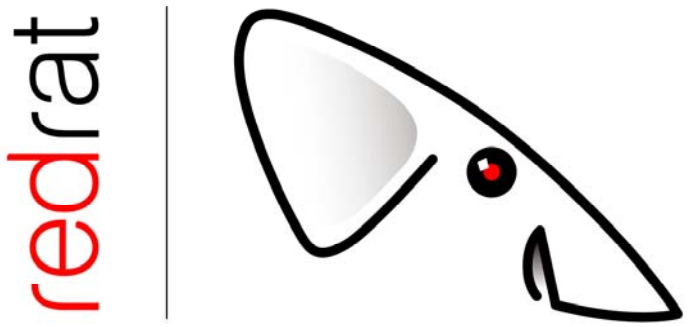
Using the RedRat3 with ShowShifter

Chris Dodge – RedRat Ltd

28 May 2004



Introduction

The *RedRatControl* application allows you to use an infrared remote control with PC applications and additionally output IR signals from third party applications for operations such as changing channels on a set-top box. This guide explains how to setup *RedRatControl* for use with ShowShifter, but for a more general introduction to using *RedRatControl*, please see the *RedRatControl Getting Started* guide.

You may be asking *"why doesn't the RedRat3 just work with ShowShifer, why do I need a separate application?"* RedRat's software is based on Microsoft's .NET whereas ShowShifter isn't, making integration a little more complex than otherwise and the use of an external program preferable. However, once *RedRatControl* has been correctly setup, you shouldn't actually notice it at all.

There are two ways in which the RedRat3 can be used with ShowShifter:

1. Outputting the IR signals to change channels on an external set-top box.
2. Controlling ShowShifter with almost any remote control that you already own.

Prerequisites and Installation

Before installing RedRat Control, please ensure that you have the following installed:

- The Microsoft .NET Framework 1.1. This is delivered on your RedRat CD, but can also be downloaded from the Microsoft website: http://www.microsoft.com/net.
- The RedRat Signal Database Utility – version 1.17 or later. When setting up control operations, remote control signals are dragged from the signal DB utility and dropped in *RedRatControl*. The signal database utility can be downloaded from the RedRat website at: http://www.redrat.co.uk/RedRat3/Software/SignalDBUtil
- The *RedRatControl* application – version 1.7 or later. This can be downloaded from the RedRat website at http://www.redrat.co.uk/RedRat3/Software/RedRatControl/.

Once *RedRatControl* has been installed, an example file can be loaded to view a typical setup for use with ShowShifter.

A Quick Start to Using RedRatControl with ShowShifter

"Ahh good" you're thinking, I don't want to read this whole document before I start experimenting with RedRatControl. A very brief introduction follows below, but if this does not make sense then jump this section and read the rest of the information in a little more detail.

Loading the Example File

Load the *ShowShifter_Demo.xml* file into *RedRatControl*. This can be found in the directory in which RedRatControl has been installed, which on a normal XP system is in:

```
C:\Program Files\RedRat\RedRat Control
```

This setup file uses a Sky Digibox remote for ShowShifter control and outputs standard NTL IR signals for changing channels on an NTL set-top box (STB).

Changing Channels on Your Set-Top Box

To create your own macros for channel changing, take the following steps:

- Start the Signal DB Utility and load signal data or capture the signals from your remote.
- Open the macro editor in RedRatControl using the *Edit → Macro* Editor menu item.
- Add a macro and rename it to something sensible e.g. "BBC1".
- Drag the required signals from the Signal DB Utility to the right-hand field in the macro editor, which for BBC1 on NTL and Telewest STBs is "1", "0", "1".
- Ensure that the *RedRat* field for each signal in the macro is the one you want to use. If not, then double click on the signal to bring up the "Macro Action Editor" and set the *RedRatLocation* to the RedRat you want to use.
- You can now test the macro from the editor using the aptly named “Test” button.
- Save the macro data from the *File → Save* or *File → Save As…* menu item.

Once you have a few macros setup, you can then try using the *SendMacroW* program to output them. This is the application that will be used in ShowShifter to change channels. It can be found in the *RedRatControl* installation directory, typically `C:\Program Files\RedRat\RedRat Control` and can be called from there or copied elsewhere. To test it, ensure that *RedRatControl* is running, open a command/DOS prompt and type:

```
SendMacroW BBC1
```

which will cause *RedRatControl* to output this macro.

