First Data Merchant Solutions Connect Payment Gateway

Connect® Integration Guide

firstdatams.co.uk

First Data Merchant Solutions

Table of Contents

1 Introduction

First Data Merchant Solutions (FDMS) offers a convenient way to accept online payments.

The FDMS Connect solution offers combined Payment Gateway and Merchant Services in one. This solution provides a simple way for connecting an online store to the FDMS Connect Payment Gateway (Connect Gateway).

The Connect Gateway manages all electronic communications with payment card processors and financial institutions.

This Integration Guide describes how to integrate your website using the Connect Gateway and provides step by step instructions on how to quickly start accepting payments from your web shop.

1.1 Technical Support

This Integration Guide contains the information required for integrating the Connect Gateway.

For additional information on settings, customisation and how to process transactions manually (by keying the information) please refer to the First Data Merchant Solutions Virtual Terminal User Guide.

Technical Support for integration testing is available via email and telephone call coverage from 09:00 to 17:00 Monday to Friday (UK time).

Email: fdmsIPGConnect@firstdatacorp.co.uk

Telephone: 01268 567 137

Web: https://www.firstdatams.co.uk/fdms/en_gb/home.html

2 Payment Page Options

The Connect Gateway provides two options for integration with your website, which is using the FDMS Hosted Payment Page or using your own Payment Page.

2.1 FDMS Hosted Payment Page

The Hosted Payment Page enables you to use ready-made form pages for the payment process that FDMS provides and hosts on First Data servers.

With this option, your Customer will be forwarded to First Data when it comes to payment and can enter the sensitive Cardholder data on First Data SSL-encrypted page, all payment information is encrypted and transmitted only via a secure 128bit SSL (Secure Socket layer) connection.

This option facilitates compliance with the Data Security Standard of the Payment Card Industry (PCI DSS) as the payment processing is completely hosted by First Data. Once the transaction has been completed, the Customer will be re-directed to your web shop and your shop system will be notified of the payment result.

You can customise the FDMS Hosted Payment Page with your own corporate design.

Please contact the Technical Support Team (see section 1.1) if you require this customisation.

2.2 Merchant Payment Page

For Merchants who prefer Customer's not to be re-directed to the First Data Payment Page, you can create your own payment forms in your individual corporate design.

Please note that if you store or process 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).

For this option, you will be required to display a secured website (lock symbol in the browser) to your customer; your website needs to provide a SSL-connection via a HTTPS-Server.

First Data Merchant Solutions PCI DSS Compliance Program, an online service providing you with all the information you require to become and remain compliant with the Payment Card Industry Data Standard (PCI DSS). Access the portal using your PCI DSS Compliance Program log in credentials at https://www.compliance.firstdatams.com.

3 Payment Modes

The Connect Gateway offers three different payment modes to choose from in order to define the range of data that shall be captured.

Depending on your individual business process, you can choose a mode that only collects payment data or decide to additionally transmit details for the invoice or shipping address.

Depending on the complexity of your business processes, you may also want to integrate the FDMS Web Service API solution. Contact the Technical Support Team (see Section 1.1) for more information.

The following sub-sections describe the three available payment modes.

3.1 PayOnly Mode

The PayOnly mode collects a minimum set of information for the transaction via the FDMS Hosted Payment Page.

A single page is presented to the Cardholder to enter the payment information (e.g. credit card number, expiry data and card code) for payment authorisation.

This mode assumes you have already collected the entire Customer's address and contact information on your Web server before re-directing the Cardholder to FDMS Hosted Payment Page. Therefore, if your business requires Customers address details etc., please ensure your website has already done this prior to re-directing the Cardholder to the FDMS Hosted Payment page.

3.2 PayPlus Mode

The PayPlus mode enables the FDMS Hosted Payment Page to collect a full set of billing information.

Once the Cardholder is directed to the Payment Page, they are presented with two pages, one for the billing information and one for the payment information. This mode is a hybrid of the other two other supported modes (i.e. FullPay and PayOnly).

This mode allows you to build a form to collect the information you want to send to the Connect Gateway (there is a standard list of fields, plus you can add your own custom fields); the Connect Gateway then takes over from there, collecting the secure transaction information and reports it to you.

3.3 FullPay Mode

The FullPay mode enables FDMS Payment Page to collect all available information (billing, shipping, and payment information).

FullPay mode allows you to send the order (transaction) total to FDMS and the Connect Gateway will collect all other required information.

