sage pay

Form Protocol and Integration Guideline

Document Index

Welcome to the Sage Pay Form Integration Method

The Sage Pay payment system provides a secure, simple means of authorising credit and debit card transactions from your website.

The Sage Pay system provides a straightforward payment interface for the customer, and takes complete responsibility for the online transaction, including the collection and encrypted storage of credit and debit card details, eliminating the security implications of holding such sensitive information on your own servers.

The Sage Pay Form integration method is designed for merchants who use shopping carts, have less experience in server side scripting, or who use shared web servers that do not offer database services. With Sage Pay Form, all transaction information is held at Sage Pay, including the full shopping basket contents, and e-mails are sent from the Sage Pay servers to you and your customers to confirm the success or failure of the transaction.

The customer is redirected to Sage Pay to enter their card details, so no sensitive information needs to be taken or stored on your site (removing the need for you to maintain highly secure encrypted databases, or obtain digital certificates).

This document explains how your website should communicate Sage Pay Form, goes on to explain how to integrate with our testing and live environments, and contains the complete Sage Pay Form Payment Protocol in the Appendix.

Overview of how Form integrated Payments work

The final "Pay Now" button on your website is your link to the Sage Pay System. Once the customer has selected their purchases, entered delivery details, billing address and so forth, all on your own site, and pressed the final proceed button, a small script on your server generates an order summary page, listing the customer's contact details, their purchases and the total order amount. At the bottom of that page is a "Pay Now" button that submits the information on that page to the Sage Pay Form payment gateway.

What the customer does not see is that whilst generating that order summary page, a simple and easy to modify piece of server-side scripting builds an encrypted hidden field that it places on the form. This field contains all the transaction information in a format that Sage Pay Form can understand. When the user clicks the "Pay Now" button, the encrypted contents of that field get POSTed to the Sage Pay system and the customer is presented with the Sage Pay payment pages, where they enter their credit/debit card details, security codes and billing address (if you have not already captured it). The Sage Pay Form payment page carries your logo, and a description (sent by your site) of the goods that the customer is paying for, so they can remain confident they are buying from you. You can even customise those payment pages to carry the look and feel of your site at no additional cost.

Once the customer has selected their payment method and entered the details, they are shown a full summary of their order (including basket contents if you have passed them to us) and asked to confirm that they wish to proceed. Sage Pay Form then requests authentication from the card issuing bank (where appropriate) and requests authorisation from your acquiring bank. Once the bank has authorised the payment (and assuming the address and card value checks have passed any rules you may have set up), it redirects your customer back to the successful payment page on your site. If the authorisation fails, the Sage Pay Form system redirects the customer to your order failure page. Both pages are sent encrypted information which you can decrypt, again using simple provided scripts, to find out what happened to the transaction and extract any useful information.

If you provide the Sage Pay Form system with an e-mail address for you and/or the customer, it also sends confirmation e-mails in the event of a successful order. If the order fails, you are mailed with details of that failure but the customer is not.

Sage Pay provides Integration Kits, which are simple worked examples in various different scripting languages that perform all the tasks described above. You simply customise these to work with your particular environment. So whether you are running .NET, ASP, PHP, Coldfusion, or Java, and whether your servers are Linux Apache or Win32 IIS, we've already done half of the work for you.

The following sections explain the integration process in more detail. The complete Sage Pay Form Payment protocol is attached in the appendix, providing a detailed breakdown of the contents of the encrypted fields sent between your servers and ours during a payment.

The Form Payment integration Process in Detail

This section details the messages exchanged between your Web servers and the Sage Pay Form system.

Step 1: The customer orders from your site.
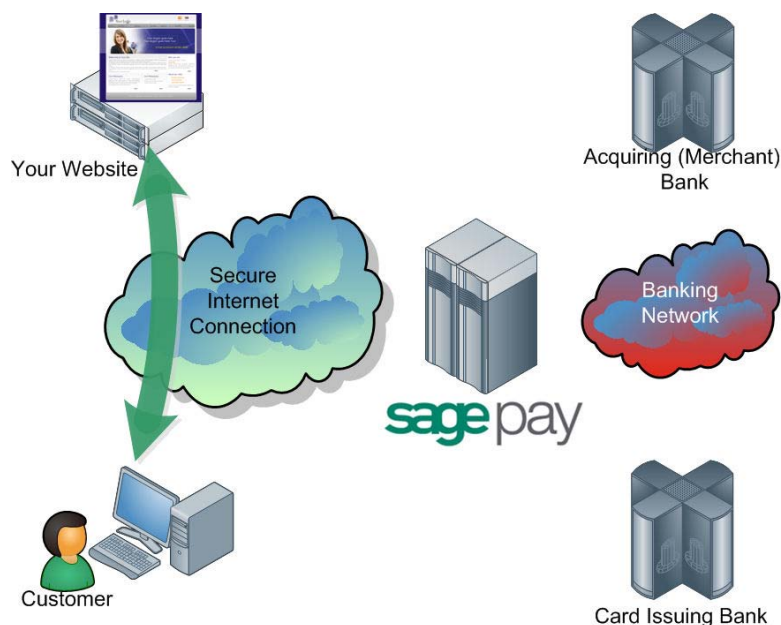
A payment begins with the customer ordering goods or services from your site. This process can be as simple as selecting an item from a drop down list, or can involve a large shopping basket containing multiple items with discounts and delivery charges. Your interaction with your customer is entirely up to you and the Sage Pay Form system only requires you to collect a few compulsory pieces of information, which are detailed in the latter part of this guide.

It is generally a good idea to identify the customer by name, e-mail address, delivery and billing address and telephone number. It is also helpful to have your server record the IP Address from which the user is accessing your system. You should store these details in your session alongside details of the customer's basket contents or other ordered goods.

YOU DO NOT NEED TO COLLECT CREDIT OR DEBIT CARD DETAILS. All your site needs to do is calculate the total cost of the order in whatever currency your site operates and present the user with a confirmation page, summarising their order and containing the transaction detail in an encrypted hidden field (see below).

Step 2: Your server builds a Confirmation Page.

Your server-side script will build an order confirmation page, displaying the full details of the purchase to the customer, including their billing and delivery addresses, basket contents, total order value and contact details.

This script will also place an HTML FORM on that page with the **action** set to the Sage Pay Form registration page. That form will also contain four hidden fields:

- **VPSProtocol** - which lets our system know which version of our messages you are using (the current version is 2.23).
- **TxType** - which lets us know which type of transaction you wish to perform. In most cases this is PAYMENT.
- **Vendor** - your unique company identifier, assigned to you by Sage Pay.
- **Crypt —** A field containing encrypted and encoded details of the transaction. This prevents the customer from being able to tamper with the contents of the order before they are submitted to us.

The contents of the crypt field are built by your script and include, amongst other things:

- A unique reference to this transaction that you generate (the VendorTxCode).
- Total transaction value and currency.
- The URLs of the order Success and Failure pages.
- Customer e-mail address for confirmation e-mails.
- Your e-mail address for notification e-mails.
- Billing and Delivery Addresses and Post Codes.
- Basket Contents and a Description of Goods.

See Appendix A for the full protocol which lists all the fields you can send if you wish, and those which are compulsory for all transactions.

The integration kits we provide contain scripts in a variety of languages that illustrate how you compose and send this message from your server to ours. Please visit the developer's area to download your kit at: http://www.sagepay.com/developers.asp

When the customer clicks the "Pay Now" button on the form, the hidden field is POSTed to Sage Pay and the customer's browser is redirected there (see the diagram overleaf).
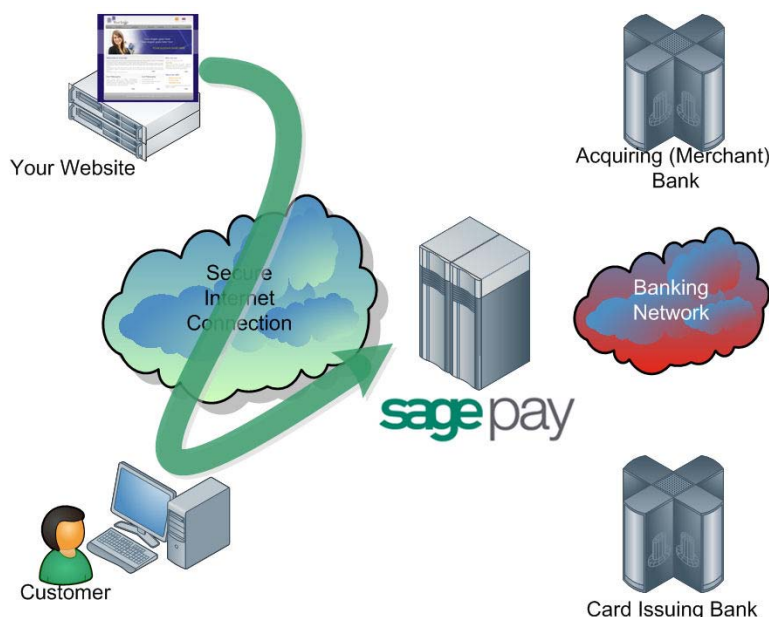
The Sage Pay system begins by validating the Crypt field contents. It first checks to ensure all the required fields are present, and that their format is correct. If any are not present or contain the wrong type of data, a validation error is sent to your Failure page if possible, or displayed on screen.

This normally only happens in the development stage so your customers are unlikely to encounter this page.

If all fields are present and correct, the information in those fields is then validated. The Vendor name is checked against our database and the currency of the transaction is validated against those accepted by your merchant accounts. The VendorTxCode is checked to ensure it has not been used before. The Basket contents are validated to ensure they have been sent in a format the Sage Pay Form system understands. The amount field is validated. Flag fields are checked... every field, in fact, is checked to ensure you have passed appropriate values.

If everything in the POST checks out, the transaction is registered with the Sage Pay Form system and a new transaction code is generated that is unique across ALL merchants using the Sage Pay systems, not just unique to you. This code, the VPSTxId (or Transaction ID), is our unique reference to the transaction, and is sent back to you at the transaction completion stage (see below).

