Classic-Ind-MC SDK Function Specification

Visual Basic

Ver. 1.0

Revision History

Rev. No.	Rev. Date	Revised Description	Person in charge	Remarks
1	2009/10/31	Created	Anuj Chaudhary	Document created
2	2009/10/31	Final Review		Review

TABLE OF CONTENTS

1 OBJECTIVE 5

2 INTRODUCTION 5

3 FUNCTIONAL OVERVIEW OF LIBRARY 5

4 STRUCTURE DETAILS 6

4.1 STRUCT SDF 6

4.2 STRUCT UDS 6

4.3 STRUCT MDF 6

5 RETURN VALUE 8

6 FUNCTION DETAILS 9

6.1 DEVICE_CONNECT() 9

6.2 GET_VERSION () 9

6.3 BIO_VERSION () 9

6.4 BIO_TEMP_CMD() 10

6.5 GET_FINGER() 10

6.6 CARD_LIST() 11

6.7 GET _DATE_DATA() 12

6.8 CAPTURE_DATA() 12

6.9 MEMORY_PERCENT() 12

6.10 GET_DATE_TIME() 13

6.11 DEVICE_CLOSE() 13

6.12 SET_DATE() 14

6.13 SET_TIME() 14

6.14 DELETE_CARD_LIST() 14

6.15 GET_DATA() 14

6.16 DATE_DELETE() 14

6.17 RESTART() 15

6.18 SET_IPADD() 15

6.19 SET_GATEWAY() 15

6.20 SET_NETMASK() 16

6.21 ENROLL_TEMPLATE() 16

6.22 FINGER_DELETE() 16

6.23 WRITE_CARD () 18

6.24 WRITE_TEMP_CARD () 18

6.25 WRITE_TEMP_CARD () 19

6.26 RESET_MACHINE () 19

7 FUNCTION DETAILS **20**

7.1 FROM_TO_CC() 20

7.2 SERVER_MAIL_ID() 20

7.3 EPORT_NUM() 21

7.4 GET_EMAIL_SETTING() 21

1 Objective

The main purpose of this document is to provide the detail of program functions of the CLASSIC-IND-MC SDK. This document can be used by developer for understanding, analysing, improving and modifying the source code of this library files.

2 Introduction

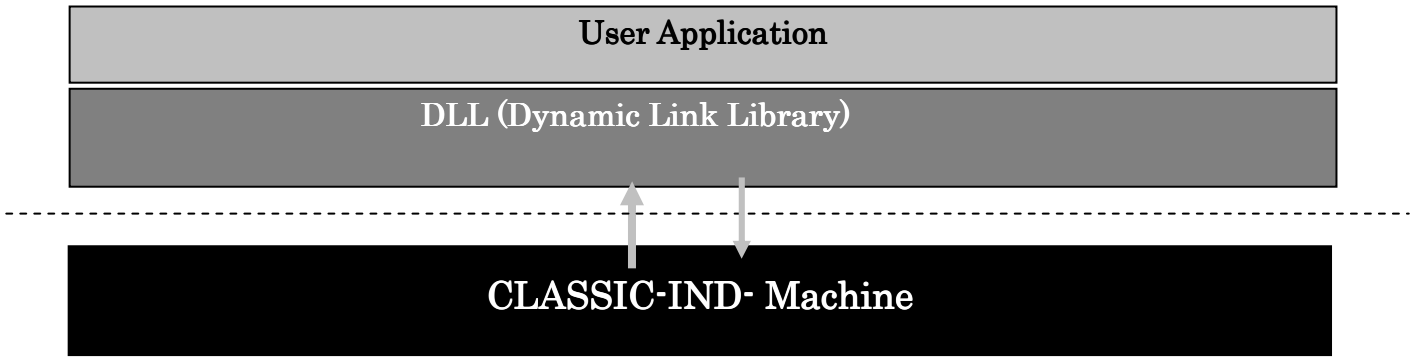
Classic-Ind-MC SDK is a software development kit that use for user develop application program. SDK provide to user in dynamic-link library document form.

When user using Classic-Ind-MC SDK to develop their own application development platform, user will be able to complete their application development in high efficiently and correctly based on Classic-Ind-MC SDK. SDK support VB Builder development. Demo program source code of VB version is available now, please contact with our engineer if have any needs.

