First Data.

Payeezy[SM] eCommerce Solution

Connect
Integration Guide

Version 2016-3 (IPG)

Contents

Getting Support

There are different manuals available for First Data's eCommerce solutions. This Integration Guide will be the most helpful for integrating the Connect solution for usage with our distribution channels in Europe, Asia, Australia, Latin America and Africa.

For information about settings, customisation, reports and how to process transactions manually (by keying in the information) please refer to the User Guide Virtual Terminal.

If you have read the documentation and cannot find the answer to your question, please contact at pghelpdesk@icicims.com and 1800 102 1673

1 Introduction

The Connect solution provides a quick and easy way to add payment capabilities to your website.

Connect manages the customer redirections that are required in the checkout process of many payment methods or authentication mechanisms and gives you the option to use secure hosted payment pages which can reduce the burden of compliance with the Data Security Standard of the Payment Card Industry (PCI DSS)

This document describes how to integrate your website using Connect and provides step by step instructions on how to quickly start accepting payments from your webshop.

Depending on your business processes, it can also make sense to additionally integrate our Web Service API solution (see Web Service API Integration Guide).

2 Payment process options

The Connect solution provides a number of different options for the payment process to support integrations where you handle most of the customer interactions on your own website up to integrations where you use ready-made form pages for the entire payment process.

In the scenarios where you prefer not to use a hosted form, you can submit the required customer data directly from your own form to First Data but please be aware that if you store or process sensitive cardholder data within your own application, you must ensure that your system components are compliant with the Data Security Standard of the Payment Card Industry (PCI DSS).

2.1 Checkout option ‘classic’

The checkout option ‘classic’ splits the payment process into multiple pages where you can easily decide, what kind of information you want to get collected by one of the gateway’s hosted forms or what you want to collect yourself within your webshop environment.

You can e.g. let customers select their preferred payment method within your webshop and submit that payment method in your request to Connect – or if you should prefer not to send the payment method, the Connect solution will automatically show a payment method selection page to your customer where they can choose from all payment methods that are activated for your store.

With three different modes, you can define the range of data that shall be captured by the payment gateway:

- PayOnly: shows a hosted page to collect the minimum set of information for the transaction (e. g. cardholder name, card number, expiry date and card code for a credit card transaction)
- PayPlus: in addition to the above, the payment gateway collects a full set of billing information on an additional page
- FullPay: in addition to the above, the payment gateway displays a third page to also collect shipping information

The most important aspect around the usage of hosted payment pages is the security of sensitive cardholder data. When you decide to let your customers enter their credit card details on the page that we provide and host on our servers for this purpose, it facilitates your compliance with the Data Security Standard of the Payment Card Industry (PCI DSS) as the payment processing is completely hosted by First Data.

The hosted pages can be customised with your own logo, colours, and font types in order to make them fit to the look and feel of your webshop. Please refer to the User Guide Virtual Terminal to learn about how to make such customisations.

2.2 Checkout option ‘combinedpage’

For the case where you decide to let your customers select the payment method on a hosted page, the checkout option ‘combinedpage’ consolidates the payment method choice and the typical next step (e.g. entry of card

details or selection of bank) in a single page which gets automatically optimized for different kinds of user devices, e.g. PC, smartphone, tablet, etc.

This hosted page also shows your merchant name at the top and allows you to display a summary of the purchased items to your customer.

Please note that this checkout option has some functional limitations in comparison to the 'classic' option:

- Supported payment methods are currently limited to: credit cards, Maestro, and MasterPass
- There are no customisation options (logo, colours, etc.)
- It makes use of technical mechanisms that may not work with out-dated browser versions

3 Getting Started

This section provides a simple example on how to integrate your website using the "classic" checkout option in PayOnly Mode. Examples are provided using ASP and PHP. This section assumes that the developer has a basic understanding of his chosen scripting language.

3.1 Checklist

In order to integrate with the payment gateway, you must have the following items:

- Store Name

  This is the ID of the store that was given to you by First Data.
  For example : 10123456789

- Shared Secret

  This is the shared secret provided to you by First Data.
  This is used when constructing the hash value (see below).

3.2 ASP Example

The following ASP example demonstrates a simple page that will communicate with the payment gateway in PayOnly mode.

When the cardholder clicks *Submit*, they are redirected to the First Data secure page to enter the card details. After payment has been completed, the user will be redirected to the merchants receipt page. The location of the receipt page can be configured.

```
<!-- #include file="ipg-util.asp"-->

<html>
  <head><title>IPG Connect Sample for ASP</title></head>
  <body>
  <p><h1>Order Form</h1></p>

  <form method="post" action=" https://test.ipg-
online.com/connect/gateway/processing ">
        <input type="hidden" name="txntype" value="sale">
        <input type="hidden" name="timezone" value="Europe/Berlin"/>
        <input type="hidden" name="txndatetime" value="<% getDateTime()
        %>"/>
        <input type=”hidden” name=”hash_algorithm” value=”SHA256”/>
        <input type="hidden" name="hash" value="<% call createHash(
        "13.00","356" ) %>"/>
        <input type="hidden" name="storename" value="10123456789" />
```

```
        <input type="hidden" name="mode" value="payonly"/>
        <input type="hidden" name="paymentMethod" value="M"/>
        <input type="text" name="chargetotal" value="13.00" />
        <input type="hidden" name="currency" value="356"/>
        <input type="submit" value="Submit">
  </form>
  </body>
</html>
```

The code presented in Appendix I represents the included file ipg-util.asp. It includes code for generating a SHA-256 hash as is required by First Data. The provision of a hash in the example ensures that this merchant is the only merchant that can send in transactions for this store.

Note, the POST URL used is for integration testing only. When you are ready to go into production, please contact First Data and you will be provided with the live production URL.