In summary, FullPay mode does all the work for you, you only need to pass the charge total to the Connect Gateway via your Website.

4 Connect Gateway Integration

This section provides a simple example on how to integrate your website into the Connect Gateway in FullPay mode assuming a non-mobile device (see Section 6 if you want a payment page flow for mobile devices, i.e. Responsive page).

Examples are provided using ASP, C#/ASP.NET, PHP and JSP.

This section assumes that the developer has a basic understanding of his chosen scripting language.

You will require a Store ID and Shared Secret to enable you progress with the integration. Contact the Technical Support Team (See Section 1.1), if this hasn't already been provided to you.

The following sub-sections will present each scripting language payment sample code with corresponding Hash file (name ipg-util).

The Hash file will includes code for generating a SHA1 Hash as required by FDMS. The provision of a Hash in the example ensures that this Merchant is the only Merchant that can send in transactions for this store.

Please note that the POST URL within the sample codes is for integration testing only. When you are ready to go into production, please contact the Technical Support Team (see Section 1.1).

4.1 ASP Sample Code

The following ASP example demonstrates a simple page that will communicate with the Connect Gateway in FullPay mode.

When the Cardholder clicks 'Submit', they are re-directed to the First Data secure pages, where they can enter their billing, shipping and payment information.

After payment has been completed, the user will be re-directed to the Merchants receipt page. The location of the receipt page can be configured.

4.1.1 ASP Payment File

```
<!-- #include file="ipg-util.asp"-->
<html>
<head><title>FDMS ASP Sample Payment Screen </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="GMT"/>
    <input type="hidden" name="txndatetime" value="<% getDateTime() %>"/>
    <input type="hidden" name="hash" value="<% call createHash(
    "13.00","826" )
%>"/>
    <input type="hidden" name="storename" value="1100000001" />
    <input type="hidden" name="mode" value="fullpay"/>
    <input type="text"   name="chargetotal" value="13.00" />
    <input type="hidden" name="currency" value="826"/>
    <input type="hidden" name="responseSuccessURL"
value="http://yourdomain.com/Thanks" />
    <input type="hidden" name="responseFailURL"
value="http://yourdomain.com/PaymentFailure" />
    <input type="submit" value="Submit">
</form>
</body>
</html>
```

The next subsection describes the ipg-util.asp

4.1.2 ASP Hash File

```
<Script LANGUAGE=JScript RUNAT=Server src="sha1.js">

</SRIPT>
<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 + chargetotal +shared secret + currency (numeric ISO value)
     */
     function createHash(chargetotal, currency) {
          // Please change the store Id to your individual Store ID
          var storename = "1100000001;
          // NOTE: Please DO NOT hardcode the secret in that script. For example read it
from a database.
          var sharedSecret = "Test123";
          var stringToHash = storename + formattedDate + chargetotal + currency +
sharedSecret;
          var ascii = getHexFromChars(stringToHash);
          var hash = calcSHA1(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>
```

Note, the included file, **ipg-util.asp** uses a server side JavaScript file to build the SHA1 Hash. This file can be provided on request (See Section 1.1). 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.

Please ensure the time zone set in the payment form corresponds to your System/Server time zone.

4.2 C#/ASP.NET Sample Code

The following .NET C# example demonstrates a simple page that will communicate with the

Connect Gateway in FullPay mode.

When the Cardholder clicks ‘Submit’, they are redirected to the First Data secure pages, where they can enter 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.

4.2.1 C#/ASP.NET Payment File

```
<html>
<head><title>FDMS C#/ASP.NET Sample Payment Screen </title></head>
<body>
<p><h1>Order Form</h1></p>
<form method="post" action="https://test.ipg-
online.com/connect/gateway/processing">
    Sale: <input type="text" name="txntype" value="sale" /><br />
    Time Zone: <input type="text" name="timezone" value="<%=strtimezone%>"
/><br />
    Date Time: <input type="text" name="txndatetime"
value="<%=txndatetime%>" /><br />
    Hash: <input type="text" name="hash" value="<%=result%>" /><br />

    Store Name: <input type="text" name="storename" value="1100000001" /><br
/>
    Mode: <input type="text" name="mode" value="fullpay" /><br />
    Charge: <input type="text" name="chargetotal" value="<%=chargetotal%>"
/><br />
    Currency: <input type="text" name="currency" value="826" /><br />
    <input type="hidden" name="responseFailURL"
value="http://yourdomainname.com/PaymentFailure.aspx”/><br />

    <input type="hidden" name="responseSuccessURL" value=""http://
yourdomainname.com/Thanks.aspx" /><br />
    <input type="submit" value="Submit" />
```

```
</form>
<body>
</html>
```