SDK development guide is a reference manual for user secondary development. After review this manual, user will be able to solve their problem in fast way during their development.

3 Functional Overview of Library

Classic-Ind-MC DDL, used to implement user application interface for creating font end application. The various interfaces (function, Structure) for writing applications that are used for accessing and manipulations of data in a machine.
Layered diagram of DDL and its interaction with application is as follows.



Picture 1

4 Structure Details

4.1 struct SDF

This type of structure returns data from the machine and size of the data.		
Member Name	Type	General Description
Data2[2048]	char	data from dll
size	int	length of return data

4.2 struct UDS

This type of structure returns status from the machine and its corresponding error code		
Member Name	Type	General Description
errcode	char	string variable
Err_status	int	status code

4.3 struct MDF

This type of structure returns data string, return size, status and total size of data		
Member Name	Type	General Description
data [2048]	char	data from dll
Ret_size	int	length of return data
ret_status	int	status code
buff_size[10]	char	total length of data

If function's return type is MDF structure, then its return data will be more then 2048 bytes in some function, which names are mention below.

Those functions, in which return data size is more then 2048 bytes, after completion of function user have to use **get_data()** function untill length of return data becomes

equals to total length of data.

- bio_temp_cmd ()
- card_list ()
- get_date_data ()
- capture_data ()

Example:

```
Dim Struct_Info As uds
Dim Strdata As String
Dim Struct_Getdata As GetData
T_Data = ""
TotalCount = 0
Dim i As Integer
Struct_Info = used_function()
If Struct_Info.status = 0 Then
   Strdata = Struct_Info.Data
   T_Data = T_Data & Strdata
   TotalCount = Struct_Info.size
   If Struct_Info.size = CDbl(Struct_Info.TotalSize) Then
   Else
      Do While i = 0
         Struct_Getdata = get_data()
         TotalCount = TotalCount + Struct_Getdata.size
         T_Data = T_Data + Struct_Getdata.PickData
         If CDbl(Struct_Info.TotalSize) = TotalCount Then
            i = 1
         End If
      Loop
   End If
   write_info (T_Data)
   save_finger (T_Data)
   write_info ("Size of Data is -:" & Struct_Info.TotalSize)
ElseIf Struct_Info.status = 1 Then
   write_info ("Finger data is not avilable")
Else
   write_info ("Connection Problem")
End                                                    If
```

5 Return Value

There is three diffrent value in return status. If retern value is 0, it mean command successfully execute. If return value is 1, it mean error in execution of command. If retern value is 2, it means network error occured.

Value	General Description
0	Success
1	Some error occured
2	Problem in Network Connection

6 Function Details

6.1 device_connect()

Function Name	**device_connect()**
Format	**int device_connect (char* ipadd, int port)**
Argument	**ipadd:** IP address of a device **port:** Port number of a device
Functionality	This function **device_connect()** is used to connect the machine.
Return Value	Upon successful completion it returns 0. Otherwise, it returns a non-zero error value.

6.2 get_version ()

Function Name	**get_version()**
Format	**MDF get_version()**
Argument	none
Functionality	The **get_version()** function uses to get the version and basic setting of a machine.
Return Value	Return a MDF structure - **Ret_status:** Upon successful completion it returns 0. Otherwise, it returns non-zero value **Data:** Multiline text string. **Ret_size**: Size of data which return by **get_version ().** **Buff_size:** Total size of data which get by machine in function.
Return Format	MMC MEMORY SERIAL INTERFACE BAUD RATE - 115200 IP ADDRESS IS: 192.168.000.088 NETMASK ADDRESS: 255.255.255.000 GATEWAY ADDRESS: 192.168.000.001 MAC ADDRESS:00-50-C2-3A-3F-09 BIOLINK BAUD RATE1-115200 SMART CARD - W/A PIN FINGER 1:N Serch

6.3 bio_version ()

Function Name	**Bio_version ()**
Format	**MDF bio_version ()**
Argument	none
Functionality	The **bio_version ()** function uses to get the Bio-metric version and basic setting of a machine.
Return Value	Return Structure **MDF** - **Ret_status:** Upon successful completion it returns 0. Otherwise, it returns non-zero value.

	Data: Multiline text string. **Ret_size**: Size of data which return by **bio_version ().** **Buff_size:** Total size of data which get by machine in function.
Return Format	FP Version 01.02 TOTAL FINGER 040
Note	Biometric Sencer should be available.

