Direct Protocol and Integration Guideline

Document Index

Welcome to the Sage Pay Direct integration method

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

Direct is designed to enable you to take card details on your own secure servers and pass them across to us for authorisation and secure storage in a server-to-server session that does not involve redirecting the customer to the Sage Pay pages. This enables you to white-label the payment process. Your customer never leaves your site (unless you are using PayPal or the 3D-Secure authentication processes) and they do not necessarily know that Sage Pay is authorising the transaction on your behalf (although in practice many merchants choose to tell their customers in case they have concerns about card number security).

To use the Direct method you will need a 128-bit SSL certificate to secure your payment pages. These can be obtained from a number of sources, including VeriSign. You will also need to be able to make HTTPS POSTs from scripts on your server (using something like OpenSSL on Linux platforms, or the WinHTTP object in Win32). If you are hosting with a third party company we recommend you talk to them about these requirements before committing to use Direct. If you cannot install a certificate for your payment pages, we would recommend using the Sage Pay Server integration instead. If you cannot perform HTTPS POSTs from your scripts, we would recommend the Sage Pay Form integration.

If you wish to support Verified by Visa and MasterCard SecureCode (the cardholder authentication systems collectively known as 3D-Secure), Direct provides a wrapper for these systems, removing the need for you to purchase and support your own Merchant Plug-In. All the messages will be created for you, and you'll simply need to redirect your customer to their issuing bank, and then send on the results of their 3D-Authentication back to Sage Pay to complete the payment process. Just like non 3D-Secure Direct transactions, the customer is never directed to Sage Pay. They leave your site to authenticate with their bank and then return to your site when they have finished.

This document explains how your Web servers communicate with Sage Pay using the Direct method, goes on to explain how to integrate with our testing and live environments. It also contains the complete Payment Protocol in the Appendix.

NOTE: Since card details will be collected via your site, you will be obliged to comply with the Payment Card Industry Data Security Standard (PCI-DSS). We have been working with our own data security partner, Trustwave, to set up a program for Sage Pay customers to make PCI DSS compliance easy and cost effective. For further information please visit the link below:

www.sagepay.com/pci-dss-compliance

Overview of how Direct integrated payments work

Direct payment requests are very simple. The interaction with your customer is entirely yours. The customer will select items or services to purchase and fill up a shopping basket. When they are ready to pay, you will first collect their name, billing and delivery address, contact details (telephone number, e-mail address and so forth) and perhaps allow them to sign up for quicker purchases in future. You will total the contents of the basket and summarise its contents for them before asking them to continue.

Your scripts should then store everything about the transaction and customer in your database for future reference. You will not need to store any card details because Sage Pay will hold those securely for you.

You will then present your customers with a payment page, secured with your 128-bit SSL certificate. This page will ask the customer for:

- The Cardholder Name as it appears on the card
- The Card Type (Visa, MasterCard, American Express etc.)
- The full Card Number without spaces or other separators
- The Start Date (if printed on the card)
- The Expiry Date
- The Issue Number (for some Maestro cards only)
- The Card Verification Value (called CVV or CV2 value. The extra three digits on the signature strip for most cards, or the 4 numbers printed on the front of an American Express card).
- The Card Holder’s Billing Address, including the Post Code (if you have not already asked for it and stored it in your database).

This page is submitted to a script on your server that retrieves and pre-validates those values (checking all fields are present, expiry dates are not in the past, the card number field only contains numbers etc.) before constructing an HTTPS POST containing your own unique reference to the transaction, the VendorTxCode (which should be stored alongside the order details in your database) and the correctly formatted data from your form. This HTTPS POST is sent to the Sage Pay gateway.

Non 3D-Secured transactions

Sage Pay validates the data sent to us, checking that it has come from a valid source and that all required information is present, before creating a transaction in our database to securely hold all the data passed to us, contacting the bank for authorisation and replying to you, in real-time, in the response part of the same HTTPS POST. In practice this takes about 2-3 seconds to complete.

The same script on your server that initiated the POST simply reads the Response from that POST to determine whether the transaction was authorised or not. It then updates your database with transaction reference values and the authorisation code (where appropriate) before displaying either a completion page to your customer, or an error page explaining why the payment was not accepted.