Step 3: Customer enters card details on Sage Pay's Server.
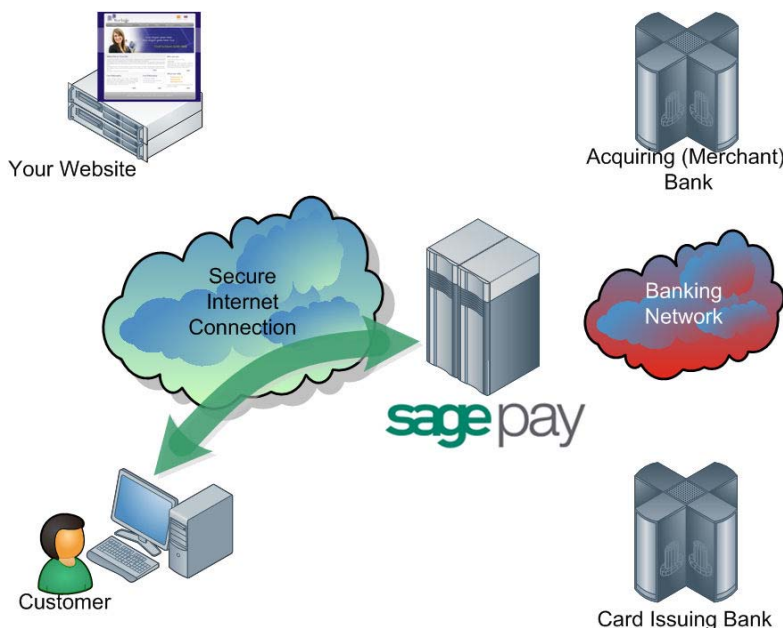
The customer is presented with a card selection page requesting their credit/debit card details. If you are a certified PayPal Business account holder and you have activated PayPal on your Sage Pay account, the PayPal option will also be displayed to your shoppers on this page. For further information about adding PayPal as a payment option on your payment pages, please visit the website:
http://www.sagepay.com/developers.asp

The card selection page will contain your company logo and the description of goods passed in Step 2 above. You can elect to customise these pages further by producing your own custom templates (please contact templates@sagepay.com if you require more information about custom templates).

Once the customer has entered their details, the Sage Pay Form system verifies that information prior to communicating with the bank, to ensure the card number is valid, the card type matches the card number, the expiry date is not in the past and, where appropriate, the issue number and start date are in the correct format. If the customer selects PayPal on the card selection page, the customer is redirected to PayPal to select their payment method, before being returned to the Sage Pay order confirmation screen.

If valid card details have been entered, the customer is presented with an order confirmation screen where they have one last chance to change their mind and cancel the transaction. If the customer decides to cancel, you will be sent a cancellation message to your Failure URL and the customer redirected there. (see step 8 below).

If the customer wishes to continue Sage Pay initiate 3D-Secure authentication checks.

Step 4: The Sage Pay System checks 3D-Secure enrolment.
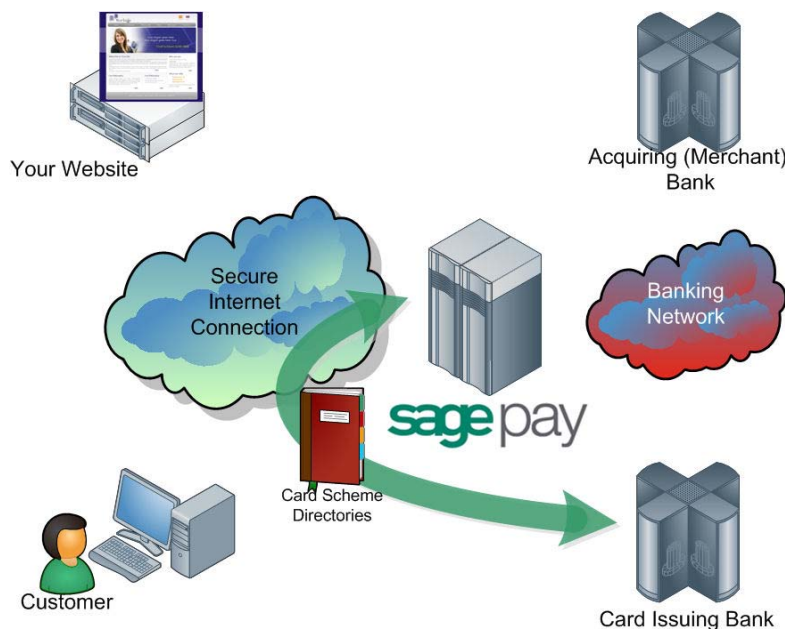
The Sage Pay's servers send the card details provided by your customer to the Sage Pay 3D-Secure Merchant Plug-In (MPI). This formats a verification request called a VEReq, which is sent to the 3D-Secure directory servers to query whether the card and card-issuer are part of the 3D-Secure scheme.

The 3D Secure directory servers send a verification response called a VERes back to our MPI where it is decoded, and the Sage Pay system is informed of the inclusion or exclusion of the card.

If the card or the issuer is not part of the scheme, or if an MPI error occurs, our server will check your 3D-Secure rule base to determine if authorisation should occur. By default you will not have a rule base established and transactions that cannot be 3D-authenticated will still be forwarded to your acquiring bank for authorisation.

If you do have a rulebase set up, our systems check the rules you have in place to determine whether you wish the customer to proceed with authorisation, or you require them to select a different payment method. In such circumstances the shopper will be returned to the card selection page for another attempt. After the 3rd unsuccessful attempt, Sage Pay Form will redirect the customer to your Failure URL (see step 8 below) with a **Status** of **REJECTED** and a **StatusDetail** indicating the reason for the failure. The **3DSecureStatus** field will contain the results of the 3D-Secure lookup. REJECTED transactions will never be authorised and the customer's card never charged, so your code should redirect your customer to an order failure page, explaining why the transaction was aborted.

If your rule base DOES allow authorisation to occur for non-3D-authenticated transactions, then the Sage Pay Form system continues with the authorisation process (jump ahead to step 7).

In most cases 3D-secure verification will be possible and process continues below.

## Step 5: Sage Pay Form redirects your customer to their Card Issuing Bank.
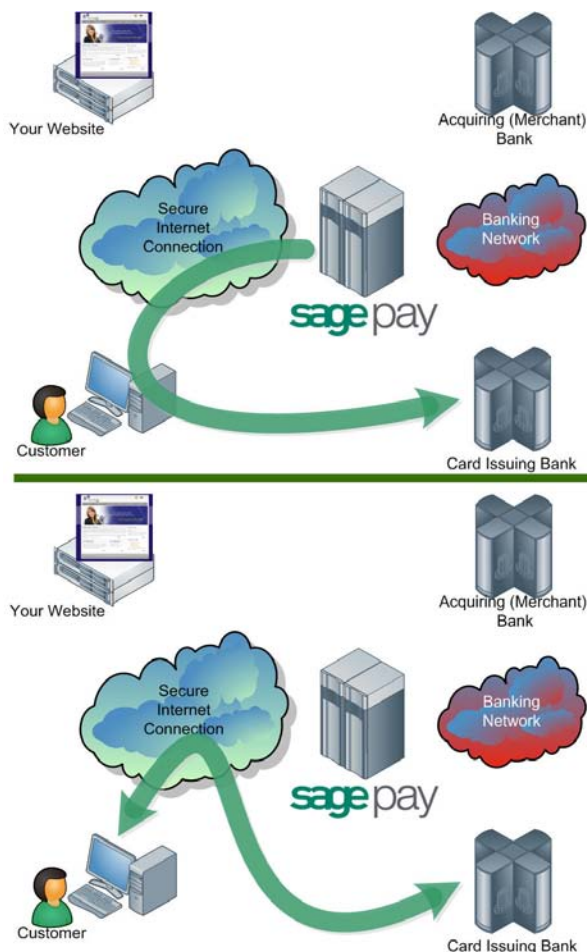
The customer's browser is redirected to their Card Issuing Bank's 3D-Secure authentication pages. These vary from bank to bank, but their purpose is to require the customer to authenticate themselves as the valid card holder.

3D-Secure is much like an online version of Chip and Pin. The customer must answer questions at their card issuer site (these might be a simple password, characters from a password, or numbers generated via card devices, depending on the level of security employed by the bank) and in so doing, the bank is validating the customer's right to use the card for the transaction on your site.

If they determine that the person attempting the transaction IS the real card holder, they assume the liability for fraudulent use of that card and you are protected from what are known as 'Chargebacks' if the cardholder subsequently claims that their card weas used fraudulently.

This level of protection for you is ONLY afforded by 3D-Secure, which is why it is a good idea to keep it enabled on your merchant account through Sage Pay. We set all new accounts with 3D-Secure active by default.

Step 6: The Issuing Bank returns the customer to Sage Pay Form.
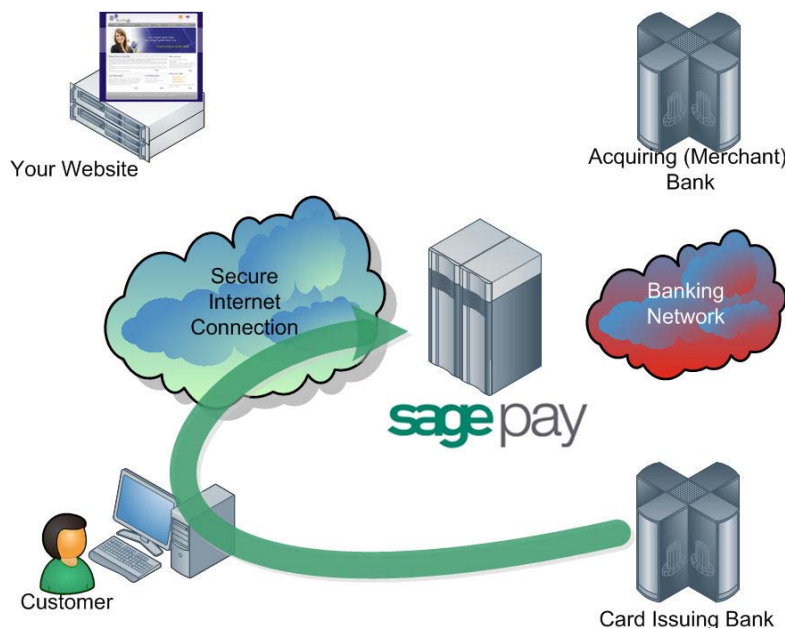
If the customer successfully completes 3D-Authentication with their bank, they are redirected to Sage Pay along with a unique authentication value (called CAVV for cards issued by Visa, and UCAF for MasterCard issued cards). This is passed to your acquiring bank during authorisation (see step 7 below) to secure the liability shift for the transaction.