6.4 bio_temp_cmd()

Function Name	**bio_temp_cmd()**
Format	**MFD bio_temp_cmd(char * cardnum)**
Argument	**Cardnum :** Eight digit card number to get finger
Functionality	The **bio_temp_cmd()** functions reads the data of template. It is used to download the template data.
Return Value	Return Structure MDF - **Ret_status:** Upon successful completion it returns 0. Otherwise, it returns non-zero error value: **Data:** Multiline text string. **Ret_size**: Size of data which return by **bio_temp_cmd().** **Buff_size:** Total size of data which get by machine in function.
Return Format	X:cccccccc:ffff……..fffffssN Where X – number of finger c – Card number of max 8 digit f – finger data(768 bytes) s – cheak sum(2 bytes) N – fixed by command
Note	Biometric Sencer should be available.

6.5 get_finger()

Function Name	**template_read ()**
Format	**MDF template_read ()**
Argument	**none**
Functionality	The **template_read ()** functions read the data of finger, which is putting on biometric sencer after execution of this function.
Return Value	Return Structure MDF - **Ret status:** Upon successful completion it returns 0. Otherwise, it returns non-zero error value: **Data:** text string of 385 bytes. **Ret_size**: Size of data which return by function **template_read ().** **Buff_size:** Total size of data which get by machine in function.
Return Format	xxx….xxxcN 384 bytes of finger data –xxx….xxx 1 bytes of check sum - cc
Note	Biometric Sencer should be available.

6.6 card_list()

Function Name	**card_list ()**
Format	**MDF card_list ()**
Argument	none
Functionality	The **card_list ()** function is used to get card list from machine.
Return Value	Return Structure MDF - **Ret_status:** Upon successful completion it returns 0. Otherwise, it returns non-zero error value: **Data:** Multiline text string in format**-**ccccccccc:fffff:X c - 8 digit card number f – 5 digit data **Ret_size**: Size of data which return by function **card_list ().** **Buff_size:** Total size of data which get by machine in function.

6.7 get _date_data()

Function Name	**get_date_data()**
Format	**MDF get_date_data()**
Argument	**none**
Functionality	The **get_date_data()** function get the date list of punching data which, is available in the machine in "**ddmmyy**" format.
Return Value	Return Structure MDF - **Ret_status:** Upon successful completion it returns 0. Otherwise, it returns non-zero error value: **Data:** Multiline text string. **Ret_size**: Size of data which return by function **get_date_data ().** **Buff_size:** Total size of data which get by machine in function.
Return Format	ddmmyy..ddmmyy……..ddmmyy

6.8 capture_data()

Function Name	**capture_data()**
Format	**MDF capture_data(char * date)**
Argument	**date:** Date for which punching data needs in **ddmmyy** format.
Functionality	The **capture_data ()** functions read the punching records of given date from the machine.
Return Value	Return Structure MDF - **Ret_status:** Upon successful completion it returns 0. Otherwise, it returns non-zero error value: **Data:** Multiline text string in format **"hhmmxxxxxxxx"**. Where- h -> Hour m -> Minute x -> Card Number. **Ret_size**: Size of data which return by function **capture_data ().** **Buff_size:** Total size of data which get by machine in function.

6.9 memory_percent()

Function Name	**memory_percent()**
Format	**MDF memory_percent ()**
Argument	**None.**
Functionality	The **memory_percent ()** read the percentage of memory of machine.
Return Value	Return Structure MDF - **Ret_status:** Upon successful completion it returns 0. Otherwise, it returns following non-zero error value: **Data:** Text string in format "**xxx**". **Ret_size**: Size of data which return by function **memory_percent ().** **Buff_size:** Total size of data which get by machine in function.

6.10 get_date_time()

Function Name	**get_date_time()**
Format	**MDF get_date_time ()**
Argument	**none**.
Functionality	The **get_date_time()** function read the date and time of machine and return it in a format of *ddmmyyhhmmss*.
Return Value	Return Structure MDF - **Ret_status:** Upon successful completion it returns 0. Otherwise, it returns following non-zero error value: **Data:** Text string in format "*DDMMYYHHMMSS*". **Ret_size**: Size of data which return by function **get_date_time ().** **Buff_size:** Total size of data which get by machine in function.