Note, the included file, ipg-util.asp uses a server side JavaScript file to build the SHA-256 hash. This file can be provided on request. To prevent fraudulent transactions, it is recommended that the 'hash' is calculated within your server and JavaScript is not used like shown in the samples mentioned.

3.3 PHP Example

The following PHP example demonstrates a simple page that will communicate with the payment gateway in PayOnly mode.

When the cardholder clicks *Submit*, they are redirected to the First Data secure page to enter the card details. After payment has been completed, the user will be redirected to the merchants receipt page. The location of the receipt page can be configured.

```
<? include("ipg-util.php"); ?>

<html>
<head><title>IPG Connect Sample for PHP</title></head>
  <body>
  <p><h1>Order Form</h1>

<form method="post" action="https://test.ipg-
online.com/connect/gateway/processing">
  <input type="hidden" name="txntype" value="sale">
<input type="hidden" name="timezone" value="Europe/Berlin"/> <input
type="hidden" name="txndatetime" value="<?php echo getDateTime() ?>"/>
  <input type=”hidden” name=”hash_algorithm” value=”SHA256”/>

<input type="hidden" name="hash" value="<?php echo createHash(
"13.00","356" ) ?>"/>
  <input type="hidden" name="storename" value="10123456789"/>
<input type="hidden" name="mode" value="payonly"/>
<input type="hidden" name="paymentMethod" value="M"/>
<input type="text" name="chargetotal" value="13.00"/>
<input type="hidden" name="currency" value="356"/>

  <input type="submit" value="Submit">
  </form>
  </body>
</html>
```

Note that the POST URL used in this example is for integration testing only. When you are ready to go into production, please contact First Data and you will be provided with the live production URL.

The code presented in Appendix II represents the included file ipg-util.php. It includes code for generating a SHA-256 hash as is required by First Data. The provision of a hash in the example ensures that this merchant is the only merchant that can send in transactions for this store.

3.4 Amounts for test transactions

When using our test system for integration, odd amounts (e. g. 13.01 EUR or 13.99 EUR) can cause the transaction to decline as these amounts are sometimes used to simulate unsuccessful authorisations.

We therefore recommend using even amounts for testing purpose, e. g. 13.00 EUR like in the example above.

4 Mandatory Fields

Depending on the transaction type, the following form fields must be present in the form being submitted to the payment gateway (X = mandatory field). Please refer to this Integration Guide's Appendixes for implementation details in relation to alternative payment methods.

Field name	**Description, possible values and format**	**„Sale" transaction**	**PreAuth***	**PostAuth***	**Void**
txntype	'sale', 'preauth', 'postauth' or 'void' (the transaction type – please note the descriptions of transaction types in the User Guide Virtual Terminal & Manager) The possibility to send a 'void' using the Connect interface is restricted. Please contact your local support team if you want to enable this feature.	X (sale)	X (preauth)	X (postauth)	X (void)
timezone	Timezone of the transaction in Area/Location format, e.g. Africa/Johannesburg America/New_York America/Sao_Paulo Asia/Calcutta Australia/Sydney Europe/Amsterdam Europe/Berlin Europe/Dublin Europe/London Europe/Rome	X	X	X	X
txndatetime	YYYY:MM:DD-hh:mm:ss (exact time of the transaction)	X	X	X	X
hash_algorithm	This is to indicate the algorithm that you use for hash calculation. The only possible value at this point is SHA256.	X	X	X	X
hash	This is a SHA hash of the following fields: storename + txndatetime + chargetotal + currency + sharedsecret. Note, that it is important to have the hash generated in this exact order. An example of how to generate a SHA-256 hash is given in Appendix I.	X	X	X	X
storename	This is the ID of the store provided by First Data.	X	X	X	X
mode	'fullpay', 'payonly' or 'payplus' (the chosen mode for the transaction when using the 'classic' checkout option)	X	X		
chargetotal	This is the total amount of the transaction using a dot or comma as decimal separator, e. g. 12.34 for an amount of 12 Euro and 34 Cent. Group separators like1,000.01 /	X	X	X	X

	1.000,01 are not allowed.				
currency	The numeric ISO code of the transaction currency, e. g. 356 for Euro (see examples below)	X	X	X	
oid	The order ID of the initial action a PostAuth or Void shall be initiated for			X	X
tdate	Exact identification of a transaction that shall be voided. You receive this value as result parameter ‚tdate' of the corresponding transaction.				X

* The transaction types ‘preauth’ and ‘postauth’ only apply to the payment methods credit card.

Currency code list:

Currency name	Currency code	Currency number
Australian Dollar	AUD	036
Brazilian Real	BRL	986
Euro	EUR	978
Indian Rupee	INR	356
Pound Sterling	GBP	826
US Dollar	USD	840
South African Rand	ZAR	710
Swiss Franc	CHF	756
Bahrain Dinar	BHD	048
Canadian Dollar	CAD	124
Chinese Renmibi	CNY	156
Croatian Kuna	HRK	191
Czech Koruna	CZK	203
Danish Krone	DKK	208
Hong Kong Dollar	HKD	344
Hungarian Forint	HUF	348
Israeli New Shekel	ISL	376
Japanese Yen	JPY	392
Kuwaiti Dinar	KWD	414
Lithuanian Litas	LTL	440
Mexican Peso	MXN	484
New Zealand Dollar	NZD	554
Norwegian Krone	NOK	578
Omani Rial	OMR	512
Polish Zloty	PLN	985
Romanian New Leu	RON	946
Saudi Rihal	SAR	682
Singapore Dollar	SGD	702
South Korean Won	KRW	410
Swedish Krona	SEK	752
Turkish Lira	TRY	949
UAE Dirham	AED	784