ShowShifter now needs to be setup to use *SendMacroW* when changing the STB channel, so in the Showshifter's channel command line, type in the full path to *SendMacroW* with the correct macro name as parameter, e.g.:

```
C:\Program Files\RedRat\RedRat Control\SendMacroW BBC1
```

Controlling ShowShifter with a Remote

The RedRatControl main window shows the mappings of input IR signals to actions that take place on the computer. If you load the example *ShowShifter_Demo.xml* file then you can see an example setup using a Sky Digibox remote to control ShowShifter. Instructions are sent to ShowShifter using ShowShifter messages, so for example the Sky Digibox remote control Vol+ button has been mapped to the VOLUME_UP ShowShifter message.

Note: The *Enable* button must be pressed in the *RedRatControl* main window before it will start to respond to IR input from the remote.

To create your own set of mappings using a different remote, take the following steps:

- Start the Signal DB Utility and load signal data or capture the signals from the remote you want to use to control ShowShifter.
- Drag either the whole remote or individual signals from the Signal DB Utility into the main pane of *RedRatControl*.
- Double click on the new row in the mapping list to bring up the mapping editor box.

- Select the ShowShifter *Mapping Type* and then the required ShowShifter message.
- Once this has been done for a few remote control signals, save the mapping set, click the *Enable* button in *RedRatControl*, start ShowShifter and test remote control input.

By experimenting with the *CommandTargetWindow* and *PostActionDelay* options for each mapping, smooth and intuitive control of ShowShifter can be obtained.

The following sections of this document will take you through this process in more detail.

Information about RedRatControl Files

A *RedRatControl* file holds all the information needed for a single "setup" on a PC, for example a setup for *"using my laptop as a media centre"*. If you were then to use your laptop in a different context, such as for presentations, you would probably want a completely different setup file – *"laptop for presentations"* - as the programs to be controlled are different and you may be using a different remote.

A *RedRatControl* file contains the following data:

Macros (for IR Output)

This is a set of IR signals that are output to perform a certain operation on external equipment, for example on NTL set-top boxes (STBs), changing to BBC1 requires outputting three IR signals to change to channel 101 which is one single macro.

Mapping Set (for IR Input)

The set of mappings from IR signals on your remote control to actions on the PC for the current application you want to control. If the PC application is ShowShifter, then you would probably map *Volume Up* on your remote to the ShowShifter message VOLUME_UP and so on with the other buttons on your remote. This is one *Mapping Set*.

However, if you switch to using Windows Media Player, you would then probably want to change the IR signal mappings so that *Volume Up* on your remote adjusted the PCs volume. This would then be a second *Mapping Set*.

RedRatControl can be setup to automatically monitor which application you have running in the foreground and automatically switch mapping sets accordingly.

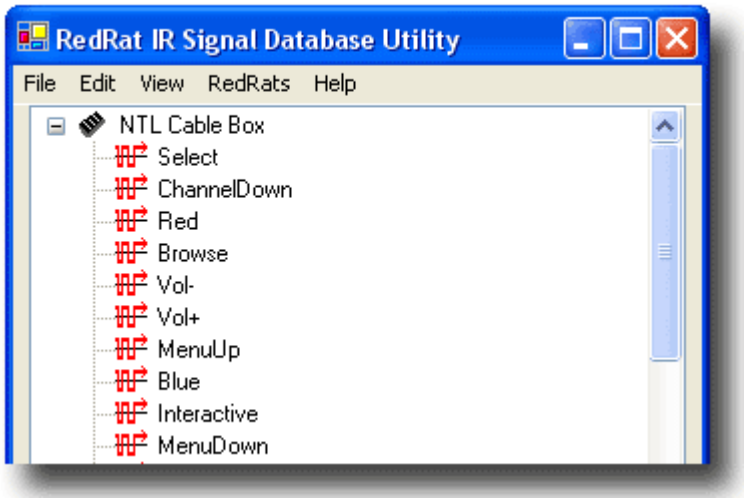
A Mapping

As is probably understood from the previous paragraph, a single mapping is the correspondence between one IR signal and an action on the PC, which could be a key press, mouse movement, volume control, or a ShowShifter message.

In Detail - Setting Up RedRatControl to Change Set-Top Box Channels

Step 1 – Obtaining the IR Signal Dataset for Your STB

The Signal DB Utility is used to collect the signals for your STB's remote, giving a list as shown below:



Either… Download IR Signal Datasets from the RedRat Website

The signal datasets for some STBs are available on the RedRat website at http://www.redrat.co.uk/IRData. These can be dragged from the webpage and dropped directly in the Signal DB Utility main window.

Or… Manually Capture IR Signals

If the dataset for your STB is not available on the RedRat website, then you will need to capture the signals from your remote:

1. Add a new device/remote giving it an appropriate name (*Edit → Add Device/Remote* menu item).
2. Select the new device/remote in the main pane and add a signal (*Edit → Add Signal*).
3. Give the signal the correct name for a button on your remote, then press *Learn IR*.
4. Holding the remote about 10cm from the front of the RedRat3, press this remote button twice with a firm press but not holding the button down too long.
5. Repeat the process for all buttons you intend to use.