6.11 device_close()

Function Name	**device_close()**
Format	**int device_close ()**
Argument	**None**.
Functionality	The **device_close ()** function closes the network connection.
Return Value	Upon successful completion it returns 0. Otherwise, it returns a non-zero error value.

6.12 set_date()

Function Name	**set_date()**
Format	**int set_date (char * date)**
Argument	**date:** string date (**ddmmyy**) to change.
Functionality	The **set_date()** used to change the date of machine.
Return Value	Upon successful completion it returns 0. Otherwise, it returns 1 for error and 2 for connection error value.

6.13 set_time()

Function Name	**set_time()**
Format	**int set_time (char * time)**
Argument	**time:** time(HHMMSS).
Functionality	The **set_time()** used to change the time of machine.
Return Value	Upon successful completion it returns 0. Otherwise, it returns 1 for error and 2 for connection error value.

6.14 delete_card_list()

Function Name	**delete_card_list()**
Format	**int delete_card_list()**
Argument	**none**
Functionality	The **delete_card_list()** function deletes the card list from machine.
Return Value	Upon successful completion it returns 0. Otherwise, it returns 1 for error and 2 for connection error value.

6.15 get_data()

Function Name	**get_data()**
Format	**SDF get_data()**
Argument	none
Functionality	The **get_data()** gets the remaining data of some commands which size is more then 2048 bytes.
Return Value	Structure sdf- **data:** Multiline text string. **size:** Size of return data.

6.16 date_delete()

Function Name	**date_delete()**
Format	**int date_delete (char * date)**
Argument	**date**: date string (**DDMMYY**), which data will delete.
Functionality	The **date_delete()** function deletes the punching data of given date from machine.
Return Value	Upon successful completion it returns 0. Otherwise, it returns 1 for error and 2 for connection error value.
Special Note	**Appearing data will never delete.**

6.17 restart()

Function Name	**restart()**
Format	**int restart ()**
Argument	**none**
Functionality	The **restart()** function uses to restart the device.
Return Value	Upon successful completion it returns 0. Otherwise, it returns 1 for error and 2 for connection error value.

6.18 set_ipadd()

Function Name	**set_ipadd()**
Format	**UDS set_ipadd (char * ipadd)**
Argument	**ipadd:** IP Address in format "**xxx.xxx.xxx.xxx**".
Functionality	The **set_ipadd()** used to change the IP address of a machine.
Return Value	Structure uds- **Errcode:** Return status of command. **Err_status:** Upon successful completion it returns 0. Otherwise, it returns 1 for error and 2 for connection error value.

6.19 set_gateway()

Function Name	**set_gateway()**
Format	**UDS set_gateway (char * gateway)**
Argument	**Gateway:** gateway address in format "**xxx.xxx.xxx.xxx**".
Functionality	The **set_gateway ()** function changes the gateway address of a machine.
Return Value	Structure uds- **Errcode:** Return status of command. **Err_status:** Upon successful completion it returns 0. Otherwise, it returns 1 for error and 2 for connection error value.

6.20 set_netmask()

Function Name	**set_netmask()**
Format	**uds set_netmask (char * netmask)**
Argument	**netmask:** netmask address in format "**xxx.xxx.xxx.xxx**".
Functionality	The **set_netmask ()** function changes the net mask address of a machine.
Return Value	Structure uds- **Errcode:** Return status of command. **Err_status:** Upon successful completion it returns 0. Otherwise, it returns 1 for error and 2 for connection error value.

6.21 enroll_template()

Function Name	**enroll_template**
Format	**uds enroll_template(char* empname, char* empcard, char* temp)**
Argument	**empname:** Employee name(maximum 16 digit) **empcard:** Card number (staff number) of employee **Temp**: File name, in which finger date is stored.
Functionality	The **enroll_template ()** function is used to upload the finger in machine.
Return Value Note	Structure uds- **Errcode:** Return status of command. **Err_status:** Upon successful completion it returns 0. Otherwise, it returns 1 for error and 2 for connection error value. Biometric Sencer should be available.

6.22 Finger_delete()

Function Name	**finger_delete()**
Format	**int finger_delete (char * card)**
Argument	**card:** 8 digit card number.
Functionality	The **finger_delete()** used to delete finger of given card number from machine.
Return Value Note	Upon successful completion it returns 0. Otherwise, it returns 1 for error if finger is not avilable and 2 for connection error value. Biometric Sencer should be available.

