

# nexgo\_mdbClient\_interface

Revisions.....	2
Introduction.....	4
Interface description.....	5
onInit.....	5
onStart.....	5
onFinish.....	6
onClose.....	6
setResult().....	6
reportToVMC.....	6
getMDBReceiveTimeStamp.....	6
Callback Interface Description.....	6
ICompletionCallback.....	6
IPaymentCallBack.....	7
Appendix 1.....	8
A1.1 PaymentAppInfo.....	8
A1.2 Transacrion Result.....	8
A1.3 onPay Param Description.....	9
A1.4 ErrCode Descrition.....	9
A1.5 mdbService Status.....	9
A1.6 vend result.....	10
A1.7 vend type.....	10
A1.8 vend param.....	10
A1.9 vending Info.....	10
Appendix 2.....	11
B1.1 ReportKey.....	11
B1.2 Param of transaction result to VMC.....	11

## Revisions

### Version 2.0.5

**There are five changes as noted below.**

1. add the onReaderCancel function to notify user cancel from vending machine;
2. add the getMDBReceviceTimeStamp function

### Version 2.0.4

**There are five changes as noted below.**

3. add the onCallDiagnostics function to notify user to respond 17H FFH command;
4. add the setManufactureCode parameter to set manufacture code in 17H 00H command;
5. Add TRANS\_RESULT\_AMOUNT parameter when payment app send trans result to the VMC;
6. add setSupportMultiVend parameter to set whether support multi-vend;
7. add setSupportRemoteVend parameter to set whether support remote-vend;

### Version 2.0.3

**There is three change as noted below.**

1. add the setWaitEndSessionTm parameter to support send cancel session automatic;
2. Change ICompletionCallback to aidl interface ;
3. add onNotifyVMCIInfo() function to get the brand and model of VM.

### Version 2.0.2

**There are three changes as noted below.**

4. add the setSupportCashSale parameter to support cash sale command.
5. add the setDeviceType parameter to set the cashless device type.
6. add the setDelayTm parameter to set the delay of MDB command respond time.

### Version 2.0

**There are two changes as noted below.**

7. add the respondAckFirst parameter to support ack first when receive the cmd.
8. change the log maxFileSize of mdbService.apk to 5MB and totalSizeCap to 35MB.

## **Version 1.9**

**There are are three changes as noted below.**

1. add the deviceMode parameter to support man in the middle mode
2. add the notifyStatus parameter to support whether notify the MDB status in the screen.
3. add the onPayResult function to get the vending result.

## **Version 1.8**

**There are are two changes as noted below.**

1. add the revalueApproved parameter to respond the ReValue command
2. add the requestRefunds parameter to support Request Refunds command.

## **Version 1.7**

**There are are three changes as noted below.**

1. add the notifyVendParam() to notify the vending machine param to payment application.
2. add the cancel Session Request command in the mdbService.apk.
3. add the ReportType.CANCEL\_SEESION\_REQUEST in the mdbclient.jar.

## **Version 1.6**

**There are are four changes as noted below.**

1. support Level 1 and 2.
2. support begin session/ always idle mode.
3. support to set Feature Level.
4. support to set scale factor.

## **Version 1.5**

**There are are five changes as noted below.**

1. setResult() function is deprecated.
2. onInit() function is deprecated.
3. onFinsih() function is deprecated.
4. Add reportToVMC() function replace setResult() function.
5. Add onAppUpdate() function.

## Introduction

### **mdbClient jar**

The mdbClient jar is responsible for connecting to mdbService, exchanging data between the mdbService and Payment App.

**mdbService.apk must be installed in the UN20 first.**

### **Default configuration parameters of UN20 device**

Mdb Version:4-2

VMC Feature Level:3

Scale Factor:1

Miscellaneous Options: 0x00

1. The payment media reader is capable of restoring funds to the user's payment media or account. Refunds may be requested.
2. The payment media reader is NOT multivend capable. Terminate session after each vend.
3. The payment media reader does NOT have a display.
4. The payment media reader does support the VEND/CASH SALE subcommand..