4.2.2 C#/ASP.NET Hash File

```
public string txndatetime, storename, chargetotal, sharedsecret, result,
currency, strtimezone;

    protected void Page_Load(object sender, EventArgs e)
    {
        txndatetime = DateTime.Now.ToString(@"yyyy\:MM\:dd-HH\:mm\:ss");
        storename = "1100000001";
        chargetotal = "31.00";
        sharedsecret = "Test123";
        currency = "826";
        strtimezone = "GMT";
        string stringToHash = storename + txndatetime + chargetotal + currency +
sharedsecret;
        string hash1 = getSHA1(stringToHash);
        string hash = computeHash(hash1);
    }
    public string computeHash(string value)
    {
        byte[] valueBytes = Encoding.ASCII.GetBytes(value);
        SHA1 sha1Alg = SHA1Managed.Create();
        byte[] resultBytes = sha1Alg.ComputeHash(valueBytes);
        result = BitConverter.ToString(resultBytes).Replace("-", "").ToLower();
        return result;
    }
    public string getSHA1(string stringa)
    {
        byte[] ascii = Encoding.ASCII.GetBytes(stringa);
        foreach (Byte b in ascii)
        {

            result += b.ToString("X");
        }
        return result.ToLower();
    }
```

Note: Please ensure the time zone set in the payment form corresponds to your System/Server time zone.

4.3 JSP Sample Code

The following JSP example demonstrates a simple page that will communicate with the Connect Gateway in FullPay mode.

When the Cardholder clicks ‘Submit’, they are re-directed to the First Data secure pages, where they can enter card details.

After payment has been completed, the user will be re-directed to the Clients receipt page. The location of the receipt page can be configured.

4.3.1 JSP Payment File

```
<html>
  <head><title>FDMS JSP Sample Payment Screen</title></head>
 <body><% Date currentDate = new Date(); %>
   <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="GMT"/>
     <input type="hidden" name="txndatetime" value="<%=
DATE_FORMAT.format(currentDate) %>"/>
     <input type="hidden" name="hash" value="<%= createHash( "13.00","826",
currentDate ) %>"/>
     <input type="hidden" name="storename" value="1100000001"/>
     <input type="hidden" name="mode" value="fullpay"/>
     <input type="text" name="chargetotal" value="13.00" />
     <input type="hidden" name="currency" value="826"/>
     <input type="hidden" name="oid" value="unique01"/>
   <input type="hidden" name="responseFailURL"
value="http://yourdomainname.com/PaymentFailure.jsp”/>
   <input type="hidden" name="responseSuccessURL" value=""http://
yourdomainname.com/Thanks.jsp" /><
     <input type="submit" value="Submit">
   </form>
</body>
</html>
```

4.3.2 JSP Hash File

```
<%@     page import="java.security.MessageDigest, java.text.SimpleDateFormat,
java.util.Date, java.io.UnsupportedEncodingException" %>

    <%!
    private static final String STORE_ID = "1100000001";
    private static final String SHARED_SECRET = "Test123";
    private static final SimpleDateFormat DATE_FORMAT = new
SimpleDateFormat("yyyy:MM:dd-HH:mm:ss");

    private String createHash(String chargetotal, String currency, Date
dateToUse) throws UnsupportedEncodingException    {

      String formattedDate = DATE_FORMAT.format(dateToUse);

      // NOTE: Please DO NOT hardcode the secret in that script. For example read
it from a database.
      String stringToHash = STORE_ID + formattedDate + chargetotal + currency +
SHARED_SECRET;
      String hexString = toHex(stringToHash);
      String hash = stringToSha1(hexString);
      return hash;
    }
    private String toHex(String value) throws UnsupportedEncodingException
    {
      byte[] bytes = value.getBytes("ISO-8859-1");
      return toHex(bytes);
    }
    private String toHex(byte[] bytes) throws UnsupportedEncodingException
    {
      StringBuilder str = new StringBuilder();
      for (int i = 0; i < bytes.length; i++){
        str.append(String.format("%02x", bytes[i]));
      }
      return str.toString();
    }

      // continue on next page of this guide
```

```
  private String stringToSha1(String stringToEncode) {
    MessageDigest md = null;
    String result = null;
    try {
      byte[] bytes = stringToEncode.getBytes("ISO-8859-1");
      md = MessageDigest.getInstance("SHA-1");
      byte[] encyptedString = md.digest(bytes);
      result = toHex(encyptedString);
    }
    catch(Exception e) {
      e.printStackTrace();
    }
    return result;
  }
%>
```

Note: Please ensure the time zone set in the payment form corresponds to your System/Server time zone.