6.23 Write_Card ()

Function Name	**Write_Card ()**
Format	**int Write_Card (char *card)**
Argument	**card:** 8 digit card number.
Functionality	The **Write_Card ()** function uses to write the card number in card by the device.
Return Value	Upon successful completion it returns 0. Otherwise, it returns 1 for error and 2 for connection error value.

6.24 write_temp_card ()

Function Name	**write_temp_card ()**
Format	**uds write_temp_card (char * empname, char* empcard, char * fingerdata).**
Argument	**empname:** Employee name(maximum 16 digit) **empcard:** Card number (staff number) of employee (maximum 8 digit). **fingerdata**: File name, in which finger date is stored.
Functionality	The **write_temp_card ()** function changes the net mask address of a machine.
Return Value Note	Structure uds- **Errcode:** Return status of command. **Err_status:** Upon successful completion it returns 0. Otherwise, it returns 1 for error and 2 for connection error value. Biometric Sencer should be available.

6.25 write_temp_card ()

Function Name	**write_temp_append ()**
Format	**uds write_temp_append (char* empcard, char * fingerdata).**
Argument	**empcard:** Card number (staff number) of employee (maximum 8 digit). **fingerdata**: File name, in which finger date is stored.
Functionality	The **write_temp_card ()** function changes the net mask address of a machine.
Return Value	Structure uds- **Errcode:** Return status of command. **Err_status:** Upon successful completion it returns 0. Otherwise, it returns 1 for error and 2 for connection error value.
Note	Biometric Sencer should be available.

6.26 Reset_Machine ()

Function Name	**Reset_Machine ()**
Format	**int Reset_Machine ()**
Argument	**none**
Functionality	The **Reset_Machine ()** function uses to reset the device.
Return Value	Upon successful completion it returns 0. Otherwise, it returns 1 for error and 2 for connection error value.

7 Function Details

7.1 From_to_cc()

Function Name	**from_to_cc()**
Format	**UDS from_to_cc(char * from, char * to, char * cc)**
Argument	**from:** mail-id of sender (max length 64 bytes). **To:** mail-id of receiver (max length 64 bytes). **Cc:** mail-id of carbon copy (max length 64 bytes).
Functionality	The **from_to_cc ()** function is used to set the mail- id for mailing in machine.
Return Value	structure uds- **Err_status:** Return status of command. **Err_status:** Upon successful completion it returns 0. Otherwise, it returns 1 for error and 2 for connection error value.
Special Note	If more then one mail-id are available, Then separate it by **";"**.

7.2 server_mail_id()

Function Name	**server_mail_id()**
Format	**UDS server_mail_id (char * server_ip, char * pwd, char * username)**
Argument	**server_ip:** IP Adress of server in format "**xxx.xxx.xxx.xxx**". **pwd:** Password of user (max length 32 bytes). **username:** Name of user of server (max length 32 bytes).
Functionality	The **server_mail_id ()** function is used to set the mail server ip, user and password in machine.
Return Value	Structure uds- **Err_status:** return status of command. **Err_status:** Upon successful completion it returns 0. Otherwise, it returns 1 for error and 2 for connection error value.

7.3 eport_num()

Function Name	**eport_num()**
Format	**UDS eport_num(char * portnum)**
Argument	**portnum:** port number to send mail (max length 2 bytes(integer)) as string**.**
Functionality	The **server_mail_id ()** function is used to set the mail port in machine.
Return Value	structure uds- **Err_status:** return status of command. **Err_status:** Upon successful completion it returns 0. Otherwise, it returns 1 for error and 2 for connection error value.

7.4 get_email_setting()

Function Name	**get_email_setting()**
Format	**MDF get_email_setting()**
Argument	**none**
Functionality	The **get_email_setting** function reads the all email settings for mailing in machine.
Return Value	Return Structure MDF - **Ret_status:** Upon successful completion it returns 0. Otherwise, it returns following non-zero error value: **Data:** string variable of version. **Ret_size**: size of data which return by **get_version ().** **Buff_size:** Total size of data which get by machine in function.
Return Format	E-MAIL - DISABLED gyan@g.com deep@star.com piyush@atsr.co.in,d@s.com 192.168.000.012 gyanendra sachan 515689 E-MAIL PORT : 1085 Location ID:ABC