If the customer does not successfully 3D-Authenticate with their issuing bank, they are passed back to the Sage Pay's server anyway, but without the CAVV/UCAF value. At this stage the Sage Pay Form system consults your 3D-Secure rule base to see if authorisation should be attempted. By default 3D-Authentication failures are NOT sent for authorisation, but all other message types are. Refer to the Sage Pay Rulebase Guide for more information about using 3D-Secure and AVS/CV2 rules.
If authorisation is not possible, your customer is returned to the card selection screen to choose an alternative payment method. After three failed attempts, the Sage Pay servers will redirect your customer to your FailureURL with a Status of **REJECTED** (see step 8), otherwise an authorisation will be gained from your acquiring bank (as in step 7).

Step 7: The Sage Pay Servers request card authorisation.
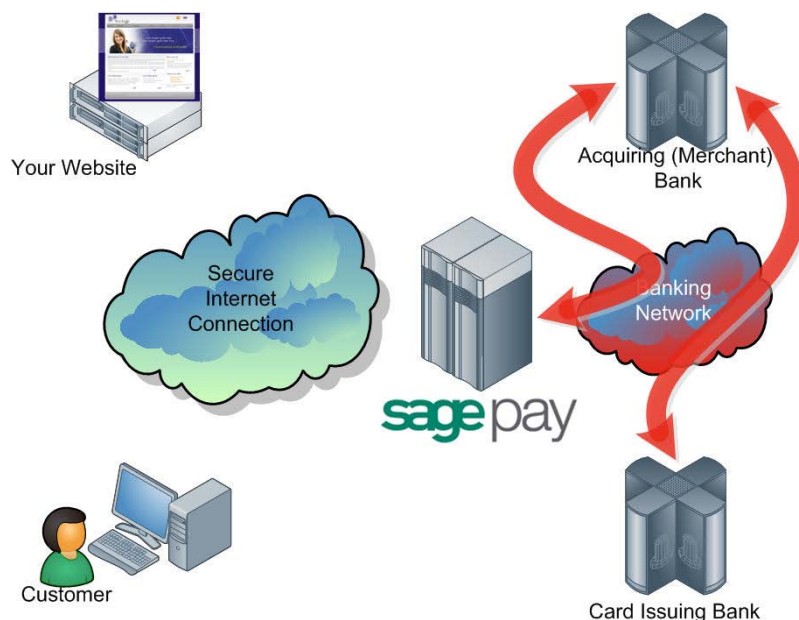
The Sage Pay services format a bank specific authorisation message (including any 3D-Secure authentication values where appropriate) and pass it to your merchant acquirer over the private banking network.

The request is normally answered within a second or so with either an authorisation code, or a failure message. This is obtained directly from the issuing bank by the acquiring bank in real time.

Whilst this communication is ongoing, the customer is shown a page containing the text, "Please wait while your transaction is authorised with the bank".

The Sage Pay Form system handles all authorisation failures in the same way, replying to your site with a **NOTAUTHED** message and a blank authorisation code (after three failed attempts. The first two failures return the customer to the card selection screen to try another card). If the acquirer does return an Authorisation code, Sage Pay Form prepares an **OK** response to send back to you (next step).

If AVS/CV2 fraud checks are being performed, the results are compared to any rule bases you have set up (see the Fraud Screening companion documentation for more information). If the bank has authorised the transaction but the card has failed the fraud screening rules you have established, Sage Pay Form immediately reverses the authorisation with the bank, requesting the shadow on the card for this transaction to be cleared, and prepares a **REJECTED** response for your website.

Please note: Some card issuing banks may decline the online reversal which can leave an authorisation shadow on the card for up to 10 working days. The transaction will never be settled by Sage Pay and will appear as a failed transaction in *My Sage Pay*.

Step 8: Sage Pay Form Redirects the Customer to your site.
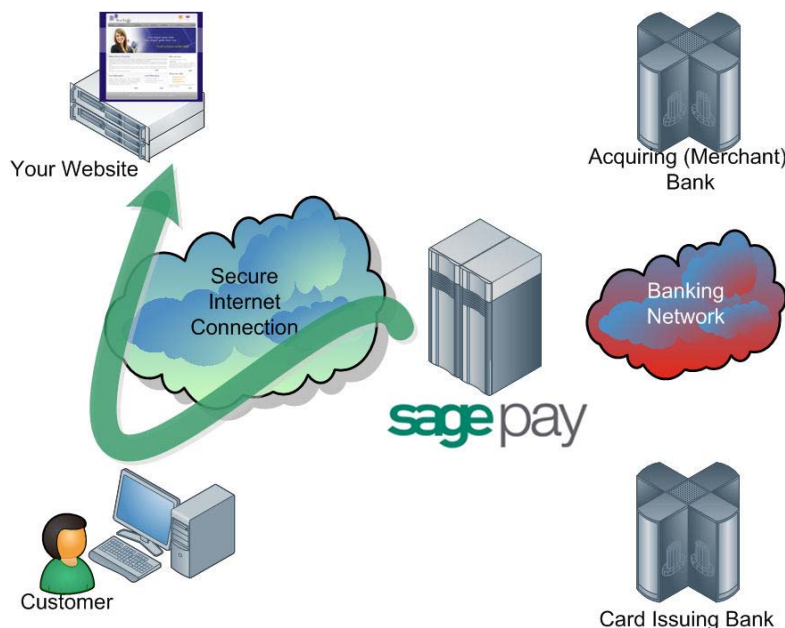
Depending on the result of the authorisation with the bank, your customer is either returned to your SuccessURL (the successful order completion page you supplied in step 2), or your FailureURL for all other transactions.

Appended to the SuccessURL or FailureURL is an encrypted field, again called Crypt, which contains the status of the transaction, the reference codes for those transactions and the fraud checking results. This field is decoded in the same manner that your original script was encoded, using the same password (which is known only to you). The contents of the Crypt field are detailed in Appendix A2.

The Status field holds either:
- o **OK** if the transaction was authorised at step 7
- o **NOTAUTHED** if the authorisation was failed by the bank
- o **ABORT** if the user decided to cancel the transaction whilst on the Sage Pay site
- o **REJECTED** if authorisation occurred but your fraud screening rules were not met, or 3D-Authentication failed three time
- o **ERROR** if an error has occurred at Sage Pay (these are very infrequent, but your site should handle them anyway. They normally indicate a problem with authorisation).

The StatusDetail field contains a human readable description of the error message.

You may wish to display some of the information contained in the crypt field to your customer, especially the reason for failure (if authorisation could not be taken). You are not required to store any of the information sent to you in a database, but if you have access to one, you may wish to do so.

You will receive an e-mail (if you supplied a VendorEMail address) with all these details in, plus details of the order and the customer who placed it. Sage Pay cannot guarantee that the e-mail will always arrive in a timely manner, however, since we have no control over what happens to it once it leaves our servers. You should not rely solely on e-mail confirmation, but regularly check your *My Sage Pay* admin area for new orders (see later).

The real time processing of the transaction by Sage Pay is now complete, but later in the day, the final stage of the process is carried out between us and the banks without you or your site needing to do anything.

Step 9: Sage Pay sends Settlement Batch Files to confirm payments.
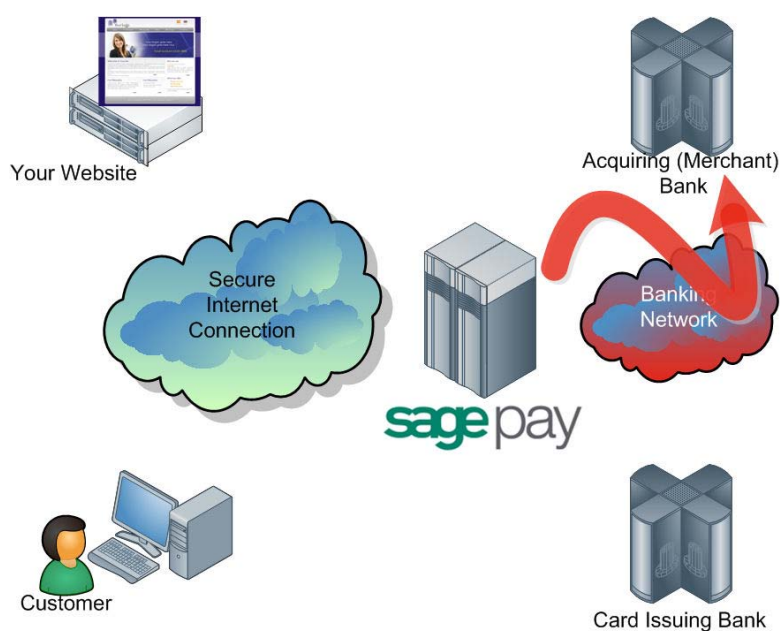
Once per day, after 2:00am, the Sage Pay system batches all authorised transactions for each acquirer and creates a bank specific settlement file.

Transactions for ALL merchants who use the same merchant acquirer are included in this file. Every transaction (excluding PayPal transactions*) that occurred from 00:00:00am until 11:59:59pm on the previous day, is included in the files.

They are uploaded directly to the acquiring banks on a private secure connection. This process requires no input from you or your site. The contents of these batches and confirmation of their delivery can be found in the *My Sage Pay* system.

