

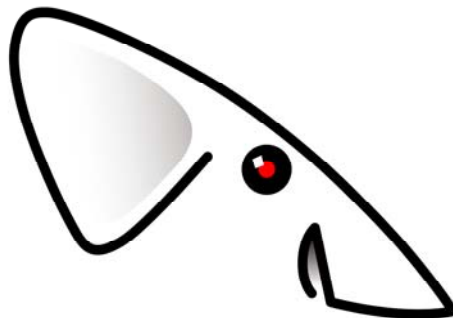
# Getting Started with the RedRat Control Application

Chris Dodge – RedRat Ltd

Version 2.04

15 February 2007

redrat



<b>INTRODUCTION</b>	<b>3</b>
<b>PREREQUISITES AND INSTALLATION</b>	<b>3</b>
<b>STEP1 – CAPTURE OF REMOTE CONTROL SIGNALS</b>	<b>3</b>
<b>STEP 2 – CREATING A MAPPING SET</b>	<b>4</b>
<b>TYPES OF COMPUTER CONTROL</b>	<b>5</b>
<b>Key Press</b>	<b>5</b>
<b>Mouse Action</b>	<b>5</b>
<b>Audio Control</b>	<b>5</b>
<b>ShowShifter Control</b>	<b>6</b>
<b>Macro Output</b>	<b>6</b>
<b>FURTHER OPTIONS AND SETTINGS</b>	<b>6</b>
Mapping set changes with foreground application	6
RedRats Menu Item	6
Options Dialog	6
General Options	6
Mouse Control	7
Logging On/Off	7
<b>OUTPUTTING IR SIGNALS</b>	<b>7</b>
<b>Creating Macros</b>	<b>7</b>
Re-ordering Actions	8
Editing Actions	8
Testing Actions and Macros	8
<b>Macro Output Using SendMacro</b>	<b>8</b>
-output <output name>	8
-verbose	8
-block	9
-help	9
<b>Using SendMacroW</b>	<b>9</b>

## Introduction

RedRat Control allows you to use an infrared remote to control PC applications and provides a simple method for PC applications to output IR signals. The control of PC applications is done by initiating various actions when it recognizes IR signal input, such as creating key strokes, mouse movement or direct PC audio control. Output of IR signals is based around *Macros* (sets of IR signals) that can then be output using a small command line utility called *SendMacro* (or *SendMacroW*).

This *getting started* guide will take you through the steps necessary to start using RedRat Control, using some basic operations of Windows Media Player as an example.

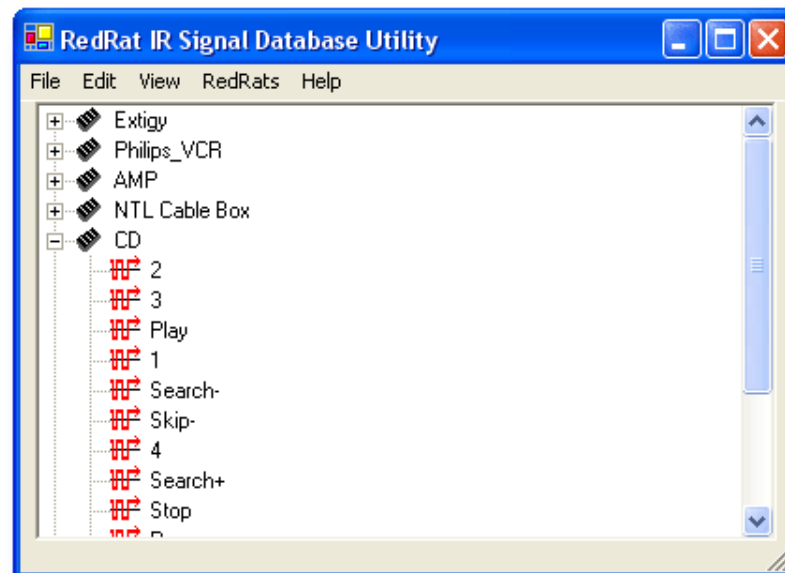
## Prerequisites and Installation

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 to *RedRatControl*. The signal database utility can be downloaded from the RedRat website at: <http://www.redrat.co.uk/RedRat3/Software/SignalDBUtil>

## Step1 – Capture of Remote Control Signals

For the remote that you want to use to control an application, capture the IR signals using the signal database utility.



