Worldpay Integration Using C++ SDK
Quick Start

Obtain IDTech C++ SDK Library version 1.0.35.030 or later.

Using the documentation provided in that package, and the demo code for examples, develop your project for your target platform. You should be able to successfully connect to the device, and execute any of the supported API commands.

Processing transaction with WorldPay will require you establish the callback to receive the WorldPay transaction results, and then initiating the transaction by executing the specified API command.

While the final results will be returned back in the WorldPay callback, it is advisable to implement all the standard transaction callbacks the SDK supports, as those callbacks may receive intermediate data or error conditions during the transaction. For example, if you were to execute a WorldPay transaction on a device without a LCD display (VP3300), you will be getting the transaction LCD callback messages through the regular EMV callback, as if you were to execute a standard, non WorldPay transaction request.

The following implementation can be seen in source code file included with SDK:
C++_Demo_Source->VP3300_Demo->_VP3300_demo_device.c

IMPLEMENTATION:

Create the callback that will receive the final results and add it to your project:

```
void WorldPayCallback(char* message, int returnCode, int emvCode){

        printf("WORLDPAY RESPONSE>>>>>>>\n");
        printf("Return Code = %d\n", returnCode);
        printf("EMV Code = %d\n", emvCode);
        printf("Message = \n%s\n", message);
}
```

Create a WorldPay data object and populate it with transaction details

```
WorldPayData data;
strcpy(data.accountID,"1188346\0");
strcpy(data.accountToken,"E1EB3EFB049DFB599F1CB454E1CFC4FD14BF90BCE74456AE9E9490D7D609B466C81A3
801\0");
strcpy(data.acceptorID , "364798674\0");
strcpy(data.license , "2y2r0h6sorxb-5ufv0z2ao3jb \0"); //  if live/production transaction, a valid license must be
provided
data.transactionType = TRANSACTION_TYPE_SALE;
//data.transactionType = TRANSACTION_TYPE_PRE_AUTH;
//data.transactionType = TRANSACTION_TYPE_COMPLETION;
//data.transactionType = TRANSACTION_TYPE_VOID;
//data.transactionType = TRANSACTION_TYPE_RETURN;
//data.transactionType = TRANSACTION_TYPE_QUICK_CHIP_START;
//data.transactionType = TRANSACTION_TYPE_QUICK_CHIP_FINISH;

strcpy(data.amount , "22.00\0");    //specify transaction Amount
data.timeout = 30;   //Timeout for transaction
data.isTest = 1;  //if test transaction, set to TRUE.  If live transactions, set to FALSE
data.enableCTLS = 1;   // Enables Contactless Transactions. data.msrOnly should be 0
data.msrOnly = 0; //for EMV + MSR, set to FALSE.  If MSR swipe only, set to TRUE. data.enableCTLS should be 0
strcpy(data.transactionID , "3830\0"); //required if COMPLETION, VOID, RETURN.  Otherwise, leave blank
strcpy(data.referenceNumber , "112233445566\0"); //required value. Set a reference number
strcpy(data.terminalID , "12345\0"); //required value.  Set a terminal identifer
strcpy(data.ticketNumber , "54321\0"); //required value. Set a ticket number
data.duplicateCheck = 0;  //set to TRUE if you want to check for duplicate transactions
data.duplicateOverride = 1;  //set to TRUE if you want to allow duplicate charges to same card
data.declinePartial = 1;  //set to TRUE if you want not allow partial approvals
```

Execute the transaction, passing the data object, the callback, and request flag

```
int requestMode = 1; //0 = Return Gateway Response
                     //1 = Return Request + Gateway Response
                     //2 = Return Request Only

int r = executeTransaction(&data,WorldPayCallback,requestMode);
  if ( r != IDG_P2_STATUS_CODE_DO_SUCCESS ) {
            char strErr[200] = {0};
            memset(strErr, 0, 200);
            device_getIDGStatusCodeString(r, strErr);
            printf(">>FAIL<< \n   -- Execute WP Transaction Failed! ErrorCode:0x%02x, Info: %s -- \n", r, strErr);
}
else{
            printf("Execute Transaction Succeeded\n");

}
```

Await and parse the results in the WorldPayCallback.