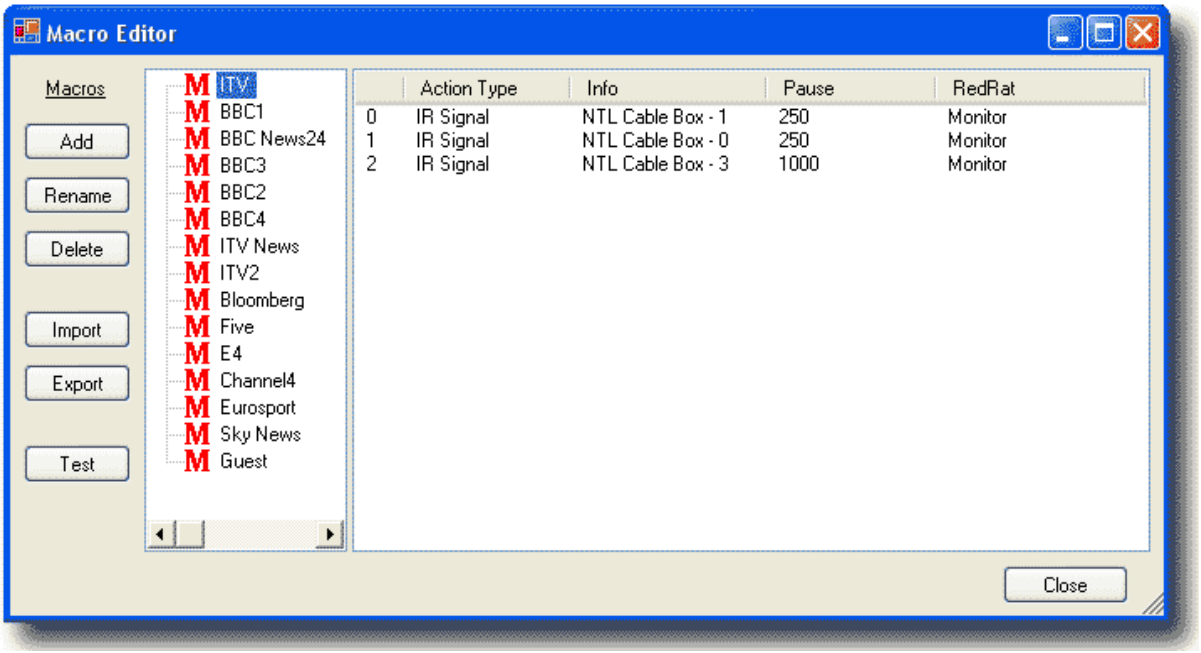
Step 2 – Creating Macros to Change Channels on Your STB

Start *RedRatControl* and open the macro editor from the *Edit → Macro Editor* menu item. It has two main panes, the left-hand one showing the list of *macros* and the right-hand pane showing the *actions* that have been setup as part of the macro. When a macro is executed, it steps through the actions in the order given in the macro pane.

Take the following steps to create a macro:

1. Click *Add* to insert an empty macro.
2. Click *Rename* to give it a name descriptive of its intended operation.
3. Ensure that the macro is selected (blue or grey background) and then drag an IR signal from the signal database utility to the right hand pane of the macro editor. This will create a new row in the action list.

4. Repeat drag and drop operation for all IR signals that are going to form part of the macro.



Each action in the action list has the following properties:

Action Type: Currently only IR signal actions are supported.

Info: This shows the remote and IR signal to be output.

Pause (units of mS): Following the output of each IR signal a default pause is given. This is so that audio/visual equipment is able to recognize the discrete IR signals rather than seeing one long stream of concatenated IR. The pause length value can be adjusted, for example when turning on a TV or set-top box, it can sometimes take a couple of seconds before it responds to further IR commands.

RedRat: The displays which RedRat is to be used for output of the signals.

Re-ordering Actions

Actions can be re-ordered by dragging them to the required position.

Editing Actions

Double clicking on an action will bring up an action editor dialog.

Testing Actions and Macros

Both single actions and complete macros can be tested by right-clicking on the action or macro respectively, which will bring up a small menu.

Step 3 – Using SendMacro (or SendMacroW)

SendMacro is a lightweight application that simply sends a message to *RedRatControl* instructing it to output the macro, and is intended to be used by any application that needs to change channel on the STB (ShowShifter in this case). *RedRatControl* must be running with the necessary macros loaded.

SendMacro is a console application, so useful for testing operation from a DOS or Command window. It can give feedback on progress and reports any errors it encounters. When used within some applications, including ShowShifter, a DOS window pops up every time it is called, so *SendMacroW* should be used instead.