5 Optional Form Fields

Field name	Description, possible values and format
cardFunction	This field allows you to indicate the card function in case of combo cards which provide credit and debit functionality on the same card. It can be set to ‘credit’ or ‘debit’. The field can also be used to validate the card type in a way that transactions where the submitted card function does not match the card’s capabilities will be declined. If you e.g. submit “cardFunction=debit” and the card is a credit card, the transaction will be declined.
checkoutoption	This field allows you to set the checkout option to ‘classic’ for a payment process that is split into multiple pages or to ‘combinedpage’ for a payment process where the payment method choice and the typical next step (e.g. entry of card details or

<table>
<tr><td></td><td>selection of bank) in consolidated in a single page.</td></tr>
<tr><td>comments</td><td>Place any comments here about the transaction.</td></tr>
<tr><td>customerid</td><td>This field allows you to transmit any value, e. g. your ID for the customer.</td></tr>
<tr><td>dccInquiryId</td><td>Inquiry ID for a Dynamic Pricing request. Used to send the Inquiry ID you have obtained via a Web Service API call (RequestMerchantRateForDynamicPricing). This value will be used to retrieve the currency conversion information (exchange rate, converted amount) for this transaction.</td></tr>
<tr><td>dynamicMerchantName</td><td>The name of the merchant to be displayed on the cardholder’s statement.The length of this field should not exceed 25 characters. If you want to use this field, please contact your local support team to verify if this feature is supported in your country.</td></tr>
<tr><td>invoicenumber</td><td>This field allows you to transmit any value, e. g. an invoice number or class of goods. Please note that the maximum length for this parameter is 48 characters.</td></tr>
<tr><td>hashExtended</td><td>The extended hash is an optional security feature that allows you to include all parameters of the transaction request. It needs to be calculated using all request parameters in ascending order of the parameter names.</td></tr>
<tr><td>item1 up to item999</td><td>The ‘item1’ to ‘item999’ parameters allow you to send basket information in the following format:

id;description;quantity;item_total_price;sub_total;vat_tax;shipping

'shipping' always has to be set to '0' for single line items. If you want to include a shipping fee for an order, please use the predefined id IPG_SHIPPING.

For other fees that you may want to add to the total order, you can use the predefined id IPG_HANDLING.

When you want to apply a discount, you should include an item with a negative amount and change accordingly the total amount of the order. Do not forget to regard the ‘quantity’ when calculating the values e.g.: subtotal and VAT since they are fixed by items. Examples:

A;Product A;1;5;3;2;0
B;Product B;5;10;7;3;0
C;Product C;2;12;10;2;0
D;Product D;1;-1.0;-0.9;-0.1;0
IPG_SHIPPING;Shipping costs;1;6;5;1;0
IPG_HANDLING;Transaction fee;1;6.0;6.0;0;0</td></tr>
<tr><td>language</td><td>This parameter can be used to override the default payment page language configured for your merchant store.
The following values are currently possible:
<table>
<tr><th>Language</th><th>Value</th></tr>
<tr><td>Chinese (simplified)</td><td>zh_CN</td></tr>
<tr><td>Chinese (traditional)</td><td>zh_TW</td></tr>
<tr><td>Dutch</td><td>nl_NL</td></tr>
<tr><td>English (USA)</td><td>en_US</td></tr>
<tr><td>English (UK)</td><td>en_GB</td></tr>
<tr><td>Finnish</td><td>fi_FI</td></tr>
<tr><td>French</td><td>fr_FR</td></tr>
<tr><td>German</td><td>de_DE</td></tr>
<tr><td>Italian</td><td>it_IT</td></tr>
<tr><td>Portuguese (Brazil)</td><td>pt_BR</td></tr>
<tr><td>Slovak</td><td>sk_SK</td></tr>
<tr><td>Spanish</td><td>es_ES</td></tr>
</table></td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
<tr><td>merchantTransactionId</td><td>Allows you to assign a unique ID for the transaction. This ID can be used to reference to this transactions in a PostAuth or Void request (referencedMerchantTransactionId).</td></tr>
<tr><td>mobileMode</td><td>If your customer uses a mobile device for shopping at your online store you can submit this parameter with the value ‘true’. This will lead your customer to a payment page flow that has been specifically designed for mobile devices.</td></tr>
<tr><td>numberOfInstallments</td><td>This parameter allows you to set the number of instalments for a Sale transaction if</td></tr>
</table>

