

---

# Rivendell RDCatch Monitor Protocol

Fred Gleason

## Table of Contents

Overview .....	1
Unprivileged Commands .....	1
Drop Connection .....	1
Send Password .....	2
Privileged Commands .....	2
Deck Event Processed .....	2
Reload Deck List .....	2
Reload Event List .....	2
Add Event .....	2
Remove Event .....	3
Update Event .....	3
Request Deck Status .....	3
Enable Metering .....	4
Stop Deck .....	4
Purge Event .....	4
Reload Time Offset .....	4
Reload Heartbeat Configuration .....	4
Heartbeat Pulse .....	4
Restart Dropbox Instances .....	5
Input Monitor State .....	5
Set Exit Code .....	5

## Overview

This defines the IP protocol used for communication between different modules of Rivendell and the **rdcatchd(8)** daemon.

Connection to ripcd is by means of a TCP SOCK\_STREAM connection to TCP port **6006**. The format of a message is as follows:

*cmd-code* [*arg*] [ . . . ]!

*cmd-code*                   A two letter command code, describing the generic action to be performed

*arg*                         Zero or more arguments, delimited by spaces or, if the last argument, by ! (see below)

!                            The ASCII character 33, indicating the end of the command sequence.

## Unprivileged Commands

No authentication is required to execute these.

### Drop Connection

End the session and drop the TCP connection.

DC!

## Send Password

Send a password to the server for authentication.

**PW *passwd*!**

*passwd* A password to be supplied before granting the client access.

**rdcatchd(8)** will respond with **PW +!** or **PW -!**, indicating the success or failure of the authentication.

## Privileged Commands

A connection must be authenticated before these can be executed.

## Deck Event Processed

Sent by RDCatchd whenever a deck event is processed.

**DE *deck-num event-num*!**

*deck-num* The number of the deck originating the event. Record decks have numbers in the range 1 through 127, while Play decks have numbers in the range 128 through 254.

*event-num* The new event state of the specified deck.

See the section called “Request Deck Status” for the list of possible deck event states.

## Reload Deck List

Reload the record/play deck configuration.

**RD!**

**rdcatchd(8)** will respond with **RS +!**.

## Reload Event List

Reload the list of scheduled events.

**RS!**

**rdcatchd(8)** will respond with **RS +!**.

## Add Event

Add event to the event list.

**RA *event-num*!**

*event-num* The number of the event to add.

**rdcatchd**(8) will respond with RA *event-num* !.

## Remove Event

Remove event from the event list.

**RR *event-num*!**

*event-num* The number of the event to remove.

**rdcatchd**(8) will respond with RR *event-num* !.

## Update Event

Update event from the event list.

**RU *event-num*!**

*event-num* The number of the event to update.

**rdcatchd**(8) will respond with RU *event-num* !.

## Request Deck Status

Request the current deck status.

**RE *deck-num*!**

*deck-num* The number of the deck for which to return status.

If *deck-num* is greater than zero, **rdcatchd**(8) will respond with RE *deck-num status id cutname* !

*deck-num* Channel number

*status* The current status of the deck. Possible values are:

**Table 1. Deck Status Codes**

Code	Meaning
0	Offline
1	Idle
2	Ready
3	Active (playing or recording)
4	Waiting (for a GPI)

*event-id* ID number of current event (from RECORDINGS table)

*cutname* The cutname of the event (present only for *status*==3).

If *channel* is zero, **rdcatchd**(8) will respond for every event whose status is non-idle.

An RE packet will also be sent automatically to all active connections upon any change of deck status.

## Enable Metering

Enable/Disable Audio Metering

**RM *state*!**

*state* 1 = Enabled, 0 = Disabled.

When metering is active, meter data packets will be periodically sent, as follows:

**RM *deck chan level*!**

*deck* Deck number

*chan* Channel, 0 = Left, 1 = Right

*level* Left audio level, in 1/100 of dbFS

## Stop Deck

Stop active event.

**SR *deck-num*!**

*deck-num* The number of the deck to stop.

## Purge Event

Report purging of one-shot event.

**PE *event-num*!**

*event-num* The number of the purged event.

Received upon completion of a one-shot event by **rdcatchd(8)** to indicate that the event has been purged.

## Reload Time Offset

Reload the time offset value from the database.

**RO!**

## Reload Heartbeat Configuration

Reload the heartbeat configuration from the database.

**RH!**

## Heartbeat Pulse

Heartbeat Pulse, for connection keep-alive.

**HB!**

## Restart Dropbox Instances

Restart all **rdimport**(1) instances for running dropboxes.

**RX!**

## Input Monitor State

Turn the input monitor on or off.

**MN *deck-num state!***

*deck-num* The number of the deck.

*state* 1 = Enabled, 0 = Disabled.

## Set Exit Code

Set the exit code of an event.

**SC *event-num code msg!***

*event-num* The number of the event.

*code* The numeric code to set.

*msg* The text message to set.