Name

rd_editcart — Rivendell Edit Cart C Library Function

Synopsis

#include <rivwebcapi/rd_editcart.h>

```
int  RD_EditCart(cart[],  edit_cart_values,  hostname[],  username[],
passwd[], ticket[], cartnumber, user_agent[], numrecs);

struct rd_cart * cart[];
struct edit_cart_values edit_cart_values;
const char hostname[];
const char username[];
const char passwd[];
const char ticket[];
const unsigned cartnumber;
const char user_agent[];
unsigned * numrecs;
```

Description

RD_EditCart is the function to use to edit the fields within a cart that already exists in the Rivendell Database.

This function edits a pre-existing cart. User must provide the cart number and any fields which they would like to change. The structure edit_cart_values must be used to tell the API which fields will be changed.

Table 1. RD_EditCart function call fields

FIELD NAME	FIELD TYPE	MEANING	REMARKS
*rd_cart	Pointer to rd_cart structure	Memory location to store cart information	Mandatory
edit_cart_values	edit_cart_values structure	This structure contains the new cart information to update	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 Authentification Ticket	Mandatory When NO User/Password Pair Provided.
cartnumber	unsigned integer	Cart Number	Mandatory
user_agent	Character Array	User Agent Value put into HTTP request	Optional (default is Rivendell-C-API/x.x.x)

FIELD NAME	FIELD TYPE	MEANING	REMARKS
*numrecs	pointer to integer	memory location for number of records returned	Mandatory

This routine expects an input structure of type edit_cart_values - listed below. Each field has a boolean flag field (starting with use_) which designates whether to update the field's value or not. One (1) = true - Use the value in the field or Zero (0) - Ignore the field.

The edit_cart_values structure must be pre-filled with zeroes and has the following format:

```
struct edit_cart_values {
  char cart_grp_name[41];
  int use_cart_grp_name;
  char cart_title[1021];
  int use_cart_title;
  char cart_artist[1021];
  int use_cart_artist;
  char cart_album[1021];
  int use_cart_album;
  int cart_year;
  int use_cart_year;
  char cart_label[257];
  int use_cart_label;
  char cart_client[257];
  int use_cart_client;
  char cart_agency[257];
  int use_cart_agency;
  char cart_publisher[257];
  int use_cart_publisher;
  char cart_composer[257];
  int use_cart_composer;
  char cart_conductor[257];
  int use_cart_conductor;
  char cart_user_defined[1021];
  int use_cart_user_defined;
  int  cart_usage_code;
  int use_cart_usage_code;
  int  cart_forced_length;
  int use_cart_forced_length;
  int cart_enforce_length;
  int use_cart_enforce_length;
  int cart_asyncronous;
  int use_cart_asyncronous;
  char cart_owner[257];
  int use_cart_owner;
  char cart_notes[4096];
  int use_cart_notes;
};
```

When successful function will return the number of records sent (numrecs) and a rd_cart structure which is stored in the provided memory locations. The rd_cart structure has the following fields:

```
struct rd_cart {
 unsigned cart_number;                /* Cart Number */
 unsigned cart_type;                  /* Cart Type */
 char cart_grp_name[41];              /* Group Name */
 char cart_title[1021];               /* Cart Title */
 char cart_artist[1021];              /* Artist */
 char cart_album[1021];               /* Album */
 int cart_year;                       /* Year */
 char cart_label[257];                /* Label */
 char cart_client[257];               /* Client */
 char cart_agency[257];               /* Agency */
 char cart_publisher[257];            /* Publisher */
 char cart_composer[257];             /* Composer */
 char cart_conductor[257];            /* Conductor */
 char cart_user_defined[1021];        /* User Defined */
 int  cart_usage_code;                /* Usage Code */
 int cart_forced_length;              /* Forced Length */
 int cart_average_length;             /* AverageLength */
 int cart_length_deviation;           /* Length Deviation */
 int cart_average_segue_length;       /* Average Segue Length */
 int cart_average_hook_length;        /* Average Hook Length */
 unsigned  cart_cut_quantity;         /* Cut Quantity */
 unsigned  cart_last_cut_played;      /* Last Cut Played */
 unsigned  cart_validity;             /* Validity */
 int cart_enforce_length;             /* Enforce Length Flag */
 int cart_asyncronous;                /* Asyncronous Flag */
 char cart_owner[257];                 /* Owner */
 char cart_notes[4096];               /* Notes */
};
```

All character arrays above are the sizes listed and are null-terminated. Character encoding is UTF-8.

The cart number is a unsigned integer.

The cart_type is 1=Audio,2=Macro.

The cart_grp_name must be a valid Rivendell DB Group.

The cart_title is 255 characters.

The cart_artist is 255 characters.

The cart_album is 255 characters.

The cart_year is a date (YYYY).

The cart_label is 64 characters.

The cart_client is 64 characters.

The cart_agency is 64 characters.

```
The cart_publisher is 64 characters.

The cart_composer is 64 characters.

The cart_conductor is 64 characters.

The cart_user_defined is 255 characters.

The cart_usage_code is  0=Feature, 1=Theme Open,
                        2=Theme Close, 3=Theme Open/Close,
                        4=Background, 5=Comm/Promo

The cart_forced_length is in milliseconds.

The cart_average_length is in milliseconds.

The cart_length_deviation is in milliseconds.

The cart_average_segue_length is in milliseconds.

The cart_average_hook_length is in milliseconds.

The cart_cut_quantity is number of cuts.

The cart_last_cut_played is self explanatory.

The cart_validity is 0=NeverValid, 1=Conditional, 2=AlwaysValid
                     3=EvergreenValid, 4=FutureValid

The cart_enforce_length is True(1) or False(0).

The cart_asyncronous is True(1) or False(0).

The cart_owner is 64 characters.

The cart_user_notes is a character array up to 1024 characters.
```

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

400 Invalid Input Parameter.

403 User Authentification Error.

404 Unauthorized or No Such Group / Cart Exists.

409 Cart Out Of Range.

500 Unable to Create Cart.

nnn Unknown Error Occurred.