The steps to create the database are as follows:

1. Add a device/remote (*Edit* → *Add Device/Remote*)
2. Select the new device and add an IR signal to it (*Edit* → *Add Signal*).

3. In the “Add New Signal” dialog, give the signal an appropriate name, then press the *Learn IR* button.
4. Point the remote at the RedRat and press the remote control button twice, separating the presses by a one or two seconds. When pressing the button, make sure it is a firm press, but not too long.
5. Add as many signals from the remote as you intend use with *RedRatControl*.
6. Save the signal DB (*File* → *Save As...*).

The captured signals can be tested to validate that they have been recorded correctly and so recognized when input by using the decode window (*View* → *Decode Window*). Once started, point the remote at the RedRat3 and press some of the buttons you aim to use.

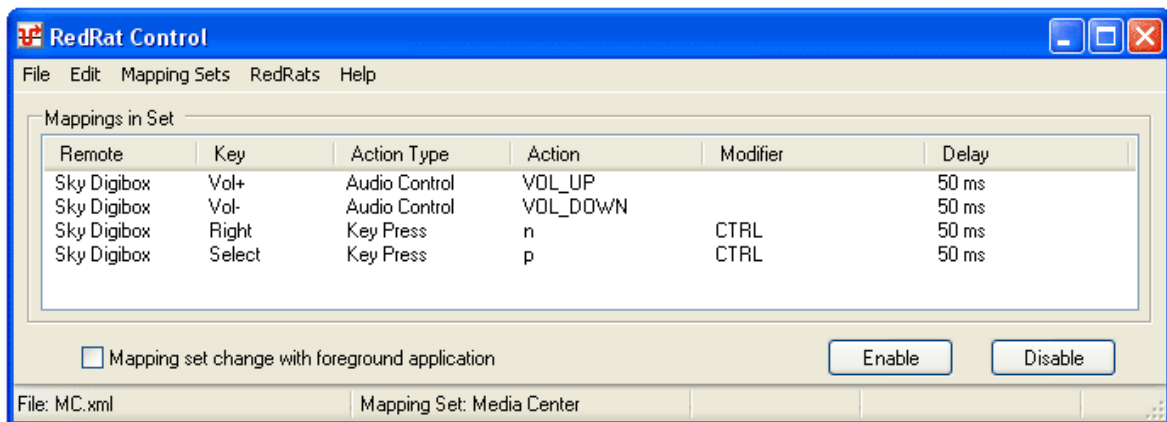
## Step 2 – Creating a Mapping Set

To briefly explain some of the terminology used here:

A **Mapping** is how a single remote control signal maps to an action.

A **Mapping Set** is a collection of such mappings, usually for a particular application. For example you may have one mapping set for Windows Media Player and another for Winamp.

Given that there are different mapping sets for different applications, one can tell *RedRatControl* to monitor which application is currently in the foreground and automatically switch mapping set to the correct one.

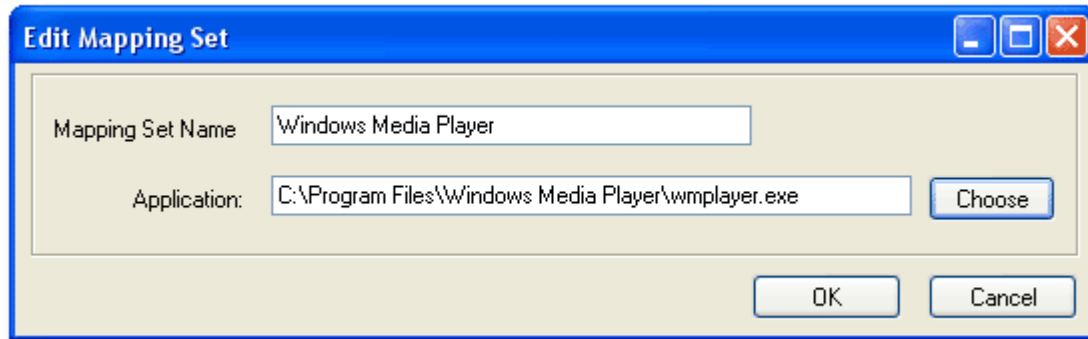


The picture above shows the *RedRatControl* interface once a simple mapping set has been created for basic control of Windows Media Player. The configured actions are:

- The *Select* button on a Sky Digibox remote will cause the key combination CTRL-P to be sent, which in Windows Media Player issues the Play/Pause command.
- Vol+ on the remote sends the F10 key press (Vol Up in Windows Media Player).
- Vol- on the remote sends the F9 key press (Vol Down in Windows Media Player).