Your own database will contain all the necessary information about the transaction, the basket contents and the customer, but you will NOT need to store the card details because the transaction IDs passed to you by the Direct system will enable you to perform all other actions against that card (refunds, additional payments, cancellations and so on). This allows you to be certain that even if your server is compromised, no card details can be gleaned from your database.

The following sections explain the integration process in more detail. The Direct Payment protocol is attached in the appendix, providing a detailed breakdown of the contents of the HTTPS message sent between your servers and ours during a payment.

A companion document, “Server and Direct Shared Protocols”, gives details of how to perform other transaction-related POSTs, such as Refunds, Repeat payments and the Release/Abort mechanisms for Deferred transactions.

Direct and 3D-Secure

Direct payments with 3D-Authentication are a little more complicated because your customer has to be forwarded to their card issuer to authenticate themselves BEFORE a card authorisation can occur. You must have 3D-Secure active on your account before you can process this type of transaction. Contact support@sagepay.com for more information about setting this up (although all new accounts are set up with 3D-secure by default. Most likely you’ll just need to enable it in your My Sage Pay admin area. This will be covered more later.)

The process of obtaining a 3D-Secured authorisation begins in the same manner as non-authenticated transactions. Your customer fills up a shopping basket on your site, you collect their details, then present them with a payment page secured with your 128-bit SSL certificate. This page POSTs to a script on your site which pre-validates the data and formats a normal server-side Direct Transaction Registration POST (see Appendix A1) which is sent to Sage Pay.

As in a non-authenticated Direct transaction, the information you POST to us is validated against your IP address list and the data checked for range errors, but if everything appears in order, rather than immediately sending the card details to your acquiring bank for authorisation, the details are instead used to send a query to the 3D-Secure directory servers. These check to see if the card and the card-issuer are enrolled in the 3D-Secure scheme.

If the card or the issuer is NOT part of the scheme, Direct checks your 3D-Secure rule base (which you can modify in our My Sage Pay screens) to determine if you wish to proceed with the authorisation in such circumstances. If the card or the issuer is not part of the scheme and your rule base allows authorisation to proceed, the card details are sent to the acquiring bank and the results of that process returned to your site in the Response object of your POST (just like a non-3D-authenticated Direct transaction, but with an additional 3DSecureStatus field informing you about the results of the card lookup).

If authorisation cannot proceed because your rules do not allow it, a **REJECTED** message is sent back in the Response object of your POST, outlining the reason for the transaction rejection.

If, however, the card AND issuer are part of the 3D-Secure scheme, Direct does not attempt to obtain an authorisation from your acquiring bank. Instead it formats and encrypts a 3D-Secure request message called a **PaReq** and replies to your Direct POST in the Response object with this message, a unique transaction code called the **MD**, and the URL of the 3D-Secure authentication pages at the cardholder's Issuing Bank (in a field called **ACSURL**). You can store the MD value if you wish to, but the ACSURL and PAReq values should NEVER be stored in your database.

Your server creates a simple, automatically-submitted HTML form that POSTs the user, the MD and the PaReq fields across to the ACSURL, along with an additional field called the **TermUrl** which points to a page on your site to which the bank will return the customer when they have been authenticated.

From the user's perspective, they will have entered their card details on your payment page, clicked submit, and will find themselves transferred to their card issuer to validate their 3D-Secure credentials.

Once the user has completed their 3D-authentication, their Issuing Bank will redirect the customer back to a script on your site pointed to by the TermUrl. The user returns to your site along with the **MD** of the transaction and the results of their authentication in an encrypted field called the **PaRes**. Like before, Direct takes care of decrypting and decoding this information for you, so your TermUrl page simply needs to format a server side HTTPS POST containing the MD and the PaRes fields (all correctly URL Encoded) and send it to Direct. You do not need to store the MD or PaRes fields in your database.

Direct examines the PaRes to determine if authentication was successful. If it was, it retrieves all the details from your original Direct POST and goes on to obtain an authorisation from the your acquiring bank. It then replies with the results in the Response object of your TermUrl POST in the same format as a non-3D Secured transaction, but with two additional fields for you to store (the **3DSecureStatus** and the **CAVV** value; a unique value which indicates that the Authentication was successful).