If the file does not transmit correctly, the system tries a further nine times at 10-minute intervals. If all 10 attempts fail the transactions for that bank are rescheduled for inclusion in the following day's batch instead. Sage Pay monitor this process each day to ensure the files have been sent, and if not, the support department correct the problem during the day to ensure the file is sent correctly

that evening (or normally resubmit the file manually the same day to ensure funds are available to all vendors more expediently).

The acquirers send summary information back to Sage Pay to confirm receipt of the file, then later more detailed information about rejections or errors. If transactions are rejected, we correct any errors and resubmit them for you. Your bank will contact you directly if there are payment related problems with the transactions.

***Important note for PayPal transactions**: PayPal transactions are settled by immediately with PayPal. The funds from your customers' PayPal payments are deposited into your PayPal Business account immediately. You can then withdraw or transfer the funds electronically into your specified bank account. Although PayPal transactions are included in the Settlement Reports displayed within *My Sage Pay*, as PayPal transactions are not settled by Sage Pay directly with the banks, we recommend you to log into your PayPal Admin area to obtain a report of your PayPal transactions.

Integrating with Sage Pay Form

Linking your Website to Sage Pay using the Form integration method involves creating one script (or modifying the example provided in the integration kits), and two completion pages, one for successful transactions, the other for failures.

Stage 1
The Sage Pay Simulator system is the starting point for your integration. This user-friendly expert system on our test environment analyses the messages your site sends to us, reports any errors therein, and simulates all possible responses from the real Sage Pay live environment.

The Sage Pay Simulator can be configured on the following URL:
https://test.sagepay.com/simulator

Payment transactions should be sent from your scripts to the following URL:
https://test.sagepay.com/Simulator/VSPFormGateway.asp

Stage 2
Once your site is able to talk to Sage Pay Simulator and process all possible outcomes, you will be able to move over to the Sage Pay Test Server. This is an exact copy of the live site but without the banks attached and with a simulated 3D-Secure environment. Authorisations on the test server are only simulated, but the user experience is identical to Live, and a version of the *My Sage Pay* pages also runs here so you can familiarise yourself with the features available to you.

The *My Sage Pay* admin system for viewing your Test transactions is at:
https://test.sagepay.com/mysagepay

Transactions from your scripts should be sent to the Sage Pay Test Server at:
https://test.sagepay.com/gateway/service/vspform-register.vsp

Stage 3
Once you are happily processing end-to-end transactions on the test server and we can see test payments and refunds going through your account, AND you've completed the online Direct Debit signup, your account on the Live Server is activated for you to start using. You will need to redirect your scripts to send transactions to the live service, send through a Payment using your own credit card, then VOID it through the *My Sage Pay* Admin service so you don't charge yourself. If this works successfully, then you are ready to trade online.

The *My Sage Pay* screens are at:
https://live.sagepay.com/mysagepay

Transactions from your scripts should be sent to the Live Site Sage Pay Form at:
https://live.sagepay.com/gateway/service/vspform-register.vsp

Stage 1: Integrating with the Sage Pay Simulator

The Sage Pay Simulator is an expert system that emulates the Sage Pay Form system and allows you to develop your site to correctly send and process the messages exchanged between your site and ours. The Simulator will provide more detailed feedback of any errors or issues than the real Sage Pay Server, allowing you to debug and enhance your code.

Log into the Sage Pay Simulator at https://test.sagepay.com/simulator and enter your Vendor Name (as you selected on the Online Registration forms) and the password (also the same as that used on those forms. You can change it in the Simulator if you wish).

If you wish to test your integration with Sage Pay before you have obtained a Merchant Account, you can do so free of charge with the Sage Pay Simulator. To register for a Simulator account, please visit our website:

https://support.sagepay.com/apply/requestsimaccount.aspx
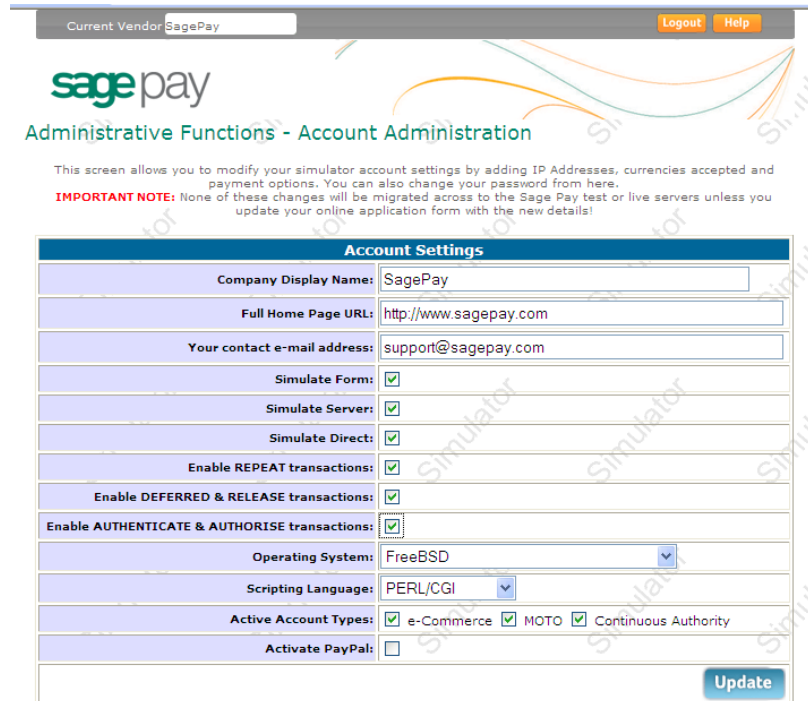
When you log in to the Sage Pay Simulator you will be presented with the main menu screen. Extensive help is provided in the Simulator (click the context sensitive Help button on each screen for more details) so this document will not cover everything in too much detail, but outlined in subsequent sections are the important steps you should take to get your site talking to the Simulator.
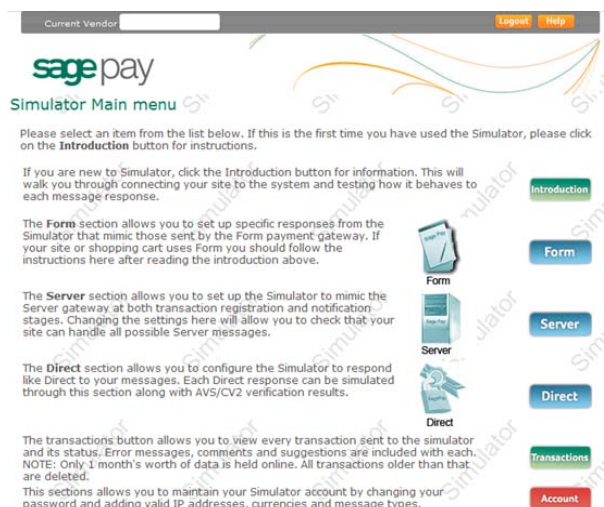
1: Sage Pay Simulator Account Set up

Click the Account button in the main menu to open the following screen:

You should ensure that:

- all company details are correct.
- all technical details about web server and platform are correct.
- the "Simulate Form" box is checked.
- all relevant payment types have been set up.
- you have at least one payment currency set up (usually GBP unless your site accepts multi-currency transactions).

Add and/or correct any entries and click the Update button to save any changes.
Back takes you back to the main menu.

2: Registering a Payment

If you don't plan to implement the protocol entirely on your own, you should install the most appropriate integration kit or worked example for your platform. These can be downloaded as part of the application process or obtained from the developer's area on the Sage Pay website http://www.sagepay.com/developers.asp.

The kits will not quite run out of the box because you have to provide some basic details about your site in the configuration files before a transaction can occur, but they will provide end-to-end examples of creating transaction registrations and handling the success and failure call-backs. Ensure you've completed all configuration in the `includes` file as detailed in the kit instructions, then locate the `OrderConfirmation` script.

This script provides a worked example of how to construct the Crypt field that Sage Pay Form needs to initiate the payment process (see Appendix A section A1 in the attached protocol).

Check that this script is sending transactions to the Sage Pay Simulator (rather than the test or live sites), then execute this page, passing it some dummy transaction data if necessary, to send a payment registration to the Simulator.

BillingSurname:	**SAGEPAYTEST** The BillingSurname is valid.
BillingFirstnames:	**SAGEPAYTEST** The BillingFirstnames is valid.
BillingAddress1:	**SAGEPAYTEST** The BillingAddress1 is valid.
BillingAddress2:	**SAGEPAYTEST** The BillingAddress2 is valid.
BillingCity:	**SAGEPAYTEST** The BillingCity is valid.
BillingPostCode:	**SAGEPAYTES** The BillingPostCode is valid.
BillingCountry:	**GB** The BillingCountry is valid.
BillingState:	No BillingState provided. Field is optional.
BillingPhone:	No BillingPhone provided. Field is optional.
DeliverySurname:	**SAGEPAYTEST** The DeliverySurname is valid.
DeliveryFirstnames:	**SAGEPAYTEST** The DeliveryFirstnames is valid.
DeliveryAddress1:	**SAGEPAYTEST** The DeliveryAddress1 is valid.
DeliveryAddress2:	**SAGEPAYTEST** The DeliveryAddress2 is valid.
DeliveryCity:	**SAGEPAYTEST** The DeliveryCity is valid.
DeliveryPostCode:	**SAGEPAYTES** The DeliveryPostCode is valid.
DeliveryCountry:	**GB** The DeliveryCountry is valid.
DeliveryState:	No DeliveryState provided. Field is optional.
DeliveryPhone:	No DeliveryPhone provided. Field is optional.

Basket:

Basket Contents (2 line(s) of detail)

Items	Quantity	Item value	Item Tax	Item Total	Line Total
IronMan	1	7.45	1.30	8.75	8.75
Delivery	1	1.50	---	1.50	1.50

The Basket is valid.

