rather than colons)
- *ip="ip address"* e.g. *ip="192.168.1.33"*

- `host="my-irNetBox.my-domain"`

5.4.2 Output Descriptor

An output descriptor specifies one or more outputs and, optionally, output power, to use when transmitting a signal. Examples of the syntax follow.

- `output="3"` – Just output 3 at default 33% power
- `output="3-8,16"` – Outputs 3, 4, 5, 6, 7, 8 and 16 (at 33%)
- `output="1:40,10-12:80"` – Output 1 at 40% and outputs 10, 11, 12 at 80%

It is important to note that different versions of the irNetBox have different capabilities when setting output power. The mark II has 3 output power levels (low, medium and high) whereas the mark III is continually adjustable. When using older hardware the hub will choose an output power level closest to the one requested. Requesting 40% power will result in 40% on a mark III but only 33% on a mark II. The default output power is 33%.

5.4.3 Repeat descriptor

A repeat descriptor may be used to override the repeat information contained within certain signals. This descriptor is optional and may only be used with modulated signals and double signals. One of the following should be provided:

- `repeats="n"` – set the repeat count for this signal to the integer *n*, not larger than 255.
- `duration="T"` – calculate the repeat count such that the signal is transmitted for *T* milliseconds. An error is returned if the calculated repeat count exceeds 255.

5.4.4 Using Files

When loading IR datasets, or working with configuration information, file names need to be given. The form is:

`file="filename"`

where the *filename* may need to be the full path if not in the directory in which RedRatHubCmd.exe was started.

5.5 Configuration

The hub can be configured by running a sequence of commands from a file. Any of the commands detailed herein may be used and the format is exactly the same. When the `save config` hub query is invoked, the hub will write out the sequence of commands required to restore any irNetBox devices added manually, including whether or not these devices have been blacklisted. Devices discovered automatically, e.g. via the `find redrats` hub query, are not saved. As automatic generation of config files is quite limited, users are encouraged to review and modify the contents of config files to fit their requirements.

Each line of the config file should contain a single command. Lines beginning with # are treated as comments and ignored.

Configuration may be loaded when the hub is started using the `--config="file"` command line option and can be (re)loaded at any time with the `load config` hub query.

6 Examples

6.1 Getting a list of RedRat hardware

```
hubquery="list redrats"
```

```
[irNetBox-III] No name 00-20-4A-D0-16-D3 (00-20-4A-D0-16-D3) at 192.168.1.49
```

```
[irNetBox-III] No name 00-20-4A-D3-1F-32 (00-20-4A-D3-1F-32) at 192.168.1.104
```

6.2 Manually adding an irNetBox

```
hubquery=" add irnetbox" host="my-irNetBox.my-domain"
```

```
[irNetBox-III] No name 00-20-4A-D0-16-D3 (00-20-4A-D0-16-D3) at 192.168.1.49
```

6.3 Saving configuration

```
hubquery=" save config" file="config.txt"
```

```
OK
```

6.4 Blacklisting an irNetBox

```
hubquery="blacklist irnetbox" ip="192.168.1.49"
```

```
[irNetBox-III] No name 00-20-4A-D0-16-D3 (00-20-4A-D0-16-D3) at 192.168.1.49 (blacklisted)
```

```
[irNetBox-III] No name 00-20-4A-D3-1F-32 (00-20-4A-D3-1F-32) at 192.168.1.104
```

6.5 Listing signals for a device

```
hubquery="list signals in dataset" dataset="sky+"
```

```
Pause
```

```
Play
```

```
Power
```

```
Record
```

```
Red
```

```
Rew
```

6.6 Retrieving firmware information

```
hardwarequery="firmware version" mac="00-20-4a-d0-16-d3"
```

```
MKIII 1.4.3 'Eiger', USB-IF 1.01 - (c) RedRat Ltd 2011
```

6.7 Signal transmission

```
ip="192.168.1.49" dataset="sky+" signal="9" output="1"
```

Or if sending via a USB RedRat3 device:

```
name="RR3-II PT2" dataset="sky+" signal="2"
```