If Direct examines your PaRes and finds that authentication was NOT successful, it again checks your 3D-Secure rule base to determine if you wish to proceed. Like the original Transaction Registration POST, if you wish to obtain authorisations for non-3D-authenticated transactions, Direct requests an authorisation from your acquiring bank and replies as normal; if not, Direct returns a **REJECTED** message and does not obtain an authorisation.

Your Terminal URL should update your database with the results of the authorisation (or lack thereof) and display a completion page to your customer.

Although more complex than a non-3D-authenticated Direct transaction, this process does remove a huge amount of the complexity involved in using your own Merchant Plug-In. Moreover, transactions which fully authenticate offer you the protection of a liability shift for card-related misuse, which is extremely valuable if you sell products or services that are likely to attract fraud.

Direct and PayPal

Sage Pay has integrated with PayPal Express Checkout, giving you the opportunity to add PayPal as a payment option on your payment pages.

This facility is available to merchants who are a certified PayPal Business Account holder. If you do not already have a PayPal Business Account, you can apply for a PayPal Account by visiting the PayPal website:

https://www.paypal-marketing.co.uk/merchantservices/sagepay/

This additional service can be included in your package at no additional cost (standard PayPal transaction fees will apply, please visit the following link for further details:

https://www.paypal.com/uk/cgi-bin/webscr?cmd=_display-receiving-fees-outside).

Sage Pay will only charge you our standard transaction rates, according to the Sage Pay package you choose.

To support PayPal Express Checkout using the Direct method involves a little more integration work at your site, but nothing more complex than is currently required for 3D-Authetication.

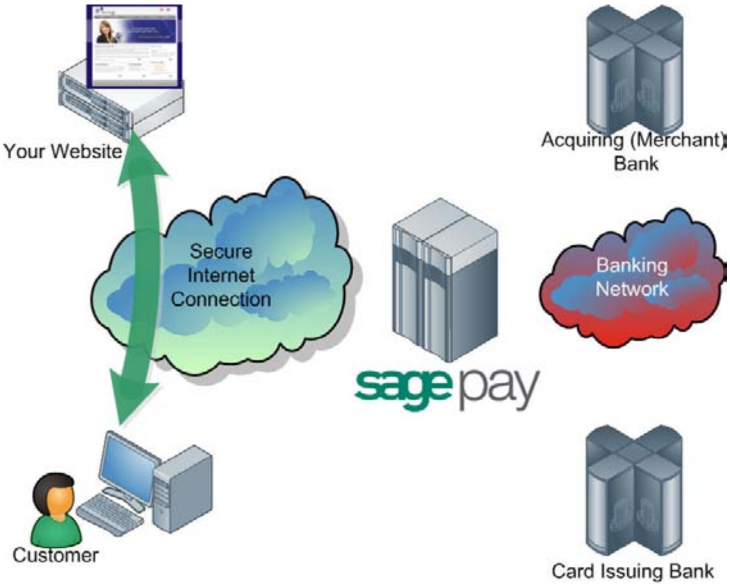
There is an initial server-to-server POST with Sage Pay, then a redirection to the PayPal logon URL. After that, there is a call back to your servers from Sage Pay, and an additional server-to-server POST to confirm the transaction and complete the process.

The Direct PayPal transaction process is detailed in the Direct PayPal Payments section later in this document.

The Direct payment process in detail (non-PayPal payments)

This section defines the messages exchanged between your Web servers and the Sage Pay Direct 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 Direct system puts no requirement on you to collect any specific set of information at this stage.