AllowGiftAid:	**0** The AllowGiftAid flag is valid.
ApplyAVSCV2:	**0** The ApplyAVSCV2 flag is valid.
Apply3DSecure:	**0** The Apply3DSecure flag is valid.

Proceed ▶

You will be presented with a screen that attempts to decode your crypt field and provides feedback on the contents.

The top section of the screen validates the hidden fields, ensuring your protocol version is correct, your transaction type is valid and your vendor name is known. If anything is incorrect, you will be informed of the error.

The crypt field is then decoded if possible. The contents are broken down for you and the sections highlighted in a traffic light manner: Greens are fine, yellows are warnings but will not stop the transaction progressing (you may have chosen, for example, not to include an optional field), but red indicates a problem you should fix.

The Proceed button only appears if there are no red errors. Clicking it takes you to the next screen.

Version Date: Friday, 14 August 2009   Version: 2.23

This page is for information only and shows you what the customer would be seeing at this stage in the real Sage Pay Form system. It explains the type of messages our system might generate and under what circumstances.

The completion page allows you to select which type of response you wish to send back to your server.

An **OK** message, indicating an authorised transaction, would redirect you back to your success page, as passed in the **SuccessURL** field.

All other message types are set to the **FailureURL**, along with the reason for failure in the **StatusDetail** field.

You can also choose the exact fraud screening results you wish to send back, to enable you to develop code that responds to these values if you wish. By default the example in the kits simply display them.

3: Examining your transactions

The Sage Pay Simulator keeps the last month's worth of simulated transactions online for you to examine at your leisure. Using the Transactions button you can view everything you've sent us to ensure the data is as you expected.

Once your site can initiate transactions AND handle the call-backs, then you've completed your basic Sage Pay Form integration and can move on to testing your site against the real Sage Pay Form, firstly on the Test Server.

Version Date: Friday, 14 August 2009   Version: 2.23

Stage 2: Testing on the Test Server

If your site works correctly against the Sage Pay Simulator then this is normally a very quick step. The Test Server is an exact copy of the Live System but without the banks attached and with a simulated 3D-Secure environment. This means you get a true user experience but without the fear of any money being taken from your cards during testing.

In order to test on the Test Server, however, you need a Test Server account to be set up for you by the Sage Pay Support team. These accounts can **only** be set up once you have completed all sections of the Online Registration forms (https://support.sagepay.com/apply/) including the Merchant Account section. Often when applying to trade online it takes a while for the Merchant Account to be assigned by your acquirer, so you may wish to ensure that you set those wheels in motion before you begin your integration with Sage Pay, to ensure things don't bottleneck at this stage.

The Support Team will set up an account for you on the Test Server under the same Sage Pay Vendor Name as your online application form and Simulator account. You will, however, be issued with different passwords for security purposes. The Support Team will let you know how to retrieve those passwords and from there how to use the *My Sage Pay* screens to look at your transactions.

To link your site to the Test Server, you need only to change your transaction registration script to send the message to the Test Server URL for Sage Pay Form rather than the Simulator. In the kits this is done simply by change the flag in the configuration scripts from SIMULATOR to TEST. If you've been developing your own scripts, then the Test Site URL for payment registration is:

https://test.sagepay.com/gateway/service/vspform-register.vsp

When your site redirects the customer you will find yourself on the real Sage Pay payment pages rather than the Simulator.
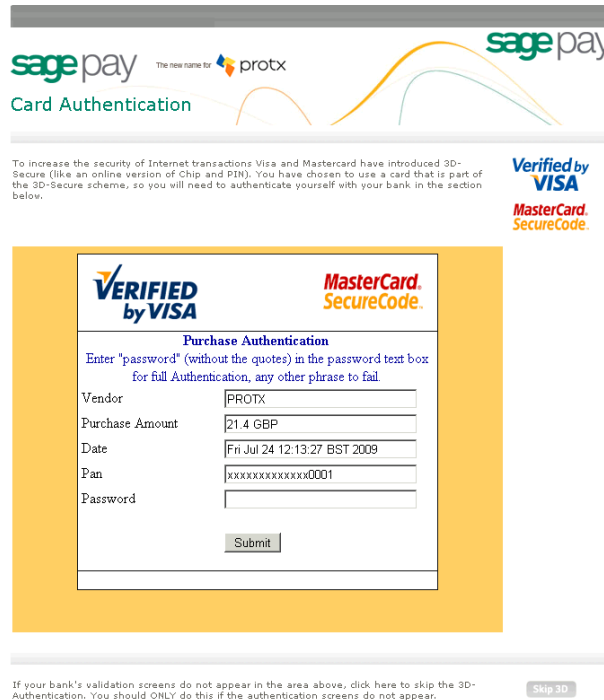
You will always receive an OK message and an Authorisation Code from the test server if you are using one of the test cards listed below. All other valid card numbers will be declined, allowing you to test your failure pages. If you do not use the correct Address, Post Code and CV2 digits, the transaction will still authorise, but you will receive NOTMATCHED messages in the AVS/CV2 checks, allowing you to test your rule-bases and fraud specific code.

Any cardholder name and start/expiry dates will be accepted for these cards so long as the dates are valid and the card not expired.

Card Type	Card Number	Issue	CV2	Address	PostCode
Visa Credit	4929 0000 0000 6		123	88	412
MasterCard Credit	5404 0000 0000 0001		123	88	412
Visa Debit / Delta	4462 0000 0000 0003		123	88	412
Solo	6334 9000 0000 0005	1	123	88	412
UK Maestro	5641 8200 0000 0005	01	123	88	412
American Express	3742 0000 0000 004		123	88	412
Visa Electron	4917 3000 0000 0008		123	88	412
JCB	3569 9900 0000 0009		123	88	412
Diner's Club	3600 0000 0000 08		123	88	412
Laser (LASER)	630499000000000044		123	88	412

If you have 3D-Secure set up on your test account, you can use the *My Sage Pay* interface to switch on the checks at this stage to test your 3D-Secure terminal URL script against a simulation of the 3D-Secure environment.

This simulation is more advanced than the Sage Pay Simulator process because it creates real 3D-secure messages. It does not talk to the Visa and MasterCard systems though, so no live authentications can occur.

At the Simulated Authentication screen, to successfully authenticate the transaction, enter "**password**" (without the quotes) into the password box. Any other phrase will fail the authentication, allowing you to test your rules and 3D-Secure response handling.

The process will then continue as per the Live Servers. Only the authorisation stage is simulated.

Once you've checked you can process an end-to-end transaction then you are almost ready to go live. Before doing so, however, you should log in to the *My Sage Pay* on the test servers to view your transactions and familiarise yourself with the interface.

The Test Server *My Sage Pay* interface

A Test Server version of the *My Sage Pay* is available to you whilst using your test account to view your transactions, refund payments, release deferred payments, void transactions etc. You should familiarise yourself with this system on the Test Server before you go live so you know how to use the system when you start to accept real, live transactions.

The Test Server *My Sage Pay* can be found at:
https://test.sagepay.com/mysagepay

When you log in to *My Sage Pay* you will be asked for a **Vendor Name**, a **User Name** and a **Password**. The first time you log in you will need to do so as your system Administrator:

- In the **Vendor Name** box, enter your Vendor Name, as selected in your Online Registration screens and used throughout the development as your unique merchant identifier.
- In the **User Name** box, enter the Vendor Name again.
- In the **Password** box, enter the *My Sage Pay* password as supplied to you by Sage Pay when your test account was set up.
- Click **Login**.

The administrator can ONLY create user accounts, unlock other accounts and change account parameters. You cannot, whilst logged in as administrator, view your transactions or take payments through the online terminal.

To use those functions, and to protect the administrator account, you need to create new users for yourself and others. Click the Add button to add a new user.

Enter a username for yourself and a password you'll remember, then ensure all the check boxes are enabled for your account. Click the Add button and your new User account will appear in the list.

Now click the Logout button and click to Log back in, this time entering:

- Your Vendor name in the **Vendor Name** box.
- The User Name of the account you just created in the **User Name** box.
- The password for the 'user' account you just created in the **Password** box.

…and click **Login**.

You are now logged in using your own account and can view your test transactions and use all additional functions. You need only log in as Administrator again if you wish to create additional users, or if you lock yourself out of your own account, you can use the Administrator account to unlock yourself.

If you happen to lock out the Administrator account, you will need to contact Sage Pay to unlock it for you: send an email to unlock@sagepay.com stating the Vendor Name and Merchant Number of the account. If you need reminding of your unique account passwords, send an email to the above and request a password retrieval link, stating the Vendor Name and Merchant Number of the account.

Detailed context sensitive help is available on every *My Sage Pay* page by clicking the Help button, so a description of the functions will not be presented here. Play with the system until you are comfortable with it though; you cannot inadvertently charge anyone or damage anything whilst on the test server.

Additional Transaction Types

Sage Pay supports a number of additional methods of registering a transaction and completing the payment.

DEFERRED transactions.

By default a PAYMENT transaction type is used in your scripts to gain an authorisation from the bank, then settle that transaction early the following morning, committing the funds to be taken from your customer's card.

In some cases you may not wish to take the funds from the card immediately, but merely place a "shadow" on the customer's card to ensure they cannot subsequently spend those funds elsewhere, and then only take the money when you are ready to ship the goods. This type of transaction is called a **DEFERRED** transaction and is registered in exactly the same way as a normal PAYMENT. You just need to change your script to send a TxType of DEFERRED when you register the transaction (protocol A1) instead of PAYMENT.

DEFERRED transactions are NOT sent to the bank for completion the following morning. In fact, they are not sent at all until you **RELEASE** them by logging into the *My Sage Pay* interface, finding the transaction and clicking the Release button.

You can release ONLY ONCE and ONLY for an amount up to and including the amount of the original DEFERRED transaction.

If you are unable to fulfil the order, you can also **ABORT** deferred transactions in a similar manner and the customer will never be charged.