Reference page:
https://atlassian.idtechproducts.com/confluence/display/KB/WorldPay+Integration+-+Home

To execute successful CONTACT test transactions, you can use any UAT USA EMV card with Contact interface, or the Fiserv Dual Interface EMV Test Cards (2 card set). To execute successful CONTACTLESS test transactions, you can use UAT USA EMV Test Cards with Contactless interface, or the Fiserv Dual Interface EMV Test Cards (2 card set). Other test cards may work if they share the same account number as the defined cards. Do not use live cards in a test environment.

For the VP3300/VP8300, you can enable Contactless transactions by setting data.enableCTLS=1 and data.msrOnly=0. To test Contactless transactions, you can use any UL Test Card that has a Contactless interface.

The device must be configured with proper terminal settings, AID's, and CAPKs. Please load the configuration file Worldpay_Device_Config.json to your device before running transactions. You can use SDK Demo, or UpdateIDT to perform the update. If you have a VP3300/VP8300, please load Worldpay_Device_Config_VP3300_CTLS.json to your device to allow both Contact and Contactless transactions.

The device must have the correct key. If a test device, it must have demo BDK 0123456789ABCDEFFEDCBA9876543210, and if production, it must have a secure data encryption key (ID Tech Key IDT-KEYINJ-D04)

This SDK is for driving ID Tech devices to capture payment information and get approval/decline from WorldPay, in addition to perform Voids and Returns. Other transaction types (Add Tip, etc) require the integrator to create their own request packets and send to WorldPay through their own methods. Please refer to Express API Interface Specification V3 for other request packets that can be created.

Communication:

The SDK attempts to contact WorldPay using standard curl http post functions. If you find your environment does not support curl, or have special communication requirement to access the internet, you can set the "requestMode" parameter of executeTransaction to a value of 2. This will instruct the SDK to collect all the card information and properly package/create the XML request packet, but instead of attempting to send to WorldPay, it will return this to the WorldPay callback, allowing your application to use it's alternate communication routines to contact WorldPay with.

For some setups, in order to securely connect to WorldPay servers, curl http requires you to include the file "ca-certificates.crt" in the same folder as the IDTech libraries. Please refer to the demo app compiled for your environment provided with the SDK for this file.

Setting the "requestMode" parameter to 1 will tell the SDK to process the transaction with WorldPay, but instead of just returning the WorldPay response packet, it will also return the WorldPay request packet that was sent. This can be used for debug purpose, or to evaluate/extract any of the collected card information as needed.

Sale vs QuickChip Transaction Types

`data.transactionType = WorldPay.TRANSACTION_TYPE.TRANSACTION_TYPE_SALE`
You use this transaction type if you know the final amount when the card is presented to the device. This will collect the card data and submit for approval at the amount you specify

`data.transactionType = WorldPay.TRANSACTION_TYPE.TRANSACTION_TYPE_QUICK_CHIP_START`
You use this transaction type if you do not know the final amount when the card is presented to the device. This will collect the card data and await the final transaction amount to be presented. Once the card data is collected, it does not need to be presented again after the final amount is known.

`data.transactionType = WorldPay.TRANSACTION_TYPE.TRANSACTION_TYPE_QUICK_CHIP_FINISH`
You use this transaction type once you know the final amount and the card data has already been previously collected with `TRANSACTION_TYPE_QUICK_CHIP_START`. If you attempt to execute this transaction type without the data previously collected, it will return an error. This will submit for approval at the final amount you specify without requiring the original card be presented again.

Going from Demo Mode to Production Mode

Once you are successful in performing test transactions, to perform live transactions, you will need to change the following four fields:

```
data strcpy(data.accountID," \0");  //needs production value
strcpy(data.accountToken," \0");   //needs production value
strcpy(data.acceptorID , "\0");   //needs production value
strcpy(data.license , "\0");   //needs production license
data.isTest = 0;    //must be 0 to perform live transactions
```

You will be provided production values for accountID, accountToken, acceptorID and license. In addition, you must change isTest to false so your transaction will be directed to the live production server at WorldPay.

Once you perform transaction, you can review them online by logging into your account at www.accessmyiq.com