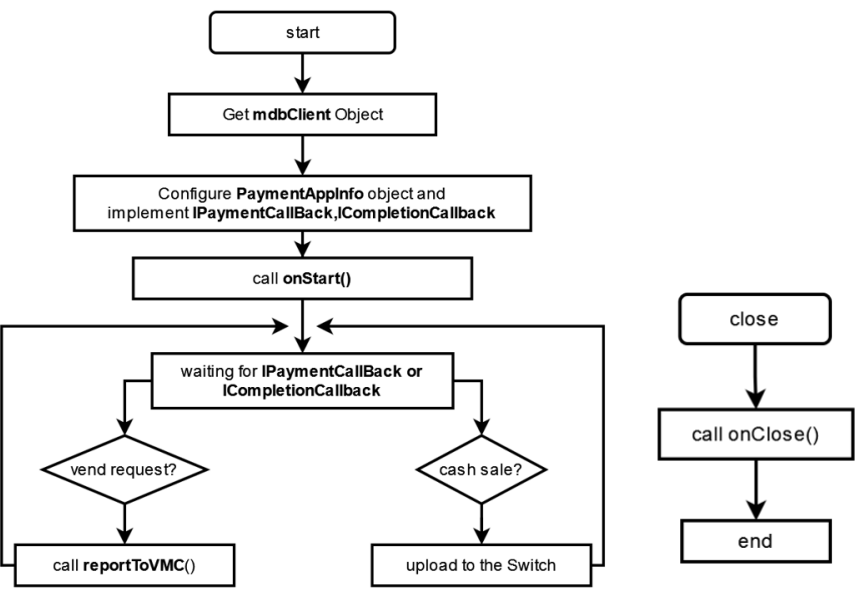
### **The library that build.gradle need to import:**

```
implementation files('libs/mdbclient_v2.0.2')  
implementation 'com.google.code.gson:gson:2.8.5'
```

### **The permission that AndroidManifest.xml need to add:**

```
<uses-permission  
android:name="android.permission.QUERY_ALL_PACKAGES"/>
```

This module operates using the basic flow chart.



## Interface description

onInit

onStart

configure the mdbService.

```
void onStart(Context context, PaymentAppInfo info,  
             IPaymentCallBack callback,ICompletionCallback iCompletionCallback);
```

Paramters:

Parameter	Description
context	android context
PaymentAppInfo	Configuration parameters of mdbService <a href="#">A1.1 PaymentAppInfo</a>
IPaymentCallBack	the callback interface of mdbService
ICompletionCallback	mdbService call result

return value:none.

**onFinish**

**onClose**

unbind mdbService and kill the process of mdbService.

void onClose();

return value:none

**setResult()**

**reportToVMC**

set the transaction result to the mdbService.

void reportToVMC (String reportData); [B1.2 Param of transaction result to VMC](#)

return value:none.

**getMDBReceiveTimeStamp**

get timestamp when receive the mdb data from vending machine

long getMDBReceiveTimeStamp();

return value:long.

## Callback Interface Description

### ICompletionCallback

The callback interface after mdbService binding succeeded or failure.

#### **onSuccess**

call mdbService success.

#### **onFailure**

call mdbService failure and notify the errCode. [A1.4 ErrCode Description](#)

#### **onAppUpdate**

mdbService update to a new version, the user needs to restart the mdbService

### **onNotifyVMCInfo**

notify the VMC brand and model

## **IPaymentCallBack**

The callback interface of mdbService

### **onPay**

The vending machine initiates a vending request, and the mdbservice calls the onPay.

public int onPay(int tradeType,String itemprice,String itemNumber);

[A1.3 onPay Param Description](#)

[A1.7 vend type](#)

### **notifyMdbStatus**

The status of mdbService.

public void notifyMdbStatus(int status); [A1.5 mdbService Status](#)

### **notifyVendResult**

The vend status between Vending Machine and the Payment App.

public void notifyVendResult(int result); [A1.6 vend result](#)

### **notifyVendParam**

Notifying the parameter of vending machine.

public void notifyVendParam(int type,byte[] data); [A1.8 vend param](#)

### **onPayResult**

get the vending info in JSON String Format.

public void onPayResult(String payResult); [A1.9 vending Info](#)

### **onCallDiagnostics**

notify the user data, and respond the diagnostic user defined data to VMC.

public void onCallDiagnostics(byte[] data);

### **onReaderCancel**

notify the user the event when user select cancel in the vending machine.

public void onReaderCancel();