SendMacroW is an invisible windows application so it will not cause a DOS window to pop up, but as a result it does not give any feedback or error messages. Therefore it is recommended that the setup be developed and tested with *SendMacro* and once working replaced with *SendMacroW*.

These two executables can be found in the directory in which the *RedRatControl* has been installed, which on most systems is `C:\Program Files\RedRat\RedRat Control`

As an example, once *RedRatControl* is running, type in the following at a command prompt:

```
SendMacro BBC1
```

This will cause *RedRatControl* to output the macro called *BBC1* to the RedRat setup for that macro (or the first found RedRat if none set).

SendMacro options are:

-output <output name>

Sends the macro out via the given RedRat (or *output group* if using an irNetBox), overriding the IR output set in the macro.

-verbose

Prints out messages to the console window tracing the program's operation. This can be useful when testing.

-help

Prints out a simple help message.

In Detail – Controlling ShowShifter with a Remote

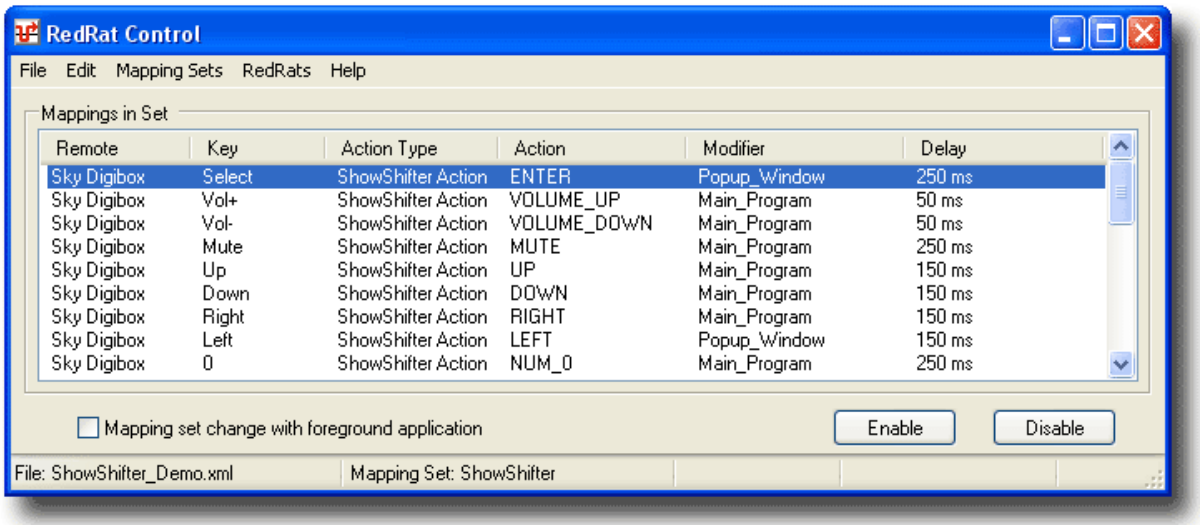
Step 1 – Capture of Remote Control Signals

The IR signals for the remote you want to use to control ShowShifter have to be captured in this step using the Signal DB Utility as described previously.

Step 2 – Creating a Mapping Set

1. Start RedRatControl and drag either single signals or the whole device/remote from the Signal DB Utility into the main pane.
2. Each remote button that you want to cause an action in ShowShifter has to be configured which is done by double clicking on it to bring up the mapping editor dialog.
3. Set the Mapping Type to *ShowShifter* and then the select the *ShowShifterCode* that is to be sent to ShowShifter whenever this remote control button is pressed.

4. Once a basic mapping set has been created, it can then be saved (*File* → *Save As…*) and tested by pressing the *Enable* button (to enable the RedRat3's IR detector), starting up ShowShifter and using your remote.



Mapping Options

To get smooth and intuitive remote control behaviour, there are a couple of fine-tuning options that can be set for each mapping.

Command Target Window

If the remote control does not seem to have any effect when ShowShifter pops up a dialog box, then set the *CommandTargetWindow* value to *Popup_Window*. This will cause messages to be sent to the popup window if one is being displayed.

Auto Repeat Action

There are two main ways in which remotes communicate long button presses:

1) A – A – A – A – A – A – A – A …
2) A – B – B – B – B – B – B – B …

In 1) above, where every signal repeat is identical, *RedRatControl* will produce an appropriate stream of actions. If they come too rapidly, the *PostActionDelay* value can be adjusted to slow them down.

In 2), *RedRatControl* may not decode the 'B' sections but it will notice incoming signal repeats. Setting *AutoRepeatAction* to true will cause it repeat the last action for every unrecognized input signal that rapidly follows a known 'A' signal section.