<table>
<tr><td></td><td>your customer pays the amount in several parts.</td></tr>
<tr><td>oid</td><td>This field allows you to assign a unique ID for your order. If you choose not to assign an order ID, the First Data system will automatically generate one for you.</td></tr>
<tr><td>paymentMethod</td><td>If you let the customer select the payment method (e. g. MasterCard, Visa) in your shop environment or want to define the payment type yourself, transmit the parameter ‘paymentMethod’ along with your Sale or PreAuth transaction.
If you do not submit this parameter, the payment gateway will display a drop-down menu to the customer to choose from the payment methods available for your shop.
Valid values are:
<table>
<tr><th>Payment Method</th><th>Value</th></tr>
<tr><td>MasterCard</td><td>M</td></tr>
<tr><td>Visa (Credit/Debit)</td><td>V</td></tr>
<tr><td>American Express*</td><td>A</td></tr>
<tr><td>Local Wallets India*</td><td>indiawallet</td></tr>
<tr><td>Maestro</td><td>MA</td></tr>
<tr><td>MasterPass*</td><td>masterpass</td></tr>
<tr><td>Netbanking (India)*</td><td>netbanking</td></tr>
<tr><td>RuPay</td><td>RU</td></tr>
</table></td></tr>
<tr><td>ponumber</td><td>This field allows you to submit a Purchase Order Number with up to 50 characters.</td></tr>
<tr><td>refer</td><td>This field describes who referred the customer to your store.</td></tr>
<tr><td>referencedMerchantTransactionID</td><td>This field allows to reference to a merchantTransactionId of a transaction when performing a Void. This can be used as an alternative to tdate if you assigned a merchantTransactionId in the original transaction request.</td></tr>
<tr><td>responseFailURL</td><td>The URL where you wish to direct customers after a declined or unsuccessful transaction (your Sorry URL) – only needed if not setup in Virtual Terminal / Customisation.</td></tr>
<tr><td>responseSuccessURL</td><td>The URL where you wish to direct customers after a successful transaction (your Thank You URL) – only needed if not setup in Virtual Terminal / Customisation.</td></tr>
<tr><td>reviewOrder</td><td>MasterPass-specific parameter for scenarios where the final amount needs to be confirmed by the customer after returning from the Wallet. Set the value for this parameter to ‘true’ in order to indicate that the final transaction amount needs to be reviewed by the cardholder.</td></tr>
<tr><td>reviewURL</td><td>MasterPass-specific parameter for scenarios where the final amount needs to be confirmed by the customer after returning from the MasterPass environment. Use this parameter to indicate where the customer shall be redirected to in order to review and complete the transaction after having clicked on “Finish shopping” within the Wallet.</td></tr>
<tr><td>shipping</td><td>This parameter can be used to submit the shipping fee, in the same format as ‘chargetotal’. If you submit ‘shipping’, the parameters ‘subtotal’ and ‘vattax’ have to be submitted as well. Note that the ‘chargetotal’ has to be equal to ‘subtotal’ plus ‘shipping’ plus ‘vattax’.</td></tr>
<tr><td>trxOrigin</td><td>This parameter allows you to use the secure and hosted payment form capabilities within your own application for Mail/Telephone Order (MOTO) payments.
Possible values are ‘MOTO‘ (for transactions where you have received the order over the phone or by mail and enter the payment details yourself) and ‘ECI‘ (for standard usage in an eCommerce environment where your customer enters the payment details).</td></tr>
<tr><td>vattax</td><td>This field allows you to submit an amount for Value Added Tax or other taxes, e.g. GST in Australia. Please ensure the sub total amount plus shipping plus tax equals the charge total.</td></tr>
</table>

* launching soon

6 Using your own forms to capture the data

If you decide to create your own forms, i. e. not to use the ones provided and hosted by First Data, there are additional mandatory fields that you need to include. These fields are listed in the following sections, depending on the mode you choose.

In addition, you should check if JavaScript is activated in your customer’s browser and if necessary, inform your customer that JavaScript needs to be activated for the payment process.

6.1 PayOnly Mode

After your customer has decided how to pay, you present a corresponding HTML-page with a form to enter the payment data as well as hidden parameters with additional transaction information.

In addition to the mandatory fields listed above, your form needs to contain the following fields (part of them can be hidden):

Field name	**Description, possible values and format**	**Credit Card (+ Visa Debit)**		**Maestro**	**Netbanking, MasterPass**
cardnumber	Your customer's card number. 12-24 digits.	X		X	
expmonth	The expiry month of the card (2 digits)	X		X	
expyear	The expiry year of the card (4 digits)	X		X	
cvm	The card code, in most cases on the backside of the card (3 to 4 digits)	X		X as an optional field "if on card"	

6.2 PayPlus Mode

Using PayPlus mode, it is possible to additionally transfer billing information to the payment gateway. The following table describes the format of these additional fields:

Field Name	Possible Values	Description
bcompany	Alphanumeric characters, spaces, and dashes	Customers Company
bname	Alphanumeric characters, spaces, and dashes	Customers Name
baddr1	Limit of 30 characters, including spaces	Customers Billing Address 1
baddr2	Limit of 30 characters, including spaces	Customers Billing Address 2
bcity	Limit of 30 characters, including spaces	Billing City
bstate	Limit of 30 characters, including spaces	State, Province or Territory
bcountry	2 Letter Country Code	Country of Billing Address
bzip	International Postal Code	Zip or Postal Code
phone	Limit of 20 Characters	Customers Phone Number
fax	Limit of 20 Characters	Customers Fax Number
email	Limit of 64 Characters	Customers Email Address

6.3 FullPay Mode

Using FullPay mode, it is possible to additionally transfer shipping information to the payment gateway. The billing information is as specified above. The following table describes the format of the shipping fields:

Field Name	Possible Values	Description
sname	Alphanumeric characters, spaces, and dashes	Ship-to Name
saddr1	Limit of 30 characters, including spaces	Shipping Address Line 1
saddr2	Limit of 30 characters, including spaces	Shipping Address Line 2
scity	Limit of 30 characters, including spaces	Shipping City
sstate	Limit of 30 characters, including	State, Province or Territory

	spaces	
scountry	2 letter country code	Country of Shipping Address
szip	International Postal Code	Zip or Postal Code

6.4 Validity checks

Prior to the authorisation request for a transaction, the payment gateway performs the following validation checks:

- The expiry date of cards needs to be in the future
- The Card Security Code field must contain 3 or 4 digits
- The structure of a card number must be correct (LUHN check)

If the submitted data should not be valid, the payment gateway presents a corresponding data entry page to the customer.

To avoid this hosted page when using your own input forms for the payment process, you can transmit the following additional parameter along with the transaction data:

```
full_bypass=true
```

In that case you get the result of the validity check back in the transaction response and can display your own error page based on this.

Please note, if the transaction is eligible for DCC (your store is configured for DCC and the customer is paying by credit card capable of DCC), your customer will be presented the DCC page despite having full_bypass set to true. This is due to regulatory reasons. You can avoid displaying of DCC choice pages by doing the DCC Inquiry yourself via our Web Service API (RequestMerchantRateForDynamicPricing).

