

AirMenu

API - [RIOCOM.DLL](#)

Version 1.0.0

(Draft)

12-04-2013

1 - CONFIG.INI

Description:

The config.ini is the configuration file of the RioCom.Dll. This file is created in the c:\X64 Folder. To activate AirMenu, insert the following lines inside the config.ini file.

ADD THIS LINES:

```
<AIRMENU>
  <USER>user@gmail.com</USER>
  <PASS>password</PASS>
  <REST_NAME>Ao Sol</REST_NAME>
  <RESTID></RESTID>
  <REST_TIMEZONE>Europe/Lisbon</REST_TIMEZONE>
  <REST_LOCALE>pt_PT</REST_LOCALE>

  <LOGIN_FUNCTION>P</LOGIN_FUNCTION>
  <LOGIN_ID>961</LOGIN_ID>
  <LOGIN_QT>1234</LOGIN_QT>
  <LOGIN_TABLE>66</LOGIN_TABLE>

  <EXPORT_MODE>automatic</EXPORT_MODE>
  <EMPLOYEE_EXPORT_MODE>automatic</EMPLOYEE_EXPORT_MODE>

  <BILL_FUNCTION>C</BILL_FUNCTION>

  <SEND_MESSAGES>NO</SEND_MESSAGES>
  <TERMINAL_2>1111</TERMINAL_2>
  <TERMINAL_3>2222</TERMINAL_3>
  <TERMINAL_10>public</TERMINAL_10>
  <API_KEY></API_KEY>
</AIRMENU>

<SEND_PLU_CODES>YES</SEND_PLU_CODES>
<EXITONCOMERROR>NO</EXITONCOMERROR>
```

Parameters description

<USER> The email of the restaurant owner
<PASS> The password
<REST_NAME> It will create a new Business inside AirMenu if it doesn't exist.

<REST_TIMEZONE> In Portugal should be:Europe/Lisbon
<REST_LOCALE> In Portugal should be:pt_PT

This parameters should be used if AirMenu needs to do Login to access the POS System.
During the radio_init, riocom.dll will use this parameters to try to open a valid session.

<LOGIN_FUNCTION>P</LOGIN_FUNCTION>
<LOGIN_ID>961</LOGIN_ID>
<LOGIN_QT>1234</LOGIN_QT>
<LOGIN_TABLE>66</LOGIN_TABLE>

<EXPORT_MODE> automatic , or manual. It will define if the order goes directly to the POS system or not.
<EMPLOYEE_EXPORT_MODE> automatic , or manual. It will define if the waiter order goes directly to the POS system or not.

<RESTID> Only to be used if the user has more than one restaurant. This parameter can be taken from the URL of the AirMenu front. www.airmenu.com/Front.html?r=xxxxxxxxx where xxxxxxxx is the parameter.

<SEND_MESSAGES>NO</SEND_MESSAGES> Waiter	Only put YES if the terminal will be used by the
<TERMINAL_2>998</TERMINAL_2>	All orders from Waiter number 998 -> terminal 2
<TERMINAL_3>555</TERMINAL_3>	All orders from Waiter number 555 -> terminal 3
<TERMINAL_10>public</TERMINAL_10>	All orders from public will go to terminal 10

2 - Starting AirMenu connection.

```
int __stdcall radio_init (int com_no,int first_terminal, int last_terminal, license_no int, int protocol)
```

Description:

This function initializes AirMenu. The computer MUST BE CONNECTED TO THE INTERNET.

Function:

```
int radio_init (  
int com_no,  
int first_terminal,  
int last_terminal,  
int license_no,  
int protocol);
```

Parameters:

com_no: Not Used, may be : 1

first_terminal: This is the terminal number the program will receive with AirMenu.

last_terminal: Not Used,

license_no: Should be the same on the Software

protocol: Should be :4

```
int __stdcall radio_close ()
```

Description:

This function closes the connection

3 - Creating the items tree

int __stdcall radio_make_tree_mkdir (char * name)

Description:

Equivalent to the MSDOS mkdir, creates a branch in the tree.

Parameters:

* name: Name of class to create.

Returns:

not used

int __stdcall radio_make_tree_cd (char * name)

Description:

Equivalent to the MSDOS cd, changes to a branch in the tree. It will create one if that doesn't exist.

Parameters:

* name: Name of class to create.

Returns:

not used

int __stdcall radio_make_tree_cd_back ();

Description:

Equivalent to the cd .. of MSDOS, moves to the previous branch.

Returns:

not used

int __stdcall radio_make_tree_entry_price(int entry_no, char *name,char *price)

Description:

Allows to introduce a particular branch of the tree.

Parameters:

entry_no: Plu value

* name: Plu Description.

* price: Plu Price.

Returns:
not used

4 - Modifiers

```
int __stdcall radio_make_tree_modifier_entry (entry_no int, char * name);
```

Description:

The previous directory becomes the name of the group of supplements.
With this instruction we add elements to the group.

Example:

```
radio_make_tree_cd "temperature of the cup"  
radio_make_tree_modifier_entry 600, "Cold Chevena"  
radio_make_tree_modifier_entry 601, "Hot cup"  
radio_make_tree_modifier_entry 602, "Morna cup"
```

We would create a new group called "temperature of the cup", with 3 elements.

Returns:

not used

```
int __stdcall radio_make_tree_modifier_min_max( char *group_name,int automatic,int min,int max,int  
default_value);
```

Description:

Assigns a modifier group to a menu item. More than one modifier group can be assigned to an item by calling this function repeatedly.

Parameters:

group_name: Title of the modifier group to be assigned.

automatic: not used

min: Minimum number of items that the end user must select in the group.

max: Maximum number of items that the end user must select in the group.

default_value: not used

5 - Sending the items to AirMenu

```
int __stdcall radio_upload_tree2 (char * Settings,int NumParametros, int RearrangeTree);
```

Description:

Sends all items to AirMenu.

Parameters:

not used

6 - Receiving orders

```
int __stdcall radio_recv5(int *function, int *terminal, int *license, int *msg_number, int *numberOfItems,int
*table1, int *table2,int *AuxData,int *place1, int *place2,int *plus,int *qtDs,unsigned char *st_table1,unsigned
char *st_table2)
```

Description :

This function checks if there is any message sent by AirMenu in the input buffer to be read by the program. If there is any message, the function makes a copy of this and sends to the variables passed as parameters.

It should be called by the program 10 times / second.

Return:

Parameters:

function: type of Message.

'P' -> Order

'C' -> Bill

terminal: will return 'first terminal', not to be used in this context

license: license number

msg_number: Message number

numberOfItems: Number of items

table1: Table number

table2: not used

place1: not used

place2: not used

int *plus is a pointer to an array of integers of 32 bits.

int *qtDs is a pointer to an array of integers of 32 bits.

st_table1: not used

st_table2: not used

ATTENTION, it can only be used with the protocol 4 active

This Function is identical to the radio_recv4, but when receiving decimal numbers, it returns a quantity in negative multiplied by 1000. Example 0.1, will return the quantity of -100.