4.4 PHP Sample Code

The following PHP example demonstrates a simple page that will communicate with the Connect Gateway in FullPay mode.

When the Cardholder clicks 'Submit', they are re-directed to the First Data secure pages, where they can enter their shipping, billing and payment information.

After payment has been completed, the user will be re-directed to the Merchants receipt page. The location of the receipt page can be configured.

4.4.1 PHP Payment File

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

<html>
<head><title>FDMS PHP Sample Payment Screen</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="CET"/>
    <input type="hidden" name="txndatetime" value="<?php echo getDateTime() ?>"/>
    <input type="hidden" name="hash" value="<?php echo createHash( "13.00","826"
) ?>"/>
    <input type="hidden" name="storename" value="1100000001"/>
    <input type="hidden" name="mode" value="fullpay"/>
    <input type="text"   name="chargetotal" value="13.00"/>
    <input type="hidden" name="currency" value="826"/>
    <input type="hidden" name="responseSuccessURL"
value="http://yourdomain.com/Thanks" />
    <input type="hidden" name="responseFailURL"
value="http://yourdomain.com/PaymentFailure" />
    <input type="submit" value="Submit">
    </form>
    </body>
</html>
```

4.4.2 PHP Hash File

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

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

    function createHash($chargetotal, $currency) {
        $storename = "1100000001";
        $sharedSecret = "Test123";

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

        $ascii = bin2hex($stringToHash);

        return sha1($ascii);
    }

?>
```

Note: Please ensure the time zone set in the payment form corresponds to your System/Server time zone.

5 Mandatory Form Fields

Depending on the transaction type, the following form fields must be present in the form being

Field name	Description, possible values and format	"Sale" transaction	"PreAuth" (Credit Cards only)	"PostAuth "(Credit Cards only)	Void
txntype	'sale', 'preauth', 'postauth' or 'void' (please note the descriptions of transaction types in the User Guide) The possibility to send a 'void' using the Connect interface is restricted. Please contact technical support team if you want to enable this feature.	X (sale)	X (preauth)	X (postauth)	X (void)
timezone	GMT, CET or EET (timezone of the transaction)	X	X	X	X
txndatetime	YYYY:MM:DD-hh:mm:ss (exact time of the transaction)	X	X	X	X
hash	This is a SHA1 hash of the following fields : storename + txndatetime + chargetotal + currency + sharedsecret. Note, that it is important to have the hash generated in this exact order. See sample codes in Section 4.	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)	X	X		

submitted to the gateway (X = mandatory field).

Field name	Description, possible values and format	"Sale" transaction	"PreAuth" (Credit Cards only)	"PostAuth "(Credit Cards only)	Void
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 Pounds and 34 Pence. Group separators like (1,000.01 / 1.000,01) are not allowed.	X	X	X	X
currency	The numeric ISO code of the transaction currency, e. g. 826 for GBP	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

Mandatory Form Fields (Continuation)

6 Optional Form Fields

Field name	Description, possible values and format	
mobileMode	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	
paymentMethod	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 payment method along with your Sale or PreAuth transaction If you do not submit this parameter, the Connect Gateway will display a drop- down menu to the customer to choose from the payment methods available for your shop Valid values are:	
	Payment method	**Value**
	MasterCard	***M***
	Visa (Credit/Debit/Electron/Delta)	***V***
	Diners	***C***
	American Express	***A***
	Maestro	***MA***
	Maestro UK/Solo	***maestroUK***
oid	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 Please ensure the Order ID is unique for every transaction, a duplicate Order ID will cause the Connect Gateway to decline a transaction	
customerid	This field allows you to transmit any value, e.g. your ID for the customer	
invoicenumber	This field allows you to transmit any value, e.g. invoice number or class of goods	
vattax	This field allows you to transmit tax, ensure the sub total amount plus tax equals the charge total	
refer	This field describes who referred the Customer to your store	
comments	Place any comments here about the transaction	
responseSuccessURL	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	
responseFailURL	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	