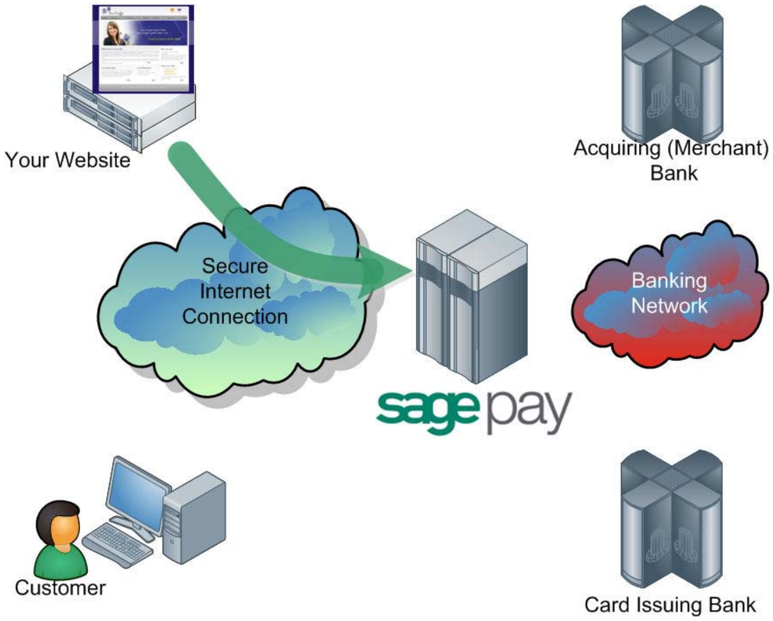
It is generally a good idea to identify the customer by name, e-mail address, delivery and billing address and telephone number. It is also helpful to have your server record the IP Address from which the customer is accessing your system. You should store these details in your database alongside details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

You then present a 128-bit SSL secured payment page into which the customer can enter their card and billing address details. This page should contain the following fields.

- The Cardholder Name as it appears on the card
- The Card Type (VISA, MC, DELTA, MAESTRO, UKE, AMEX, DC, JCB, LASER)
- The full Card Number without spaces or other separators
- The Start Date (if printed on the card)
- The Expiry Date
- The Issue Number (for some Maestro cards only)
- The Card Verification Value (called CVV or CV2 value: The extra three digits on the signature strip for most cards, or the 4 numbers printed on the front of an American Express card).
- The Card Holder's Billing Address, including the Post Code (if you have not already asked for it and stored it in your database).

If you wish to provide list boxes for Start and Expiry Dates, please be aware that Visa now issue cards valid for up to 20 years.

Step 2: Your server registers the payment with Sage Pay.



Once the customer has decided to proceed, a script on your web server will construct a payment registration message (see Appendix A1) and POST it via HTTPS to the Direct payment URL.

This POST contains your **Vendor Name** (chosen by you on the Sage Pay online application form, or assigned to you by Sage Pay when your account is created) and your own unique reference to this payment (in a field called **VendorTxCode**, which you must ensure is a completely unique value for each transaction).

The message also contains the total value and currency of the payment, and billing and delivery address details for the customer. You can specify a brief description of the goods bought to appear in your reports, plus the entire basket contents if you wish. The card details themselves are passed in dedicated fields whose format can be found in Appendix A1. You can also pass, contact numbers and e-mail addresses, flags to bypass or force fraud checking for this transaction and 3D-Secure reference numbers and IDs where such checks have been carried out.

Because this message is POSTed directly from your servers to ours across a 128-bit encrypted session, no sensitive information is passed via the customer's browser, and anyone who attempted to intercept the message would not be able to read it. Using the Direct method, you can be assured that the information you send us cannot be tampered with or understood by anyone other than us. Your script sends the payment registration message in the Request object of the HTTPS POST and the response from Direct (see steps 4 and 9 below) is in the Response object of the same POST.

On receipt of the POST, the Sage Pay gateway begins by validating its contents.

It first checks to ensure all the required fields are present, and that their format is correct. If any are not present a reply with a **Status** of **MALFORMED** is generated, with the **StatusDetail** field containing a human readable error message stating which field is missing. This normally only happens during development stage whilst you are refining your integration.

If all fields are present, the information in those fields is then validated. The Vendor name is checked against a pre-registered set of IP addresses, so that Direct can ensure the POST came from a recognised source. 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 amount field is validated. Flag fields are checked, in fact, every field is checked to ensure you have passed the correct types of value. If any of the information does not check out, a reply with a **Status** of **INVALID** is returned, again with a human readable error message in **StatusDetail** explaining what was invalid.

If you receive either a MALFORMED or INVALID message you should use the detailed response in the StatusDetail error message to help debug your scripts. If you receive these messages on your live environment, you should inform your customer that there has been a problem registering their transaction, then flag an error in your back-office systems to help you debug. You can e-mail the Sage Pay Support team (support@sagepay.com) for help with your debugging issues.

