Worldpay Windows Integration
Quick Start

Using NuGet, Import IDTechSDK_STD 3.2.4.387 (or later) into the project.

Implement the MessageCallback

```
private void MessageCallBack(IDTechSDK.IDT_DEVICE_Types type, DeviceState state, byte[]
data, IDTTransactionData cardData, EMV_Callback emvCallback, RETURN_CODE
transactionResultCode, string ident)
```

Set the MessageCallback as the SDK callback

```
IDT_Device.setCallbackIP(MessageCallBack);
```

Plug in a USB device and monitor the Connect/Disconnect callback to understand device connection status

```
case DeviceState.Connected:
   output( "Connected " + IDTechSDK.Profile.IDT_DEVICE_String(type, connect,
ident),ident);
   break;
case DeviceState.Disconnected:
   output( "Disconnected " + IDTechSDK.Profile.IDT_DEVICE_String(type, connect,
ident),ident);
    break;
```

Please refer to the main SDK documentation for more advance topic like communicating with multiple devices and connecting through Serial or IP

Create a WorldPay data object and populate it with transaction details

```csharp
if (wp == null) wp = new WorldPay();

 WorldPay.WorldPayData data = new WorldPay.WorldPayData();

 data.accountID = "1188346";
 data.accountToken =
"E1EB3EFB049DFB599F1CB454E1CFC4FD14BF90BCE74456AE9E9490D7D609B466C81A3801";
 data.acceptorID = "364798674";
 data.license = " 2y2r0h6sorxb-5ufv0z2ao3jb"; //if live/production transaction, a
valid license must be provided
 data.transactionType = WorldPay.TRANSACTION_TYPE.TRANSACTION_TYPE_SALE;   //specify
Sale, Pre-Auth, or Completion
// data.transactionType =  WorldPay.TRANSACTION_TYPE.TRANSACTION_TYPE_PRE_AUTH;
//specify Sale, Pre-Auth, or Completion
// data.transactionType =  WorldPay.TRANSACTION_TYPE.TRANSACTION_TYPE_COMPLETION;
//specify Sale, Pre-Auth, or Completion
// data.transactionType =  WorldPay.TRANSACTION_TYPE.TRANSACTION_TYPE_VOID;
// data.transactionType =  WorldPay.TRANSACTION_TYPE.TRANSACTION_TYPE_RETURN;
// data.transactionType =  WorldPay.TRANSACTION_TYPE.TRANSACTION_TYPE_QUICK_CHIP_START;
         Start QuickChip -> Collect Card Data
// data.transactionType =  WorldPay.TRANSACTION_TYPE.TRANSACTION_TYPE_QUICK_CHIP_FINISH;
         Finish QuickChip -> Ready to submit with final amount known
 data.amount = "1.00";    //specify transaction Amount
 data.timeout =  30;  //Timeout for transaction
 data.isTest = true;  //if test transaction, set to TRUE.  If live transactions, set
to FALSE
 data.msrOnly = false; //for EMV + MSR, set to FALSE.  If MSR swipe only, set to TRUE
 data.transactionID = ""; //required if COMPLETION.  Otherwise, leave blank
 data.referenceNumber = ""; //required value.  If blank, SDK will put in random number
 data.terminalID = "12345"; //required value.  Set a terminal identifer
 data.ticketNumber = ""; //required value.  If blank, SDK will put in random number
 data.duplicateCheck = false;  //set to TRUE if you want to check for duplicate
transactions
data.duplicateOverride = true;  //set to TRUE if you want to allow duplicate charges
to same card
data.enableCTLS = true;  //set to TRUE if you want to allow Contactless transactions
(VP3300/VP8300 only)
data.deviceID = ""; //set to device identifier to target specific device, or leave
empty to target first/only device
data.declinePartial = true;  //set to TRUE if you want not allow partial approvial
from completing


 wp.returnRequest = true; //set to TRUE if you would like the request packet returned
with the response
```

Define the callback that will receive the transaction results. The cardData and receipt are
optional elements if you wanted to understand what initial card data was captured, or if you
want to build a receipt from the transaction data.

```csharp
void WPCallback(IDT_DEVICE_Types sender, string xmlResponse, RETURN_CODE resultCode,
EMV_RESULT_CODE emvCode, string ident, IDTTransactionData cardData, WorldPay.ReceiptData
receipt)
        {
```

```
        output("\r\n\r\nResponse received from Worldpay: " + resultCode.ToString() +
"\r\n\r\n", ident);
        output(xmlResponse, ident);
    }
```

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

```
bool requestOnly = false; // If this is TRUE, the SDK will return the request packet
instead of contacting WorldPay with it. For integrations that want to control
communication with WorldPay servers

string ident = data.deviceID; // If you want to target a specific device, you specify
either as a parameter you pass to executeTransaction, or through data.deviceID


RETURN_CODE rt = wp.executeTransaction(data, WorldPayCallback, requestOnly, ident);

if (rt == RETURN_CODE.RETURN_CODE_DO_SUCCESS)
{
    output("Start Transaction Successful\r\n", IDT_Device.currentIdent());
}
else
{
    output("Start transaction failed Error Code: " + "0x" + String.Format("{0:X}",
(ushort)rt) + ": " + IDTechSDK.errorCode.getErrorString(rt) + "\r\n",
IDT_Device.currentIdent());
}
```

Await and parse the results in the WorldPayCallback.

```
void WPCallback(IDT_DEVICE_Types sender, string xmlResponse, RETURN_CODE resultCode,
EMV_RESULT_CODE emvCode, string ident, IDTTransactionData cardData, WorldPay.ReceiptData
receipt)
    {
        output("\r\n\r\nResponse received from Worldpay: " + resultCode.ToString() +
"\r\n\r\n", ident);
        output(xmlResponse, ident);
    }
```

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.

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
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.

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.accountID = "";  //needs production value
data.accountToken = "";   //needs production value
data.acceptorID = "";  //needs production value
data.isTest = false;  //must be false to perform live transactions
data.license = "";  //needs production license
```

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