Optional Form Fields (Continuation)

<table>
<tr><th>Field name</th><th>Description, possible values and format</th></tr>
<tr><td rowspan="8">language</td><td colspan="2">This value can be used to override the default payment page language configured for your merchant store. The following values are currently possible:</td></tr>
<tr><td>Language</td><td>language</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>hashExtended</td><td colspan="2">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>trxOrigin</td><td colspan="2">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>full_bypass</td><td colspan="2">This parameter allows you to avoid the FDMS Hosted Payment Page when using your own input forms for the payment process you can transmit this parameter with the value true

full_bypass=true

This will enable you to get the result of validity checks performed by the Connect Gateway (see section 7) back in the transaction response and can display your own error page based on this</td></tr>
</table>

7 Validations

Prior to the authorisation request for a transaction the Connect 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 Connect Gateway presents a corresponding error page to the Cardholder (see Section 6 Optional parameter ***full_bypass***).

8 Merchant Owned Payment Page

Merchants, who wish to use their own Payment Page without the need to re-direct the Cardholder to the FDMS Hosted Payment Page, shall be required to submit additional mandatory fields to the fields in Section 5 in this Guide.

It is also strongly recommended for the Merchant to check if JavaScript is activated in a Customer's browser and inform the Customer that JavaScript needs to be activated for the payment process to work.

The following sub-sections shall detail the fields required for Merchants using their own Payment Page depending on the payment mode selected.

8.1 PayOnly Mode

Merchants using their own Payment Page with the *PayOnly* mode will present an 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 in Section 5, your form needs to contain the following fields- part of them can be hidden, (X = mandatory field).

Field name	Description, possible values and format	Credit Card (+Visa Debit / Electron / Delta)	Maestro	Maestro UK
cardnumber	Your Customer's card number. 12-24 digits	X		X
expmonth	The expiry month of the card (2 digits)	X	X	X
expyear	The expiry year of the card (4 digits)	X	X	X
cvm	Card security code at the back of the card (3 to 4 digits)	X	X as an optional field "if on card"	(X)
Issuenumber	UK Maestro issue number (1 to 2 digits)			(X) Mandatory if cvm not set

8.2 PayPlus Mode

In addition to the Cardholder details (see Section 8.1); Clients using their own Payment Page with the *PayPlus* mode can submit billing information to the Connect Gateway.

The following table also 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 45 Characters	Customers Email Address

8.3 FullPay Mode

In addition to the Cardholder details (see Section 8.1), Merchants using their own Payment Page with the *FullPay* mode can submit shipping information to the Connect Gateway. The Connect Gateway will echo back the additional details in the authorisation response.

The following table describes the format of these additional 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 spaces	State, Province or Territory
scountry	2 letter country code	Country of Shipping Address
szip	International Postal Code	Zip or Postal Code

9 Additional Custom Fields

You may send as many custom fields to the Connect 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 15 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.

Ensure your website and privacy policy explains about the data you collect from Customers and how you will use it.

10 3DSecure

The Connect Gateway includes the ability to authenticate transactions using Verified by Visa and MasterCard SecureCode. 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 you 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 (GICC ECI 7).

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 is not yet determined and the session expires (20 minutes) 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 for 3DSecure 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.

Contact the Technical Support Team (see Section 1.1) to arrange testing.

11 Data Vault

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

11.1 Store or Update Card details

The Data Vault product enables you to store or update payment information when performing a transaction. To perform this, 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 will be stored under this ID (***hosteddataid***) if the transaction was successful. In cases where the submitted 'hosteddataid' already exists for your store, the stored payment information will be updated.

11.2 Initiate Payment Transaction

The Data Vault product enables you to initiate payment transactions using stored data**.**

If you stored Cardholder information using the Data Vault option (see Section 11.1), you can perform transactions using the 'hosteddataid' without the need to pass the credit card data again.

Please note that it is not allowed to store the card security 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 security code (CVV2).

When using multiple Store IDs, it is possible to access stored card data records of a different Store ID than 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'.

11.3 Avoiding Duplicate Card details

The Data Vault product enables you to 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' (see Section 11.1), the transaction will be declined.

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

12 Recurring Payments

For card transactions, it is possible to install recurring payments using The Connect Gateway.

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 999	Number of installments to be made including the 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 the Connect Gateway can be modified or cancelled using the Virtual Terminal or Web Service API.