DEFERRED transactions work well in situations where it is only a matter of days between the customer ordering and you being ready to ship. Ideally all DEFERRED transaction should be released within 6 days (according to card scheme rules). After that the shadow may disappear from the card before you settle the transaction, and you will have no guarantee that you'll receive the funds if the customer has spent all available funds in the mean time. If you regularly require longer than 6 days to fulfil orders, you should consider using AUTHENTICATE and AUTHORISE instead of DEFERRED payments (see below)

DEFERRED transactions remain available for RELEASE for up to 30 days. After that time they are automatically ABORTed by the Sage Pay systems.

Additional notes for using Deferred/Release with PayPal transactions

Unlike a normal Sage Pay DEFERRED transaction, no shadow is placed on the customer's card for a PAYPAL DEFERRED transaction. An order is simply registered with the PayPal account and a successful authorisation for a DEFERRED transaction only confirms the availability of funds and does not place any funds on hold.

When you RELEASE a DEFERRED PayPal transaction PayPal applies best efforts to capture funds at that time, but there is a possibility that funds will not be available.

We recommend that you do not ship goods until after obtaining a successful release.

REPEAT payments

If you have already successfully authorised a customer's card using as PAYMENT, a released DEFERRED or an AUTHORISE (see below) you can charge an additional amount to that card using the **REPEAT** transaction type, without the need to store the card details yourself.

If you wish to regularly REPEAT payments, for example for monthly subscriptions, you should ensure you have a "Continuous Authority" merchant number from your bank (please contact your acquiring bank for further details), but ad-hoc REPEATs do not require a Continuous Authority merchant number. REPEAT payments cannot be 3D-Secured, or have CV2 checks performed on them (unless you supply those values again. Sage Pay are not allowed to store CV2 numbers) so you are better to make use of Authenticate and Authorise if you need to vary the transaction amount on a regular basis.

You can only REPEAT a PayPal transaction if the initial transaction was set up as a PayPal Reference transaction (with BillingAgreement set to 1. See the Appendix for details).

AUTHENTICATE and AUTHORISE

The AUTHENTICATE and AUTHORISE methods are specifically for use by merchants who are either (i) unable to fulfil the majority of orders in less than 6 days (or sometimes need to fulfil them after 30 days) or (ii) do not know the exact amount of the transaction at the time the order is placed (for example, items shipped priced by weight, or items affected by foreign exchange rates).

Unlike normal PAYMENT or DEFERRED transactions, AUTHENTICATE transactions do not obtain an authorisation at the time the order is placed. Instead the card and card holder are validated using the 3D-Secure mechanism provided by the card-schemes and card issuing banks, with a view to later authorisation.

Your site will register your transaction with a TxType of **AUTHENTICATE**, and redirect the customer to Sage Pay Form to enter their card details. Sage Pay Form will contact the 3D-Secure directories to check if the card is part of the scheme. If it is not, then the card details are simply held safely at Sage Pay and your SuccessURL is sent a Status of **REGISTERED** (This also happens if you do not have 3D-Secure active on your account or have used the Apply3DSecure flag to turn it off).

If, however, the card *is* part of the 3D-Secure scheme, the customer is redirected to their card issuing bank for authentication (just like a normal 3D-Secure payment, see steps 5 and 6 in the Payment Process above). Here they will authenticate themselves and be returned to Sage Pay Form.

If they have not passed authentication, your rule base is consulted to check if they can proceed for authorisation anyway. If not, your FailureURL is sent a Status of **REJECTED**. If they failed authentication but can proceed, your SuccessURL is sent a **REGISTERED** status. If the user passed authentication with their bank and a CAVV/UCAF value is returned, your SuccessURL is sent a Status of **AUTENTICATED** and a **CAVV** value for you to store if you wish.

In all cases, the customer's card is never authorised. There are no shadows placed on their account and your acquiring bank is not contacted. The customer's card details and their associated authentication status are simply held at Sage Pay for up

to 90 days (a limit set by the card schemes, 30 days for International Maestro cards) awaiting an **AUTHORISE** or **CANCEL** request from your site.

To charge the customer when you are ready to fulfil the order, you'll need to log into *My Sage Pay*, select the Authenticated/Registered transaction and click **Authorise**. You can Authorise any amount up to 115% of the value of the Authentication and use any number of Authorise requests against an original Authentication so long as the total value of those authorisations does not exceed the 115% limit, and the requests are inside the 90 days limit. This is the stage at which your acquiring bank is contacted for an auth code. AVS/CV2 checks are performed at this stage and rules applied as normal. This allows you great flexibility for partial shipments or variable purchase values. If the AUTHENTICATE transaction was AUTHENTICATED (as opposed to simply REGISTERED) all authorisations will be fully 3D-Secured, so will still receive the fraud liability shift.

When you have completed all your Authorisations, or if you do not wish to take any, you can select **CANCEL** from the *My Sage Pay* screens to archive away the Authentication and prevent any further Authorisations being made against the card. This happens automatically after 90 days.

NB: For PayPal transactions, you can use the Authenticate and Authorise Payment Type but the transaction will only ever be **REGISTERED** (because PayPal do not support 3D-Secure). Similarly to Releasing a Deferred transaction, we recommend you to Authorise the transaction via the *My Sage Pay* area when you are ready to ship the goods and take the funds.

REFUNDs and VOIDs

Once a PAYMENT, AUTHORISE or REPEAT transaction has been authorised, or a DEFERRED transaction has been RELEASEd, it will be settled with the acquiring bank early the next morning and the funds will be moved from the customer's card account, across to your merchant account. The bank will charge you for this process, the exact amount depending on the type of card and the details of your merchant agreement.

If you wish to cancel that payment before it is settled with the bank the following morning, you can **VOID** transactions through the *My Sage Pay* interface. VOIDed transactions can NEVER be reactivated though, so use this functionality carefully.

Once a transaction has been settled, however, you can no longer VOID it. If you wish to return funds to the customer you need to **REFUND** the transaction, again through *My Sage Pay*.

You can REFUND any amount up to the value of the original transaction. You can even send multiple refunds for the same transaction so long as the total value of those refunds does not exceed the value of the original transaction.

You **cannot VOID** a PayPal transaction, but you are able to **REFUND** a PayPal transaction.

Stage 3: Going Live

Once Sage Pay receives your application your account will be created and details will be sent to the bank for confirmation. The bank will be expected to confirm your merchant details within 3 to 5 working days. Once both the Direct Debit (filled out during application) and the confirmation of your merchant details reach Sage Pay, your account will become Live automatically.

This does not mean you will immediately be able to use your Live account

You must ensure you have completed testing of your account before you are granted access to your Live account. Details can be found below:

http://www.sagepay.com/developers/integration_manual/processes_go_live.asp

NB – Without confirmation from the bank and without Direct Debit submission, Sage Pay will not be able to set your account Live. You will only be charged by Sage Pay when your account has valid Direct Debit and confirmation of your merchant details from the bank.

Once your Live account is active, you should point your website transaction registration scripts to the following URL:

https://live.sagepay.com/gateway/service/vspform-register.vsp

You should then run an end-to-end transaction through your site, ordering something relatively inexpensive from your site and paying using your own valid credit or debit card. If you receive an authorisation code, then everything is working correctly.

You should then log into the Live Server *My Sage Pay* screens at https://live.sagepay.com/mysagepay and in a similar manner to the test server, first log in as the Administrator, then create a Live System User account for yourself, log in as that user, locate your test transaction and **VOID** it, so you are not charged for the transaction. At this stage the process is complete.

It is worth noting here that none of the users you set up on the *My Sage Pay* system on the Test Server are migrated across to Live. This is because many companies use third party web designers to help design the site and create users for them during test that they would not necessarily like them to have in a live environment. You will need to recreate any valid users on the Live system when you first log in.

Congratulations, you are live with Sage Pay Form

Well done. Hopefully the process of getting here was as painless and hassle free as possible. You'll be pleased to know that now you are live we don't cut the strings and run away. You should contact us with any transaction queries that arise or for any help you need with the *My Sage Pay* system.

Here are the best ways to reach us and the best people to reach:

- If you require any information on additional services, e-mail sales@sagepay.com

- If you have a query regarding a Sage Pay invoice, e-mail finance@sagepay.com

- If you have a question about a transaction, have issues with your settlement files, are having problems with your payment pages or *My Sage Pay* screens, or have a general question about online payments or fraud, e-mail support@sagepay.com with your Sage Pay Vendor Name included in the mail.

- If you have any suggestions for future enhancements to the system, or additional functionality you'd like to see added, please e-mail feedback@sagepay.com with your comments. We do take all comments on board when designing upgrades, although we may not be able to answer every mail we get.

- You can call us as well on **0845-111-4455**, for any type of enquiry.
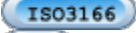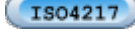
We will also keep you updated about major system changes, new reports and other enhancements via the Updates section in *My Sage Pay*, plus your e-mail address will be added to our group mail list used to alert you to upgrades and other pending events.

You can also always check our system availability and current issues on the Sage Pay Monitor page at http://www.sagepay.com/system_monitor.asp

Thanks again for choosing Sage Pay, and we wish you every success in your e-commerce venture.

Appendix A - The Sage Pay Form 2.23 Protocol

This section details the Sage Pay Form Protocol transaction registration POST, and the contents of the Crypt fields passed back and forth between your website and ours. The format and size of each field is given, along with accepted values and characters. The legend below explains the symbols:

Symbol	Description	Symbol	Description
á	Accented Characters	CR LF	New line (Carriage Return and Line Feed)
&	Ampersand character	0-9	Numbers
@	At sign	+	Plus sign
:	Colon	()	Parentheses
,	Comma	;	Semi-colon
{}	Curly Brackets	'	Apostrophe (single quote)
.	Full Stop/Period	/\	Backslash and Forward Slash
-	Hyphen		Space
Aa	Letters (A-Z and a-z)	_	Underscore
ISO3166	ISO 3166-1 2-letter country codes	BASE64	Valid Base64 characters (A-Z,a-z,0-9,+ and /)
US	Valid 2-letter US States	ISO4217	ISO 4217 3-letter Currency codes
RFC1738	RFC 1738 compliant HTTP(S) URL All non-compliant characters, including spaces, should be URL Encoded		
<HTML>	Valid HTML with no active content. Script will be filtered. Includes all valid letters, numbers, punctuation and accented characters.		
RFC532n	RFC 5321/5322 (see also RFC 3696) compliant e-mail Addresses.		