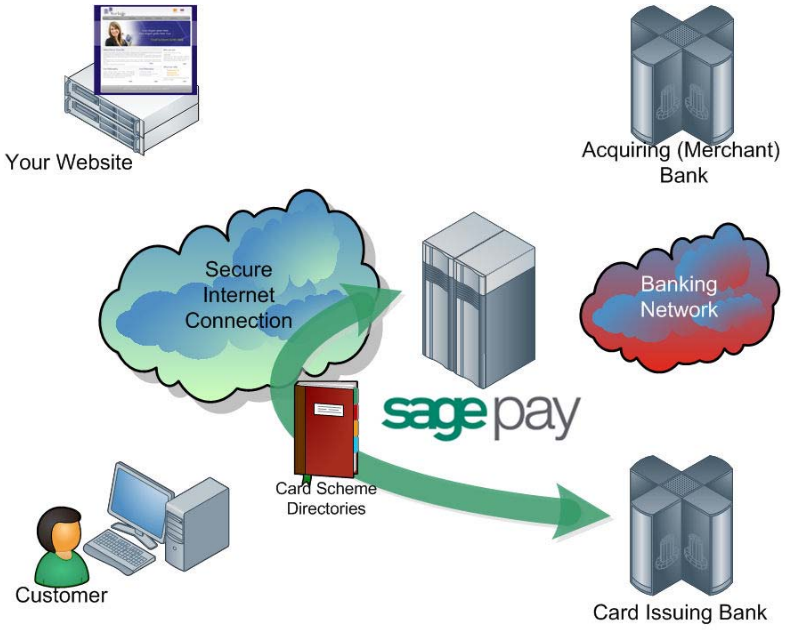
The integration kits we provide contain scripts in a variety of languages that illustrate how you compose and send this message from your server to ours. These can be downloaded as part of the application process or obtained from the download area on our website: http://www.sagepay.com/help/downloads.

Please note that you will need to log sign into the website to access these files.

When your transaction is registered with the Sage Pay gateway, a new transaction code is generated that is unique across ALL vendors using the Sage Pay systems, not just unique to you. This code, the **VPSTxId**, is our unique reference to the transaction and is returned to you in the response part of the POST after we've requested authorisation for you. This reference, whilst not the most easily remembered number, will allow us to immediately find your transaction if you have a query about it.

If your Sage Pay account is not set up with 3D-Secure or 3D-Authentication is not active for this transaction, the next step is for the system to obtain an authorisation, so skip ahead to step 8. If, however, 3D-Secure is active on your account, continue at step 3.

Step 3: Sage Pay Direct checks 3D-Secure enrolment.



The Sage Pay gateway sends the card details provided in your post 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 servers send a verification response, VERes, back to the Sage Pay MPI where it is decoded and informed of the inclusion or exclusion of the card.

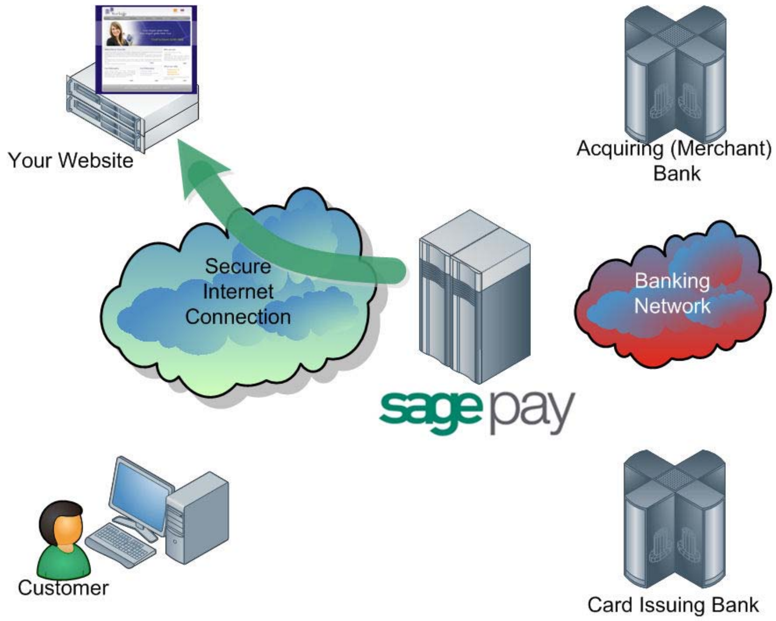
If the card or the issuer is not part of the scheme, or if an MPI error occurs, our server will check your 3D-Secure rule base to determine if authorisation should occur. For information regarding 3D Secure rule bases please refer to the Sage Pay Fraud Prevention Advice Guides, which can be downloaded from our website: http://www.sagepay.com/help/downloads. By default your account 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 your rulebase rejects the transaction due to your criteria not being reached, the gateway replies with a **Status** of **REJECTED** and a **StatusDetail** indicating why. 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, the Sage Pay gateway continues as though 3D-Secure is not active on your account. Jump ahead to step 8 in these circumstances, for the authorisation stage.