13 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 See section 13.2
oid	Order ID
refnumber	Reference number
status	Transaction status
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 This parameter 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 1 Hash using the following parameter string: ***sharedsecret + approval_code + chargetotal + currency + txndatetime + storename*** 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
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. For card payments, the response code '00' is the most common response for an approval

Transaction Response (Continuation)

Field name	Description
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 DINERS/DISCOVER MAESTRO Filled with "N/A" for any payment method which is not a Credit card or Debit card
response_code_3dsecure NB: This is only applicable to 3DSecure transactions.	Return code indicating the classification of the transaction: **1** – Successful authentication (GICC ECI 11/10) **2** – Successful authentication without AVV (GICC ECI 11/10) **3** – Authentication failed / incorrect password (transaction declined) **4** – Authentication attempt (GICC ECI 13/12) **5** – Unable to authenticate / Directory Server not responding (GICC ECI 7) **6** – Unable to authenticate / Access Control Server not responding (GICC ECI 7) **7** – Cardholder not enrolled for 3D Secure (GICC ECI 13/12) **8** – Invalid 3D Secure values received, most likely by the credit card issuing bank's Access Control Server (ACS)

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
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.

13.1 Transaction Notification

The FDMS Payment Gateway can send the transaction response (result parameters defined in the table above) to a defined URL.

To use this notification method, you can specify a URL in the Customisation section of the Virtual Terminal or submit the URL in the following additional transaction parameter:

transactionNotificationURL

```
<input type="hidden" name="transactionNotificationURL"
value="http://yourdomain.com/Notification" />
```

Please note that

- No SSL handshake, verification of SSL certificates will be done in this process
- The Notification URL needs to listen on port 80 (http)– other ports are not supported
- The response hash parameter for validation (using SHA1 algorithm) 'notification_hash' is calculated as follows:

  chargetotal + sharedsecret + currency + txndatetime + storename + approval_code

13.2 Approval Code Details

An example Approval Code for a successful transaction will look like this:

Y:356887:0000144820:PPXM:0612789753. Below table explains the values.

<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>Y</td><td>Successful</td></tr>
<tr><td>356887</td><td>Issuer’s approval code or processor reference number</td></tr>
<tr><td>0000144820</td><td>FDMS EMEA Payment Gateway Internal ID</td></tr>
<tr><td>PPXM</td><td>First three characters (PPX) indicate Address Verification Service Results:
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>PPX</td><td>No address data provided or Address not checked by the Card Issuer.</td></tr>
<tr><td>YYY</td><td>Card Issuer confirmed that street and postcode match with their records.</td></tr>
<tr><td>YNA</td><td>Card Issuer confirmed that street matches with their records but postcode does not match.</td></tr>
<tr><td>NYZ</td><td>Card Issuer confirmed that postcode matches with their records but street does not match.</td></tr>
<tr><td>NNN</td><td>Both street and postcode do not match with the Card Issuer’s records</td></tr>
<tr><td>YPX</td><td>Card Issuer confirmed that street matches with their records. The Issuer did not check the postcode.</td></tr>
<tr><td>PYX</td><td>Card Issuer confirmed that postcode matches with their records. The Issuer did not check the street.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td>
<table>
<tr><td>XXU</td><td>Card Issuer did not check the AVS information</td></tr>
</table>
The last alphabetic character in the middle (M) is a code indicating whether the card security code matched the card-issuing bank's code. An "M" indicates that the code matched.
This code may or may not be present, depending on whether the card security code was passed and the service was available for the type of card used. Below is a table showing all the possible return codes and their meanings.
<table>
<tr><th>Value</th><th>Meaning</th></tr>
<tr><td>M</td><td>Card Security Code Match</td></tr>
<tr><td>N</td><td>Card Security Code does not match</td></tr>
<tr><td>P</td><td>Not processed</td></tr>
<tr><td>S</td><td>Merchant has indicated that the card security code is not present on the card</td></tr>
<tr><td>U</td><td>Issuer is not certified and/or has not provided encryption keys</td></tr>
<tr><td>X</td><td>No response from the credit card association was received</td></tr>
<tr><td></td><td>A blank response should indicate that no code was sent and that there was no indication that the code was not present on the card.</td></tr>
</table>
</td></tr>
<tr><td>0612789753</td><td>FDMS EMEA Payment Gateway Internal ID</td></tr>
</table>

An example decline or failed transaction will look like this: ***N:54: EXPIRED CARD***

END OF DOCUMENT