7 Additional Custom Fields

You may send as many custom fields to the payment gateway as you wish. Custom field values are returned along with all other fields to the response URL.

It is also possible to document up to fifteen custom fields in your store configuration. You may use these fields to gather additional customer data geared toward your business specialty, or you may use them to gather additional customer demographic data which you can then store in your own database for future analysis.

8 3D Secure

The Connect solution includes the ability to authenticate transactions using Verified by Visa, MasterCard SecureCode and American Express SafeKey. If your credit card agreement includes 3D Secure and your Merchant ID has been activated to use this service, you do not need to modify your payment page.

If you are enabled to submit 3D Secure transactions but for any reason want to submit specific transactions without using the 3D Secure protocol, you can use the additional parameter authenticateTransaction and set it to either “true” or “false”.

Example for a transaction without 3D Secure:

```
<input type="hidden" name="authenticateTransaction" value="false"/>
```

In principle, it may occur that 3D Secure authentications cannot be processed successfully for technical reasons. If one of the systems involved in the authentication process is temporarily not responding, the payment transaction will be processed as a “regular” eCommerce transaction (ECI 7). **A liability shift to the card issuer**

for possible chargebacks is not warranted in this case. If you prefer that such transactions shall not be processed at all, our technical support team can block them for your Store on request.

Credit card transactions with 3D Secure hold in a pending status while cardholders search for their password or need to activate their card for 3D Secure during their shopping experience. During this time when the final transaction result of the transaction is not yet determined, the payment gateway sets the Approval Code to „?:waiting 3dsecure“. If the session expires before the cardholder returns from the 3D Secure dialogue with his bank, the transaction will be shown as “N:-5103:Cardholder did not return from ACS”.

Please note that the technical process of 3D Secure transactions differs in some points compared to a normal transaction flow. If you already have an existing shop integration and plan to activate 3D Secure subsequently, we recommend performing some test transactions on our test environment.

9 Data Vault

With the Data Vault product option you can store sensitive cardholder data in an encrypted database in First Data’s data centre to use it for subsequent transactions without the need to store this data within your own systems.

If you have ordered this product, the Connect solution offers you the following functions:

- **Store or update payment information when performing a transaction**
  Additionally send the parameter ‘hosteddataid’ together with the transaction data as a unique identification for the payment information in this transaction. Depending on the payment type, credit card number and expiry date or account number and bank code will be stored under this ID if the transaction has been successful. In cases where the submitted ‘hosteddataid’ already exists for your store, the stored payment information will be updated.

- **Initiate payment transactions using stored data**
  If you stored cardholder information using the Data Vault option, you can perform transactions using the ‘hosteddataid’ without the need to pass the credit card or bank account data again.
  Please note that it is not allowed to store the card code (in most cases on the back of the card) so that for credit card transactions, the cardholder still needs to enter this value. If you use First Data’s hosted payment forms, the cardholder will see the last four digits of the stored credit card number, the expiry date and a field to enter the card code.

  When using multiple Store IDs, it is possible to access stored card data records of a different Store ID then the one that has been used when storing the record. In that way you can for example use a shared data pool for different distributive channels. To use this feature, submit the Store ID that has been used when storing the record as the additional parameter ‘hosteddatastoreid’.

- **Avoid duplicate cardholder data for multiple records**
  To avoid customers using the same cardholder data for multiple user accounts, the additional parameter ‘declineHostedDataDuplicates’ can be sent along with the request. The valid values for this parameter are ‘true’/’false’. If the value for this parameter is set to ‘true’ and the cardholder data in the request is already found to be associated with another ‘hosteddataid’, the transaction will be declined.

See further possibilities with the Data Vault product in the Integration Guide for the Web Service API.

10 Recurring Payments

For credit card and PayPal transactions, it is possible to install recurring payments using Connect. To use this feature, the following additional parameters will have to be submitted in the request:

Field Name	Possible Values	Description
recurringInstallmentCount	Number between 1 and	Number of installments to be made including the

	999	initial transaction submitted
recurringInstallmentPeriod	day week month year	The periodicity of the recurring payment
recurringInstallmentFrequency	Number between 1 and 99	The time period between installments
recurringComments	Limit of 100 characters, including spaces	Any comments about the recurring transaction

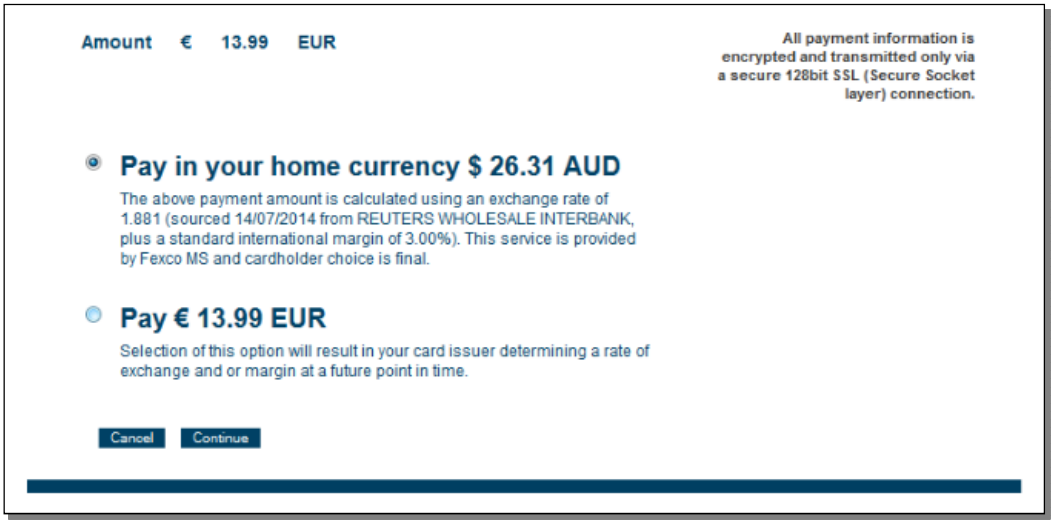
Note that the start date of the recurring payments will be the current date and will be automatically calculated by the system.

