

WorldNet Integration Using C++ SDK

Quick Start

Obtain IDTech C++ SDK Library version 1.0.35.037 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 WorldNet will require you establish the callback to receive the WorldNet transaction results, and then initiating the transaction by executing the specified API command.

While the final results will be returned back in the WorldNet 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 WorldNet 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 WorldNet 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 WorldNetCallback(char* message, int returnCode, int emvCode){  
  
    printf("WORLDNET RESPONSE>>>>>>\n");  
    printf("Return Code = %d\n", returnCode);  
    printf("EMV Code = %d\n", emvCode);  
    printf("Message = \n%s\n", message);  
}
```

Create a WorldNet data object and populate it with transaction details

```
WorldNetData data;
strcpy(data.terminal, "4480001\0");
strcpy(data.apiKey, "afae1c3fa41612acca86b435f65edcf3c930e3fe84b4dab4ef14321eb242890e09919c6e45e0a0506f523db0b5e6da2589c5182b1aca06b91558100619576e88\0");
strcpy(data.license, "2y2r0h6sorb-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;

strcpy(data.amount, "1.00\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
strcpy(data.transactionID, "3830\0"); //required if COMPLETION, VOID, RETURN. Otherwise, leave blank
strcpy(data.orderID, "112233445566\0"); //required value. Set a reference number
strcpy(data.additionalTags, "\0"); //set the additional tags
```

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_WorldNet(&data, WorldNetCallback, 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 WN Transaction Failed! ErrorCode:0x%02x, Info: %s -- \n", r, strErr);
}
else{
    printf("Execute Transaction Succeeded\n");
}
}
```

Await and parse the results in the WorldNetCallback.

Reference page:

<https://idtechproducts.atlassian.net/wiki/spaces/KB/pages/396001281/WorldNet+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.msrsOnly=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 `WorldNet-VP3300-Production-Config.json` to your device before running transactions. You can use SDK Demo, or UpdateIDT to perform the update.

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-D01)

This SDK is for driving ID Tech devices to capture payment information and get approval/decline from WorldNet, 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 WorldNet 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 WorldNet 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_WorldNet` 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 WorldNet, it will return this to the WorldNet callback, allowing your application to use it's alternate communication routines to contact WorldNet with.

Setting the "requestMode" parameter to 1 will tell the SDK to process the transaction with WorldNet, but instead of just returning the WorldNet response packet, it will also return the WorldNet 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 Transaction Type

```
data.transactionType = WorldNet.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.

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.terminal, "\0"); //needs production value
strcpy(data.apiKey, "\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 terminal, apiKey, and license. In addition, you must change isTest to 0 so your transaction will be directed to the live production server at WorldNet.