# Appendix 1

## Param Description

---

### A1.1 PaymentAppInfo

---

Attributes	Description
clientId	The client id of brand,like "nexgo"
model	The model of device
countryCode	Country / Currency Code - String.Like "0840"
decimalPlaces	The number of decimal places used to communicate monetary values between the VMC and the payment media reader.0x02 is default value.
responseTime	Application Maximum Response Time - seconds.
logLevel	DIABLE,DEBUG,INFO,WARN,ERROR,ALL
readerFeatureLevel	LEVEL_1, LEVEL_2, LEVEL_3
enableAlwaysIdle	0-didn't support always idle 1-support always idle in LEVEL_3
scaleFactor	set scale factor
monetaryFormat	set 0 as default value
revalueApproved	true: Revalue Approved false: Revalue Denied
requestRefunds	true: support request refunds false: didn't support
deviceMode	CASHLESS: cashless device mode, TRANSMIT: man in the middle mode
notifyStatus	true: show MDB status in the screen false: didn't show the message
respondAckFirst	true: ack first when receive the command false: ack data directly
supportCashSale	true:support cash sale, false:didn't support cash sale
deviceType	0 - cashless device #1(1xH), 1-cashless device #2(6xH)
delayTm	delay time(ms). 0 is default value ,if set this parameter > 0,the mdb communication will delay to respond data
waitEndSessionTm	If this time is exceeded, UN20 automatically initiates cancellation sessions to avoid situations where vending machines do not issue vend cancellations. If set 0, the parameter didn't work
manufactureCode	Payment application can use this parameter to respond the manufacture code which send to VMC in the 17H 00H command.
supportMultiVend	Set Cashless device to support multi-vend(also need VMC support).
supportReoteVend	Set Cashless device to support remote-vend(also need VMC support).

### A1.2 Transacrion Result

---

result	Description
--------	-------------

TRADE_UNKNOWN	UNKNOWN
TRADE_SUCCESS	The transaction of payment App is approve
TRADE_FAILURE	The transaction of payment App is failure
TRADE_CANCEL	The transaction is cancel

### A1.3 onPay Param Description

---

Parameter	Description
tradeType	Cash Sale or Vend Request
itemPrice	item amount form vend machine
itemNumber	item number

### A1.4 ErrCode Description

---

errCode	Description
0	success
-1001	device didn't support mdb protocol
-1002	parameter error
-1003	the mdbService is busy
-1004	device error

### A1.5 mdbService Status

---

status	Description
UNKNOWN	UNKNOWN
INACTIVE	INACTIVE
DISABLE	DISABLE
ENABLE	ENABLE
SESSION_IDLE	SESSION_IDLE
VEND	VEND
VEND_CANCEL	VEND_CANCEL
END_SESSION	END_SESSION

SERVICE_ERR	SERVICE ERROR
-------------	---------------

## A1.6 vend result

---

result	Description
VEND_SUCCESS	Vending Machine vend successful
VEND_FAILURE	Vending Machine vend failure
VEND_CANCEL	Vending Machine vend cancel
VEND_END_SESSION	the vend session end

## A1.7 vend type

---

type	Description
VEND_REQUEST	Vending request
CASH_SALE	Cash Sale

## A1.8 vend param

---

type	Description
MAX_PRICE	max price of the vending machine(0000~FFFF)
MIN_PRICE	min price of the vending machine(0000~FFFF)
IDLE_MODE	the idle mode of vending machine(0-idle 1-always dille)

## A1.9 vending Info

---

type	Description
itemSlot	the item slot
itemInfo	A list including itemAmount and paymentMethod
vendStatus	the status of vend result
machineStatus	the status of machine

# Appendix 2

## Param Description

### B1.1 ReportKey

Attributes	Description
TYPE	Report type to VMC
TRADE_RESULT	Transaction result
FUNDS	Funds of remote vend
ITEMNUMBER	Item number of remote vend
ITEMOPTION	Item option of remote vend

### B1.2 Param of transaction result to VMC

param	mandatory	type	description
TYPE	Y	ReportType	type of trans result
TRADE_RESULT	Y	TransResult	result of the trans
TRADE_RESULT_AMOUNT	Y	String	The trans result amount
USER_DEFINED_DATA	O	String	User defined data

e.g.

```
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    mLog.debug("onActivityResult requestCode:{} resultCode:{}",requestCode,resultCode);
    if (requestCode == 200&&resultCode==1) {
        if(data!=null){
            final int result = data.getIntExtra( name: "TransResult",TransResult.TRADE_UNKNOWN.ordinal());
            final String transAmount = data.getStringExtra( name: "TransAmount");
            mLog.debug("result:{} transAmount:{}",result,transAmount);
            mInitParamHandler.post(() -> {
                JSONObject report = new JSONObject();
                try {
                    report.put(ReportKey.TYPE,ReportType.TRADE_RESULT.ordinal());
                    report.put(ReportKey.TRADE_RESULT,result);
                    report.put(ReportKey.TRANS_RESULT_AMOUNT,transAmount);
                    MdbServiceManager.getInstance().reportToVMC(report.toString());
                } catch (JSONException e) {
                    e.printStackTrace();
                }
            });
        }
    }
}
```