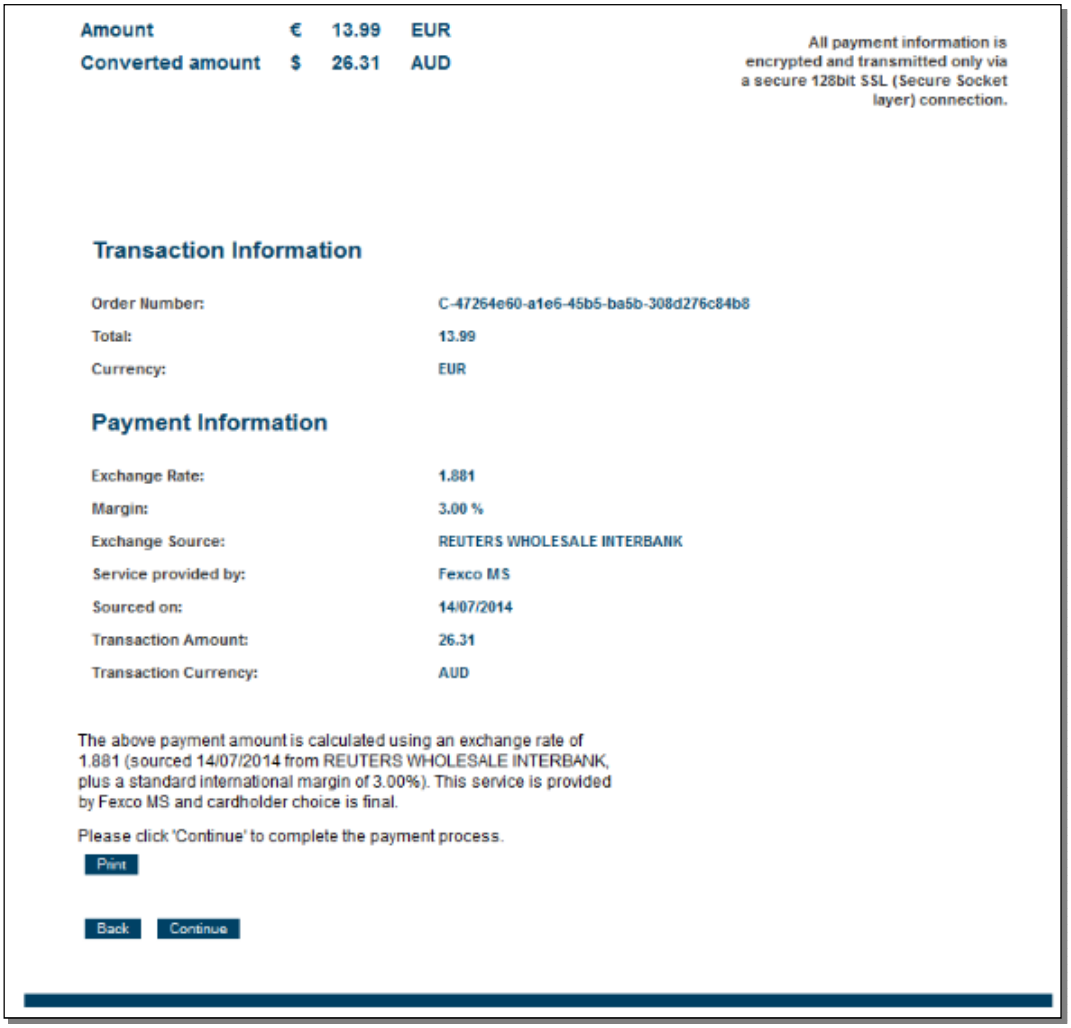
The recurring payments installed using Connect can be modified or cancelled using the Virtual Terminal or Web Service API.

11 Global Choice™ and Dynamic Pricing

With First Data's Global Choice™, foreign customers have the choice to pay for goods and services purchased online in their home currency when using their Visa or MasterCard credit card for the payment. The currency conversion is quick and eliminates the need for customers to mentally calculate the estimated cost of the purchase in their home currency. International Visa and MasterCard eCommerce customers can make informed decisions about their online purchases and eradicate any unexpected pricing or foreign exchange conversions on receipt of their monthly statements.

If your Store has been activated for this product option, the Connect solution automatically offers a currency choice to your customers if the card they use has been issued in a country with a currency that is different to your default currency.





Please note that for compliance reasons First Data's Global Choice can only be offered on transactions that take place in full at that time (e.g. Sale, Refund) and not on any delayed settlement (e.g. pre/post auth, recurring) due to the fluctuation of the rate of exchange.

Another option for your foreign customers is to display all pricing within your online store in their home currency using our Dynamic Pricing solution. This solution removes the need for your company to set pricing in any other currency other than your home currency.
Please see the Integration Guide for our Web Service API for details on how to request the exchange rates.

If your Store has been activated for this product option and you want to submit the payment transaction via our Connect solution, you need to send the DCC Inquiry ID that you have received along with the exchange rate request in the parameter 'dccInquiryId'.