A1: Transaction Registration

The final confirmation page on your website should contain an HTML FORM with the Action set to the Sage Pay Form submission URL and the following 4 hidden fields as part of that Form.

Form Fields

Name	Format	Values	Comments
VPSProtocol	Alphanumeric. Fixed 4 characters.	**2.23** ONLY	Default or incorrect value is taken to be 2.23
TxType	Alphanumeric Max 15 characters.	**PAYMENT**, **DEFERRED** or **AUTHENTICATE** ONLY	TxType should be in capital Letters.
Vendor	Alphanumeric Max 15 characters.	Vendor Login Name Aa 0-9 - _	Used to authenticate your site. This should contain the Sage Pay Vendor Name supplied by Sage Pay when your account was created.
Crypt	Alphanumeric Max 16k characters	All other transaction information, encrypted then encoded. See below. BASE64 @	Your site builds the Crypt field in real time for each order. The contents of the field are described below.

The Crypt Field

The Crypt field should contain all the other transaction information (see the next section) in plain text as Name=Value fields separated by '&' characters. This string should then be encrypted using the AES/CBC/PCKS#5 algorithm and the pre-registered Encryption password, then subsequently Base64 encoded to allow safe transport in an HTML form.

The functions to perform these steps (EncryptandEncode) are included in the kits and can be used in your own script pages.

Crypt Field Contents (continued overleaf)

Name	Format	Values	Comments
VendorTxCode	Alphanumeric Max 40 characters	Vendor Transaction Code Aa 0-9 {} . - _	This should be your own reference code to the transaction. Your site should provide a completely unique VendorTxCode for each transaction.
Amount	Numeric. 0.01 to 100,000.00	Amount for the Transaction containing minor digits formatted to 2 decimal places where appropriate. 0-9 , .	Must be positive and numeric, and may include a decimal place where appropriate. Minor digits should be formatted to two decimal places. e.g. 5.10, or 3.29. Values such as 3.235 will be rejected. Minimum for no minor unit currencies like JPY is 1.
Currency	Alphanumeric 3 characters	Three-letter currency code to ISO 4217 Examples: **GBP**, **EUR** and **USD** ISO4217	The currency must be supported by one of your Sage Pay merchant accounts or the transaction will be rejected.
Description	Alphanumeric Max 100 characters	Free text description of goods or services being purchased <HTML>	The description of goods purchased is displayed on the Sage Pay Form payment page as the customer enters their card details.
SuccessURL	Alphanumeric Max 2000 characters	Full qualified URL (including http:// or https:// header). RFC1738	The URL of the page/script to which the user is redirected if the transaction is successful. You may attach parameters if you wish. Sage Pay Form will also send an encrypted field containing important information appended to this URL (see below).
FailureURL	Alphanumeric Max 2000 characters	Full qualified URL (including http:// or https:// header). RFC1738	The URL of the page/script to which the user is redirected if the transaction is not successful, aborted or an error occurs. You may attach parameters if you wish. Sage Pay Form will also send an encrypted field containing important information appended to this URL (see below).
Optional: CustomerName	Alphanumeric Max 100 characters	The customer's name. Aa á /\ & . - '	If provided the customer's name will be included in the confirmation e-mails and stored in *My Sage Pay*.
Optional: CustomerEMail	Alphanumeric Max 255 characters	The customer's e-mail address. NOTE: If you wish to use multiple email addresses, you should add them using the **:** (colon) character as a separator. e.g. me@mail1.com:me@mail2.com RFC532n	If provided, the customer will be e-mailed on completion of a successful transaction (but not an unsuccessful one).

Optional: VendorEMail	Alphanumeric Max 255 characters	An e-mail address on which you can be contacted when a transaction completes. NOTE: If you wish to use multiple email addresses, you should add them using the **:** (colon) character as a separator. e.g. me@mail1.com:me@mail2.com [RFC532n]	If provided, an e-mail will be sent to this address when each transaction completes (successfully or otherwise).
Optional: SendEMail	Flag	**0 =** Do not send either customer or vendor e-mails **1** = Send customer and vendor e-mails if addresses are provided (DEFAULT) **2** = Send vendor e-mail but NOT the customer e-mail	If you do not supply this field, 1 is assumed and e-mails are sent if addresses are provided.
Optional: eMailMessage	Alphanumeric Max 7500 characters	A message to the customer which is inserted into the successful transaction e-mails only. [<HTML>]	If provided this message is included toward the top of the customer confirmation e-mails.
BillingSurname	Alphanumeric Max 20 characters	Customer's surname [Aa] [á] [/\] [&] [.] [−] [']	In Protocol 2.23, unlike previous protocols, the Billingxxxxx columns are compulsory.
BillingFirstnames	Alphanumeric Max 20 characters	Customer's first names [Aa] [á] [/\] [&] [.] [−] [']	
BillingAddress1	Alphanumeric Max 100 characters	First line of billing address [Aa] [á] [0−9] [+] ['] [/\] [&] [:] [,] [.] [−] [CR LF] [()]	
Optional: BillingAddress2	Alphanumeric Max 100 characters	Second line of billing address [Aa] [á] [0−9] [+] ['] [/\] [&] [:] [,] [.] [−] [CR LF] [()]	
BillingCity	Alphanumeric Max 40 characters	City component of the address [Aa] [á] [0−9] [+] ['] [/\] [&] [:] [,] [.] [−] [CR LF] [()]	
BillingPostCode	Alphanumeric Max 10 characters	The Post/Zip code of the Card Holder's Billing [Aa] [0−9] [−]	
BillingCountry	Alphanumeric Max 2 characters	ISO 3166-1 country code of the cardholder's billing address [ISO3166]	

Optional*: BillingState	Alphanumeric Max 2 characters	State code for US customers only* [US]	
Optional: BillingPhone	Alphanumeric Max 20 characters	Phone number at billing address [0-9] [-] [Aa] [+] [ ] [()]	
DeliverySurname	Alphanumeric Max 20 characters	Customer's surname [Aa] [á] [ ] [/\] [&] [.] [-] [′]	In Protocol 2.23, unlike previous protocols, the Deliveryxxxx columns are compulsory.
DeliveryFirstnames	Alphanumeric Max 20 characters	Customer's first names [Aa] [á] [ ] [/\] [&] [.] [-] [′]	
DeliveryAddress1	Alphanumeric Max 100 characters	First line of delivery address [Aa] [á] [0-9] [ ] [+] [′] [/\] [&] [:] [,] [.] [-] [CR LF] [()]	
Optional: DeliveryAddress2	Alphanumeric Max 100 characters	Second line of delivery address [Aa] [á] [0-9] [ ] [+] [′] [/\] [&] [:] [,] [.] [-] [CR LF] [()]	
DeliveryCity	Alphanumeric Max 40 characters	City component of the address [Aa] [á] [0-9] [ ] [+] [′] [/\] [&] [:] [,] [.] [-] [CR LF] [()]	
DeliveryPostCode	Alphanumeric Max 10 characters	The Post/Zip code of the Card Holder's delivery address [Aa] [0-9] [ ] [-]	
DeliveryCountry	Alphanumeric Max 2 characters	ISO 3166-1 country code of the cardholder's delivery address [ISO3166]	
Optional*: DeliveryState	Alphanumeric Max 2 characters	State code for US customers only* [US]	
Optional: DeliveryPhone	Alphanumeric Max 20 characters	Phone number at delivery address [0-9] [-] [Aa] [+] [ ] [()]	
Optional: Basket	Alphanumeric Max 7500 characters	See the next page for the Format of the Basket field [<HTML>]	You can use this field to supply details of the customer's order. This information will be displayed to you in *My Sage Pay*.
Optional: AllowGiftAid	Flag	**0** = No Gift Aid Box displayed (default) **1** = Display Gift Aid Box on payment screen.	This flag allows the gift aid acceptance box to appear for this transaction on the payment page. This only appears if your vendor account is Gift Aid enabled.

Optional: ApplyAVSCV2	Flag	**0** = If AVS/CV2 enabled then check them. If rules apply, use rules. (default) **1** = Force AVS/CV2 checks even if not enabled for the account. If rules apply, use rules. **2** = Force NO AVS/CV2 checks even if enabled on account. **3** = Force AVS/CV2 checks even if not enabled for the account but DON'T apply any rules.	Using this flag you can fine tune the AVS/CV2 checks and rule set you've defined at a transaction level. This is useful in circumstances where direct and trusted customer contact has been established and you wish to override the default security checks. **This field is ignored for PAYPAL transactions**
Optional: Apply3DSecure	Flag	**0** = If 3D-Secure checks are possible and rules allow, perform the checks and apply the authorisation rules. (default) **1** = Force 3D-Secure checks for this transaction if possible and apply rules for authorisation. **2** = Do not perform 3D-Secure checks for this transaction and always authorise. **3** = Force 3D-Secure checks for this transaction if possible but ALWAYS obtain an auth code, irrespective of rule base.	Using this flag you can fine tune the 3D Secure checks and rule set you've defined at a transaction level. This is useful in circumstances where direct and trusted customer contact has been established and you wish to override the default security checks. **This field is ignored for PAYPAL transactions**
Optional: BillingAgreement	Flag	**0 =** This is a normal PayPal transaction, not the first in a series of payments (default) **1** = This is the first in a series of PayPal payments. Subsequent payments can be taken using REPEAT.	**This field must be set for PAYPAL REFERENCE transactions All non-PayPal transactions can be repeated without this flag.** If you wish to register this transaction as the first in a series of regular payments, this field should be set to 1. If you do not have a PayPal account set up for use via Sage Pay, then this field is not necessary and should be omitted or set to 0.