The following steps will take you through the process of creating a mapping set as given above.

1. Start Windows Media Player, or if you do not want to control a different application, then start it now.
2. Start *RedRatControl* and from the Signal Database Utility drag either single signals or the whole remote into the main *RedRatControl* window. As we do not yet have a name for the mapping set, the dialog box shown below will prompt you for one.



- Give the mapping set a name (e.g. Windows Media Player) and then either enter the full name of the application executable to be controlled in the *Application* box, or press *Choose*. The *Choose* button brings up a list of running applications from which the appropriate one can be selected.
3. In the main *RedRatControl* window, double-click in turn on each of the remote control signals that you want to use. This brings up a box in which you can set the action type and values specific to that type, for example *KeyPress* with key value of 'p'.
  4. Now that a simple mapping set has been setup, save it (*File* → *Save As*) and test it using the *Enable* button. This enables IR input from the RedRat, so bring Windows Media Player to the foreground and validate that pressing remote control signals has the desired effect.

## Types of Computer Control

Each input IR signal can initiate an action on the computer, which can be selected from the options given below.

### **Key Press**

As described in the previous section, key presses can be simulated, including modifier keys SHIFT (to give capitals), CTRL, ALT and ALT GR.

### **Mouse Action**

Mouse movement and clicks can be simulated. If simulating mouse movement, the default post action delay is 0 so that smooth mouse movement is obtained by using as many input IR signals as possible.

### **Audio Control**

This directly controls the PC's audio, i.e. volume up, down and mute. The default post action delay following the mute action is 500ms (half a second) so that a slightly prolonged IR signal button press won't cause the mute state to be toggled many times.

If multiple soundcards are available on the system, then the default soundcard at the time of program start will be controlled.

### **ShowShifter Control**

ShowShifter has a set of defined Windows messages for IR remote control commands which can be send directly from *RedRatControl*.

### **Macro Output**

A macro (sequence of IR signals) can be output as the response to an incoming IR signal. The macro has to have already been setup using the *Macro Editor* – see the section on creating macros.

## **Further Options and Settings**

### **Mapping set changes with foreground application**

As mentioned previously, *RedRatControl* can automatically detect which foreground application is running and set the mapping set accordingly. If this is not required, then remove the check from the check box.

Mapping sets can be manually selected using the *Mapping Sets* menu item.

### **RedRats Menu Item**

If you have more than one RedRat attached to the computer, then use the *RedRats* menu to find then and select which one is to take remote control input.

### **Options Dialog**

Found under the *File* → *Options* menu item, this dialog box allows you to set the following options:

#### **General Options**

**Default delay following input signal:** Inserts dead time following the execution of an action before the next remote control signal is acted on. Why is this needed? Some remotes repeat signal output quite rapidly, so anything more than the lightest button press will cause multiple signals to be sent, causing the action to be initiated multiple times. Putting in a pause (dead-time) following each signal input will ensure that what is intended to be a single button press will actually be translated as a single action. Units are in milliseconds.

**Enable IR input on startup:** If the program is required to accept IR input as soon as it is started, then check this option.

**Hide window when minimized:** Removes the window from the taskbar when its minimized. A small icon remains in the notification area (right) on the task bar.

**Default mapping set:** If *RedRatControl* is configured so that the mapping set changes with the foreground application, then one mapping set can be chosen as the default mapping set, so the application will use this if the foreground application is not recognized or no mapping set is given for that application.

## Mouse Control

**Mouse acceleration factor:** If a remote control button is held down to move the mouse a long distance across the screen, then the mouse speed will gradually increase. The rate of increase is controlled with this option.

**Maximum mouse speed:** If the mouse acceleration continues, then it soon reaches a speed that is difficult to manage. This option sets the maximum speed the mouse can reach, in units of screen pixels per input signal.