12 Transaction Response

Upon completion, the transaction details will be sent back to the defined 'responseSuccessURL' or 'responseFailURL' as hidden fields:

Field name	Description
approval_code	Approval code for the transaction. The first character of this parameter is the most helpful indicator for verification of the transaction result. 'Y' indicates that the transaction has been successful 'N' indicates that the transaction has not been successful

	"?" indicates that the transaction has been successfully initialised, but a final result is not yet available since the transaction is now in a waiting status. The transaction status will be updated at a later stage.
oid	Order ID
refnumber	Reference number
status	Transaction status, e.g. 'APPROVED', 'DECLINED' (by authorisation endpoint or due to fraud prevention settings) or 'FAILED' (wrong transaction message content/parameters, etc.).
txndate_processed	Time of transaction processing
tdate	Identification for the specific transaction, e. g. to be used for a Void
fail_reason	Reason the transaction failed
response_hash	Hash-Value to protect the communication (see note below)
processor_response_code	The response code provided by the backend system. Please note that response codes can be different depending on the used payment type and backend system. While for credit card payments, the response code '00' is the most common response for an approval, the backend for giropay transactions for example returns the response code '4000' for succesful transactions.
fail_rc	Internal processing code for failed transactions
terminal_id	Terminal ID used for transaction processing
ccbin	6 digit identifier of the card issuing bank
cccountry	3 letter alphanumeric ISO code of the cardholder's country (e.g. USA, DEU, ITA, etc.) Filled with "N/A" if the cardholder's country cannot be determined or the payment type is not credit card
ccbrand	Brand of the credit or debit card: MC VISA AMEX MAESTRO RUPAY Filled with "N/A" for any payment method which is not a credit card or debit card

For 3D Secure transactions only:

response_code_3dsecure	Return code indicating the classification of the transaction: **1** – Successful authentication (VISA ECI 05, MasterCard ECI 02) **2** – Successful authentication without AVV (VISA ECI 05, MasterCard ECI 02) **3** – Authentication failed / incorrect password (transaction declined) **4** – Authentication attempt (VISA ECI 06, MasterCard ECI 01) **5** – Unable to authenticate / Directory Server not responding (VISA ECI 07) **6** – Unable to authenticate / Access Control Server not responding (VISA ECI 07) **7** – Cardholder not enrolled for 3D Secure (VISA ECI 06) **8** – Invalid 3D Secure values received, most likely by the credit card issuing bank's Access Control Server (ACS) Please see note about blocking ECI 7 transactions in the 3D Secure section of this document.

For Global Choice™ transactions only:

dcc_foreign_amount	Converted amount in cardholder home currency. Decimal number with dot (.) as a decimal separator.
dcc_foreign_currency	ISO numeric code of the cardholder home currency. This transaction is performed in this currency. String.
dcc_margin_rate_percentage	Percent of margin applied to the original amount. Decimal number with dot (.) as a decimal separator.
dcc_rate_source	Name of the exchange rate source (e.g. Reuters Wholesale Inter Bank). String.
dcc_rate	Exchange rate. Decimal number with dot (.) as a decimal separator.
dcc_rate_source_timestamp	Exchange rate origin time. Integer - Unix timestamp (seconds since 1.1.1970).

For MasterPass transactions only:

redirectURL	When reviewOrder has been set to 'true', the response contains the URL that you need to finalize the transaction

Additionally when using your own error page for negative validity checks (full_bypass=true):

fail_reason_details	Comma separated list of missing or invalid variables. Note that 'fail_reason_details' will not be supported in case of PayPlus and FullPay mode.
invalid_cardholder_data	**true** – if validation of card holder data was negative **false** – if validation of card holder data was positive but transaction has been declined due to other reasons

In addition, your custom fields and billing/shipping fields will also be sent back to the specific URL.

Please consider when integrating that new response parameters may be added from time to time in relation to product enhancements or new functionality.

The parameter 'response_hash' allows you to recheck if the received transaction response has really been sent by First Data and can therefore protect you from fraudulent manipulations. The value is created with a SHA Hash using the following parameter string:

```
sharedsecret + approval_code  + chargetotal + currency + txndatetime
+ storename
```

The hash algorithm is the same as the one that you have set in the transaction request.

Please note that if you want to use this feature, you have to store the 'txndatetime' that you have submitted with the transaction request in order to be able to validate the response hash.

In addition, it is possible that the payment gateway sends the above result parameters to a defined URL. To use this notification method, you can specify an URL in the Customisation section of the Virtual Terminal or submit the URL in the following additional transaction parameter 'transactionNotificationURL'.

Please note that:

- No SSL handshake, verification of SSL certificates will be done in this process.
- The Notification URL needs to listen either on port 80 (http) or port 443 (https) – other ports are not supported.
- The response hash parameter for validation (using the same algorithm that you have set in the transaction request) 'notification_hash' is calculated as follows:

  ```
  chargetotal + sharedsecret + currency + txndatetime + storename
  + approval_code.
  ```

Appendix I – How to generate a SHA-256 Hash

Example

- storename = 98765432101
- txndatetime = 2013:07:16-09:57:08
- chargetotal = 1.00
- currency = 826
- sharedsecret = TopSecret

Step 1. Collect selected parameters: storename, txndatetime, chargetotal, currency and sharedsecret and join the parameters' values to one string (use only parameters' values and not the parameters' names).

98765432101 2013:07:16-09:57:08 1.00 826 TopSecret

Step 2. Convert the created string to its ascii hexadecimal representation.

39383736353433323130313231333a30373a31362d30393a35373a3038312e3030383236546f70536563726574

Step 3. Pass the ascii hexadecimal representation of the created string to the SHA-256 algorithm.

SHA256(39383736353433323130313231333a30373a31362d30393a35373a3038312e3030383236546f70536563726574)

Step 4. Use the value returned by the SHA-256 algorithm and submit it to our payment gateway in the given form.