Basket Contents

The shopping basket contents can be passed in a single, colon-delimited field, in the following format:

```
Number of lines of detail in the basket field:
Item 1 Description:
Quantity of item 1:
Unit cost item 1 without tax:
Tax applied to item 1:
Cost of Item 1 including tax:
Total cost of item 1 (Quantity x cost including tax):
Item 2 Description:
Quantity of item 2:
....
Cost of Item n including tax:
Total cost of item n
```

IMPORTANT NOTES:

o   The line breaks above are included for readability only. No line breaks are needed; the only separators should be the colons.

o   The first value "The number of lines of detail in the basket" is **NOT** the total number of items ordered, but the total number of rows of basket information. In the example below there are 6 items ordered, (1 DVD player and 5 DVDs) but the number of lines of detail is 4 (the DVD player, two lines of DVDs and one line for delivery).

So, for example, the following shopping cart…

Items	Quantity	Item value	Item Tax	Item Total	Line Total
Pioneer NSDV99 DVD-Surround Sound System	1	424.68	74.32	499.00	499.00
Donnie Darko Director's Cut	3	11.91	2.08	13.99	41.97
Finding Nemo	2	11.05	1.94	12.99	25.98
Delivery	---	---	---	---	4.99

Would be represented thus:

```
4:Pioneer NSDV99 DVD-Surround Sound System:1:424.68:74.32:499.00: 499.00:Donnie Darko Director's
Cut:3:11.91:2.08:13.99:41.97: Finding Nemo:2:11.05:1.94:12.99:25.98: Delivery:---:---:---:
---:4.99
```

If you wish to leave a field empty, you must still include the colon. e.g.

```
DVD Player:1:199.99:::199.99
```

A2: Transaction Completion

For Sage Pay Form transactions, Sage Pay cannot guarantee to return the customer to your website. If the customer closes their browser mid-way through a transaction, or if something goes wrong at any redirect stages, it will be up to you to check the status of the transactions on the *My Sage Pay* reporting screens.

In normal circumstances, however, where the customer does not close their browser and there are no redirection problems, Sage Pay Form will return them to your site, either to the **SuccessURL** (in the event the transaction was successful), or the **FailureURL** (in all other circumstances).

The system will append to the SuccessURL or FailureURL a field called CRYPT, in the manner:

[ResponseURL]?crypt=[encrypted_information]

or if the URL already has your own fields attached, it will be appended thus:

[ResponseURL]?vendor1=test&vendor2=test2&crypt=[encrypted_information]

The SuccessURL and FailureURL fields should point to scripts on your server that extract the information in the crypt field and use it to update your database (if you have one) and/or format an appropriate response page for the customer. This is not compulsory, however, and you may choose to simply direct customers to a static HTML page that ignores the contents of the crypt field. In such cases, you will need to manually check the *My Sage Pay* report pages to determine if a transaction succeeded or failed. In fact, we recommend you always check the *My Sage Pay* pages before sending any goods just to confirm the status of each transaction.

The Crypt field contains the plain text shown overleaf as Name=Value fields separated by '&' characters, subsequently encrypted using the AES/CBC/PKCS#5 cipher and your pre-registered password, then Base64 encoded. This is exactly the same process that your scripts performed at the transaction registration stage. To read the contents, you must Base64 decode the field, Decrypt it with your Encryption password, then split the contents out into manageable fields. Routines to perform this (DecodeandDecrypt and getToken) are included in the kits and can be used in your own script pages.

Response Crypt Field Contents (continued overleaf)

Name	Format	Values	Comments
Status	Alphanumeric Max 20 characters	**OK** – Transaction completed successfully with authorisation. **NOTAUTHED** – The Sage Pay system could not authorise the transaction because the details provided by the Customer were incorrect, or insufficient funds were available. **MALFORMED** – Input message was missing fields or badly formatted – normally will only occur during development and vendor integration. **INVALID** – Transaction was not registered because although the POST format was valid, some information supplied was invalid. e.g. incorrect vendor name or currency. **ABORT** – The Transaction could not be completed because the user clicked the CANCEL button on the payment pages, or went inactive for 15 minutes or longer. **REJECTED** – The Sage Pay System rejected the transaction because of the fraud screening rules you have set on your account. **AUTHENTICATED** – The 3D-Secure checks were performed successfully and the card details secured at Sage Pay. **REGISTERED** – 3D-Secure checks failed or were not performed, but the card details are still secured at Sage Pay. **ERROR** – A problem occurred at Sage Pay which prevented transaction completion.	In the case of **NOTAUTHED**, the Transaction **has** completed through the Sage Pay System, but it has not been authorised by the bank. A status of **REJECTED** means the bank may have authorised the transaction but your own rule bases for AVS/CV2 or 3D-Secure caused the transaction to be rejected. In the cases of **ABORT, MALFORMED, INVALID** and **ERROR** (see below) the Transaction **has not** completed through Sage Pay and can be retried. **AUTHENTICATED** and **REGISTERED** statuses are only returned if the TxType is **AUTHENTICATE**. Please notify Sage Pay if a Status report of **ERROR** is seen, together with your VendorTxCode and the StatusDetail text.
StatusDetail	Alphanumeric Max 255 characters	Human-readable text providing extra detail for the Status message	You should always check this value if the Status is not **OK**.
VendorTxCode	Alphanumeric Max 40 characters	Your unique Vendor Transaction Code	Same as sent by your servers in Step A1.
VPSTxId	Alphanumeric 38 characters	The Sage Pay ID to uniquely identify the Transaction on our system.	Only present if Status not **INVALID**, **MALFORMED** or **ERROR**
TxAuthNo	Long Integer	Sage Pay unique Authorisation Code for a successfully authorised transaction.	Only present if Status is **OK**.

Amount	Numeric	The total value of the transaction.	Should match that sent in A1. Included to allow non-database driven users to react to the total order value.
AVSCV2	Alphanumeric Max 50 characters	Response from AVS and CV2 checks. Will be one of the following: **ALL MATCH, SECURITY CODE MATCH ONLY, ADDRESS MATCH ONLY, NO DATA MATCHES** or **DATA NOT CHECKED.**	Provided for Vendor info and backward compatibility with the banks. Rules set up at the Sage Pay server will accept or reject the transaction based on these values. More detailed results are split out in the next three fields. Not present if the Status is **AUTHENTICATED** or **REGISTERED.**
AddressResult	Alphanumeric Max 20 characters	**NOTPROVIDED, NOTCHECKED, MATCHED, NOTMATCHED**	The specific result of the checks on the cardholder's address numeric from the AVS/CV2 checks. Not present if the Status is **AUTHENTICATED** or **REGISTERED**
PostCodeResult	Alphanumeric Max 20 characters	**NOTPROVIDED, NOTCHECKED, MATCHED, NOTMATCHED**	The specific result of the checks on the cardholder's Post Code from the AVS/CV2 checks. Not present if the Status is **AUTHENTICATED** or **REGISTERED**
CV2Result	Alphanumeric Max 20 characters	**NOTPROVIDED, NOTCHECKED, MATCHED, NOTMATCHED**	The specific result of the checks on the cardholder's CV2 code from the AVS/CV2 checks. Not present if the Status is **AUTHENTICATED** or **REGISTERED**
GiftAid	Flag	**0** = The Gift Aid box was not checked this transaction. **1** = The user checked the Gift Aid box on the payment page	This field is always present even if GiftAid is not active on your account.
3DSecureStatus	Alphanumeric Max 50 characters	**OK** - 3D Secure checks carried out and user authenticated correctly. **NOTCHECKED** – 3D-Secure checks were not performed. **NOTAVAILABLE** – The card used was either not part of the 3D Secure Scheme, or the authorisation was not possible. **NOTAUTHED** – 3D-Secure authentication checked, but the user failed the authentication. **INCOMPLETE** – 3D-Secure authentication was unable to complete. No authentication occurred. **ERROR** - Authentication could not be attempted due to data errors or service unavailability in one of the parties involved in the check.	This field details the results of the 3D-Secure checks (where appropriate) **NOTCHECKED** indicates that 3D-Secure was either switched off at an account level, or disabled at transaction registration with a setting like Apply3DSecure=2
CAVV	Alphanumeric Max 32 characters	The encoded result code from the 3D-Secure checks (CAVV or UCAF).	Only present if the **3DSecureStatus** field is **OK**

AddressStatus	Alphanumeric Max 20 characters	Either **NONE**, **CONFIRMED** or **UNCONFIRMED**	PayPal Transactions Only. If AddressStatus is confirmed and PayerStatus is verified, the transaction may be eligible for PayPal Seller Protection. To learn more about PayPal Seller Protection, please contact PayPal directly or visit: https://www.paypal.com/uk/cgi-bin/webscr?cmd=p/gen/ua/policy_spp-outside#spp-policy for further information.
PayerStatus	Alphanumeric Max 20 characters	Either **VERIFIED** or **UNVERIFIED**	
CardType	Alphanumeric Max 15 characters	**VISA**, **MC**, **DELTA**, **SOLO**, **MAESTRO**, **UKE**, **AMEX**, **DC**, **JCB**, **LASER**, **PAYPAL**	MC is MasterCard, UKE is Visa Electron. MAESTRO is both UK and International Maestro. AMEX and DC (DINERS) can only be accepted if you have additional merchant accounts with those acquirers.
Last4Digits	Numeric Max 4 characters	The last 4 digits of the card number used in this transaction. PayPal transactions have 0000	This field is supplied to allow merchants using wallet systems to identify the card to their customers

A3: Sage Pay Form Full URL Summary

The table below shows the complete web addresses to which you send the messages detailed above.

Transaction Registration (PAYMENT, DEFERRED, AUTHENTICATE)	
Sage Pay Simulator:	https://test.sagepay.com/Simulator/VSPFormGateway.asp
TEST System:	https://test.sagepay.com/gateway/service/vspform-register.vsp
Live System:	https://live.sagepay.com/gateway/service/vspform-register.vsp

Please ensure that your firewalls allow outbound and inbound Port 443 (HTTPS only!) access in order to communicate with our servers (on Simulator/Test/Live).