

---

## Name

rd\_listcartscuts — Rivendell List Cart/Cuts C Library Function

## Synopsis

```
#include <rivwebcapi/rd_listcartscuts.h>

int   RD_ListCartsCuts(cart[], hostname[], username[], passwd[],
ticket[], group_name[], filter[], type[], user_agent[], numrecs);

struct rd_cart * cart[];
const char hostname[];
const char username[];
const char passwd[];
const char ticket[];
const char group_name[];
const char filter[];
const char type[];
const char user_agent[];
unsigned * numrecs;

struct rd_cut *RD_ListCartsCuts_GetCut(carts[], cart_rec, cut_rec);

struct rd_cart carts[];
int cart_rec;
int cut_rec;

void RD_ListCartsCuts_Free(carts[], numrecs);

struct rd_cart carts[];
unsigned numrecs;
```

## Description

This function lists the fields in one or more pre-existing cart(s). The user can also filter the records by group name, cart type, and/or a user provided search string. Unlike **RD\_ListCarts(7)**, **RD\_ListCartsCuts(7)** will also fetch the full list of cuts associated with each cart.

**Table 1. RD\_ListCartsCuts function call fields**

| FIELD NAME | FIELD TYPE                   | MEANING                                   | REMARKS  |
|------------|------------------------------|---|--|
| *cart      | Pointer to rd_cart structure | Memory location to store cart information | Mandatory                                      |
| hostname   | Character Array              | Name Of Rivendell DB Host                 | Mandatory                                      |
| username   | Character Array              | Rivendell User Name                       | Mandatory When NO Ticket Provided              |
| passwd     | Character Array              | Rivendell User Password                   | Mandatory When NO Ticket Provided              |
| ticket     | Character Array              | Rivendell Authentication Ticket           | Mandatory When NO User/Password Pair Provided. |

| FIELD NAME | FIELD TYPE         | MEANING  | REMARKS                                     |
|------------|--------------------|--|---|
| group_name | character array    | Group Name   | Optional                                    |
| filter     | character array    | Filter - searches for specific strings within the following fields: Title, Artist, Client, Agency, Album, Label, Publisher, Composer, Conductor, Song_Id, User_Defined, and Cart Number. | Optional                                    |
| type       | character array    | Type of Cart records. If set can be Audio, or Macro.   | Optional                                    |
| user_agent | Character Array    | User Agent Value put into HTTP request   | Optional (default is Rivendell-C-API/x.x.x) |
| *numrecs   | pointer to integer | memory location for number of records returned   | Mandatory                                   |

When successful the function will return the number of records sent (numrecs) and a `rd_cart` structure which is stored in the provided memory locations. (See the **rd\_listcartcuts(7)** man page for a listing of the `rd_cart` structure).

## Accessing Cut Information

Information about the cuts associated with each cart can be accessed from the returned `rd_cart` structure through use of the **RD\_ListCartsCuts\_GetCut()** function. The `cart_rec` parameter should range between 0 and one less than the value returned in the `numrecs` argument to **RD\_ListCartsCuts**, while the `cut_rec` parameter should range between 0 and one less than the value of the `cart_cut_quantity` member of the cart's `rd_cart` structure.

Cut information is returned in the form of a `rd_cut` structure. (See the **rd\_listcartcuts(7)** man page for a listing of the `rd_cut` structure).

## Freeing Memory

When the returned `rd_cart` structure is no longer required, it should be freed by passing it to the **RD\_ListCartsCuts\_Free()** function.

## RETURN VALUE

On success, zero is returned. Using the provided parameters an `rd_cart` structure is returned and the number of records is returned.

If a server error occurs a -1 is returned. If a client error occurs a specific error number is returned.

## Errors

403 User Authentication Error.

404 No Such Group Exists.

nnn Unknown Error Occurred.