If the card and the card issuer are both part of the scheme, the sage Pay gateway continues with 3D-Authentication by replying to your post with a **Status** of **3DAUTH** (see the next step).

Step 4: Direct replies to your registration POST.



The Sage Pay servers stores all the information from your Transaction Registration POST in our secure database before replying (see Appendix A3). The **Status** field will be set to **3DAUTH** with a **StatusDetail** informing you to redirect your customer to their Issuing Bank to complete 3D-Authentication.

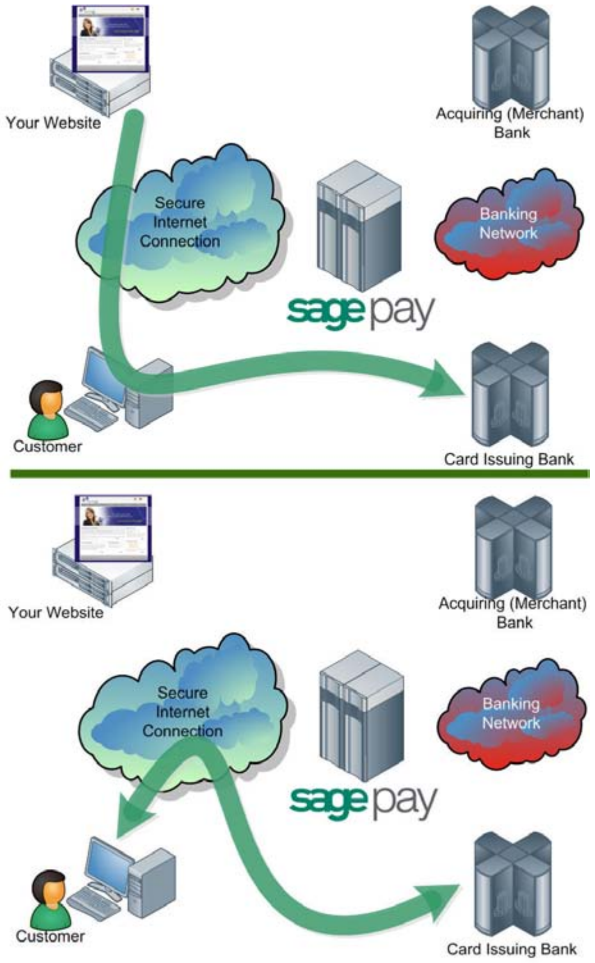
A unique identifier to your transaction called the **MD** is passed along with a preformatted, encrypted field called **PAReq**. This is the 3D-Secure message that the customer's card Issuing Bank decodes to begin the 3D-authentication process. The PAReq is created and encrypted by the Sage Pay MPI and you should not attempt to modify it. If you do, the 3D-Secure authentication step will fail and this, in turn, will fail your transaction.

A field called **ACSURL** (Access Control Server URL) contains the fully qualified address of the customer's card Issuing Bank's 3D-Secure module, as provided by the directory service (see step 3 above). The last field is the **3DSecureStatus** field, which will always contain **OK** for transactions ready for 3D-authentication.

You do not need to store any of these values in your database. You can store the MD value if you wish, but the ACSURL and PAReq values should NEVER be stored. These values need only be used in the next step to redirect your customer to their Issuing Bank and should then be discarded.