## Logging On/Off

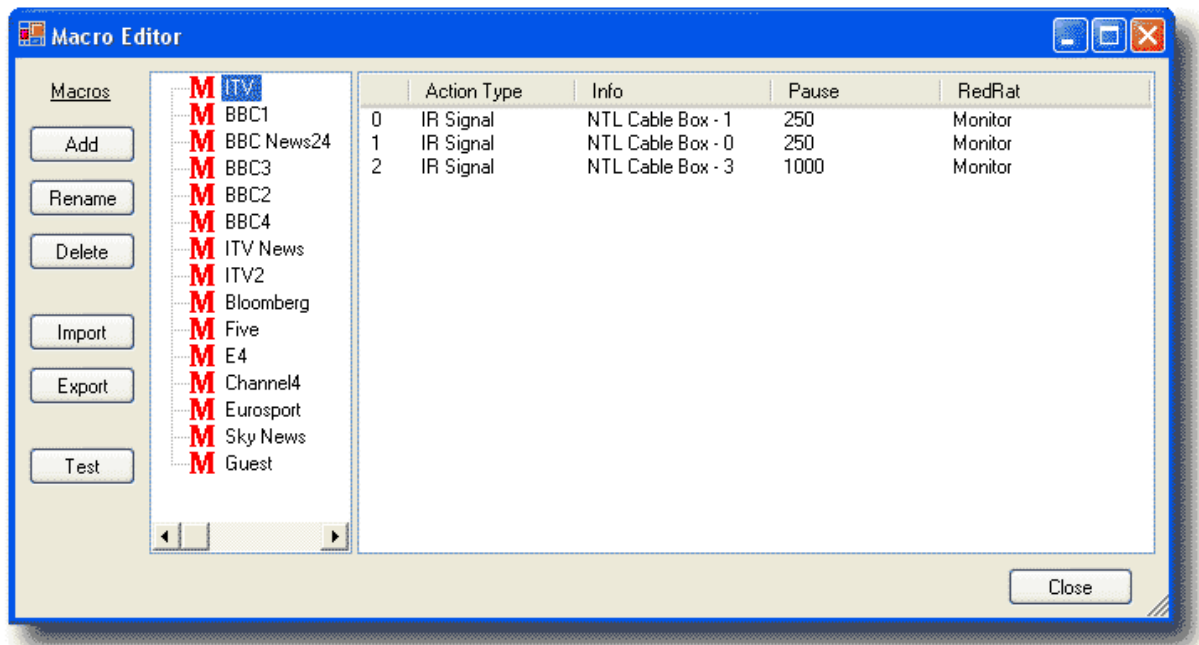
This enables/disables logging of IR signal input, which can be useful if the program does not seem to produce the desired effects, or you would like to track IR signal input for some other reason. Log output is sent to the selected file.

## Outputting IR Signals

RedRatControl has also been designed to support the output of IR signals from third party programs, such as webservers or PVR (computer based video) applications. The basic unit of output IR operation is the *Macro*, i.e. one or more IR signals, which are created with the macro editor

## Creating Macros

Start the macro editor from *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 important so that audio/visual equipment is able to recognize the discrete IR signals rather than seeing one long stream of concatenated IR. This 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.

## Macro Output Using SendMacro

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`

If you want to send a macro from another application, such as a web server in response to a button press, then the *SendMacro* executable can be used in conjunction with *RedRatControl*. 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 the 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.

## **-block**

In some situations, you may not want *SendMacro* to return until it has completed the macro output, especially if used within systems that send long list of macros for output. By using this option, *SendMacro* will block while the macro is being output, and will only complete once the macro output is finished.

## **-help**

Prints out a simple help message.

## ***Using SendMacroW***

As *SendMacro* is a console application, in many situations it will cause a DOS or command window to be shown briefly while it is run. To overcome this program, use the *SendMacroW* version of the program instead, which is a **Windows** application rather than a console application. It is identical in operation to *SendMacro*, but does not have the **-verbose** or **-help** options, however all other command line information is passed in the same way.

Please note that when using the **-block** option from a command line, it appears as if *SendMacroW* returns immediately. This is in fact not the case, but as it is a Windows application, it starts in its own execution environment and so is not run in the DOS execution environment.