3d7e75aa0b4e0e1d4a7ac87e451e64692cced46f4358ef35a69d96721341243c

```
<input type="hidden" name="hash"
value="3d7e75aa0b4e0e1d4a7ac87e451e64692cced46f4358ef35a69d96721341243c
"/>
```

Appendix II – ipg-util.asp

```
<Script LANGUAGE=JScript RUNAT=Server src="sha256.js">
</SCRIPT>
<Script LANGUAGE=JScript RUNAT=Server>
      var today = new Date();
      var formattedDate = today.formatDate("Y:m:d-H:i:s");

      /*
            Function that calculates the hash of the following
            parameters:
             - Store Id
             - Date/Time(see $dateTime above)
             - chargetotal
             - shared secret
             - currency (numeric ISO value)
      */
      function createHash(chargetotal, currency) {
            // Please change the store Id to your individual Store ID
            var storename = "10123456789;
            // NOTE: Please DO NOT hardcode the secret in that
script. For example read it from a database.
            var sharedSecret = "sharedsecret";

            var stringToHash = storename + formattedDate +
chargetotal + currency + sharedSecret;

            var ascii = getHexFromChars(stringToHash);

            var hash = calcSHA256(ascii);

            Response.Write(hash);
      }
      function getHexFromChars(value) {
            var char_str = value;
            var hex_str = "";
            var i, n;
            for(i=0; i < char_str.length; i++) {
                  n = charToByte(char_str.charAt(i));
                  if(n != 0) {
                        hex_str += byteToHex(n);
                  }
            }
            return hex_str.toLowerCase();
      }

      function getDateTime() {
            Response.Write(formattedDate);
      }
</SCRIPT>
```

Appendix III – ipg-util.php

```
<?php
	$dateTime = date("Y:m:d-H:i:s");

	function getDateTime() {
		global $dateTime;
		return $dateTime;
	}

	function createHash($chargetotal, $currency) {
		$storename = "10123456789";
$sharedSecret = "sharedsecret";

		$stringToHash = $storename . getDateTime() . $chargetotal
. $currency . $sharedSecret;

		$ascii = bin2hex($stringToHash);

		return hash('sha256',$ascii);
	}

?>
```

Appendix VI – MasterPass

Refer to the following information when integrating MasterPass as a payment method.

MasterPass is a digital wallet solution provided by participating banks and supported by MasterCard. When purchasing online, customers log in to their MasterPass account and select a stored card for the payment. MasterPass allows users to store MasterCard, Maestro, VISA, American Express and Diners cards. Please note that your customers will however only be able to select the card brands that your Store has been set up for in general.

To learn more about MasterPass, please visit www.masterpass.com.

Checkout Process with MasterPass

The checkout process with MasterPass can be initiated with a "BUY WITH MasterPass" button that you place on your website either as a specifically alternative checkout option or next to other payment methods that you offer.

When consumers click this button, you construct a 'sale' or 'preauth' request with the parameter 'paymentMethod' set to 'masterpass'.

This will take your customer to the MasterPass login screen, from there to the subsequent pages of the digital wallet and finally back to your web shop (responseSuccessURL, responseFailURL or reviewURL).

Alternatively you can let your customers select the payment method on the gateway's hosted payment method selection page. If you prefer that option, simply do not submit the parameter 'paymentMethod'.

Good to know prior the integration

- The **Billing Address** for a MasterPass transaction is associated with the card stored inside the wallet thus even if you should use the payment gateway's 'PayPlus' or 'FullPay' mode, there will be no additional entry form for the Billing Address when a customer uses MasterPass. The Billing Address stored in the wallet will also automatically override any billing address data you may send within your transaction request to the gateway. You will always receive the Billing Address from the wallet in the transaction response - even in 'PayOnly' mode, which is different compared to other payment methods.

- If you use the gateway's 'FullPay' mode, the **Shipping Address** can be selected by the customer inside the wallet (no additional page for that from the gateway). If you use the 'PayOnly' or 'PayPlus' mode, the Shipping Address selection in the wallet gets omitted as a non-required step. Thus you can send the Shipping Address with your request and your customers will not have to select/provide it again inside the wallet (it reduces the number of steps in the transaction flow when purchasing e.g. software products available as downloads where no shipping address is really required).

- For the cases where the shipping address and thus **Shipping Fee** is not clear yet when your customer enters the wallet process by clicking the 'BUY WITH Masterpass' button, you can send additional parameters in your transaction request which allow you to present a final confirmation page with the final amount to your customers when they return from the wallet. The parameter 'reviewOrder' needs to be set to 'true' in order to indicate that the final transaction amount needs to be reviewed by your customer before completion. In addition you will need to provide the URL for your confirmation page in the parameter 'reviewURL'. When your customer confirms the final amount on this page, you will need to send a request to finalise the transaction to the 'redirectURL' that you received in your response from the gateway. This final request needs to include: oid, ipgTransactionId, subtotal, shipping, vattax, chargetotal, currency and hashExtended..

- When your Store is activated for **3D Secure**, these settings will also apply to your MasterPass transactions. In the specific case of MasterPass, the authentication process will however be handled by MasterCard inside the wallet. For that reason, the parameter 'authenticateTransaction' has no effect for

MasterPass transactions and the supported programmes are limited to MasterCard SecureCode and Verified by Visa (no American Express SafeKey).

- The **Card Code** (CVV2/CVC2/4DBC) is not required for MasterPass transactions. At the time when a customer adds a card to the wallet, the Card Code gets entered and checked once. No further Card Code entry is required from your customers.

- .

- MasterPass is not available for **Betting/Casino Gambling** merchants (MCC 7995).

Activate MasterPass for your Test Store

- Obtain the credentials for the sandbox consumer accounts listed in the online documentation provided by MasterCard.
- Make sure your payment gateway test Store ID has been enabled for MasterPass.