The first step of the Direct transaction is now complete. You have registered a 3D-Secure transaction with Sage Pay; we have stored your payment details and replied with everything you need to send your customer for 3D-Authentication. The next parts of the process, steps 5 and 6, are out of our control and rely on a communication between you, your customer and your customer's card Issuing Bank.

Step 5: You redirect your customer to their Issuing Bank.



The registration page code on your server should check the **Status** field, and when a **3DAUTH** status is found, build a simple, auto-submitting form (see the example below) which sends the **MD**, **PaReq** and an additional field, the **TermUrl**, to the address specified in the **ACSURL**, and send this form to your customer's browser.

This has the effect of redirecting your customer to their card Issuer's 3D-Authentication site whilst sending to that site all the information required to perform the authentication.

The **TermUrl** field is a fully qualified URL which points to the page on your servers to which the customer is sent once the 3D-authentication is completed (see step 6 below). Example code for this page is included in the integration kits provided by Sage Pay; see the example asp code below using an IFrame:

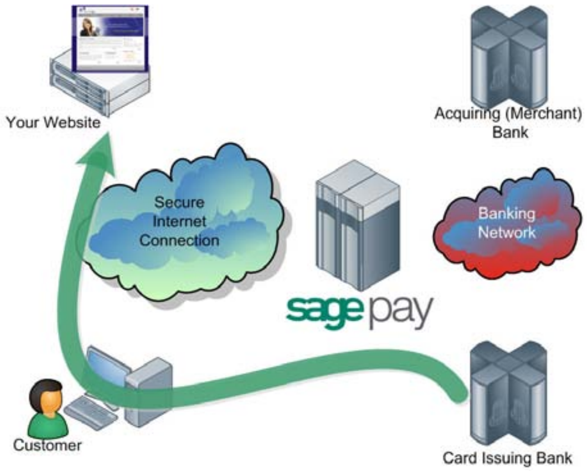
```
<IFRAME SRC="3DRedirect.asp" NAME="3DIFrame" WIDTH="100%" HEIGHT="500" FRAMEBORDER="0">
        <% 'Non-IFRAME browser support
        response.write "<SCRIPT LANGUAGE=""Javascript""> function OnLoadEvent()
{ document.form.submit(); }</" & "SCRIPT>"
        response.write "<html><head><title>3D Secure Verification</title></head>"
        response.write "<body OnLoad=""OnLoadEvent();"">"
        response.write "<FORM name=""form"" action=""" & strACSURL &""" method=""POST"">"
        response.write "<input type=""hidden"" name=""PaReq"" value=""" & strPAReq
&"""/>"
        response.write "<input type=""hidden"" name=""TermUrl"" value=""" &
strYourSiteFQDN & strVirtualDir & "/3DCallback.asp?VendorTxCode=" & strVendorTxCode &
"""/>"
        response.write "<input type=""hidden"" name=""MD"" value=""" & strMD &"""/>"

        response.write "<NOSCRIPT>"
        response.write "<center><p>Please click button below to Authenticate your
card</p><input type=""submit"" value=""Go""/></p></center>"
        response.write "</NOSCRIPT>"
        response.write "</form></body></html>"%>
</IFRAME>
```

The values in Red are those extracted from the Sage Pay response and built by your script. If your user has Javascript enabled, they simply redirect to their Issuing Bank site. If not, they will be presented with the message in the NOSCRIPT section and need to click it to go to their Issuing Bank.

At this stage the customer has left your site, and you must wait for them to be sent back to you by the Issuing Bank. NOTE: You can either redirect the customer's entire browser page to their Issuing Bank ACSURL, or more commonly, use an inline frame to redirect them. Visa recommend using inline frames for continuity of customer experience, but if you do so, remember to add code to support IFRAME incapable browsers (as above).

ADDITIONAL NOTE: When you forward the PAReq field to the ACSURL, please ensure you pass the PAReq value that we send you, in a field called **PaReq** (note the lower case "a"). Many ACSURL pages are case sensitive, and will not see the data if you pass an upper case A. See the example code above for how to submit the data.

Step 6: 3D-Authentication is carried out and your site called back.



Your customer completes the 3D-authentication process at their Issuing Bank's website.

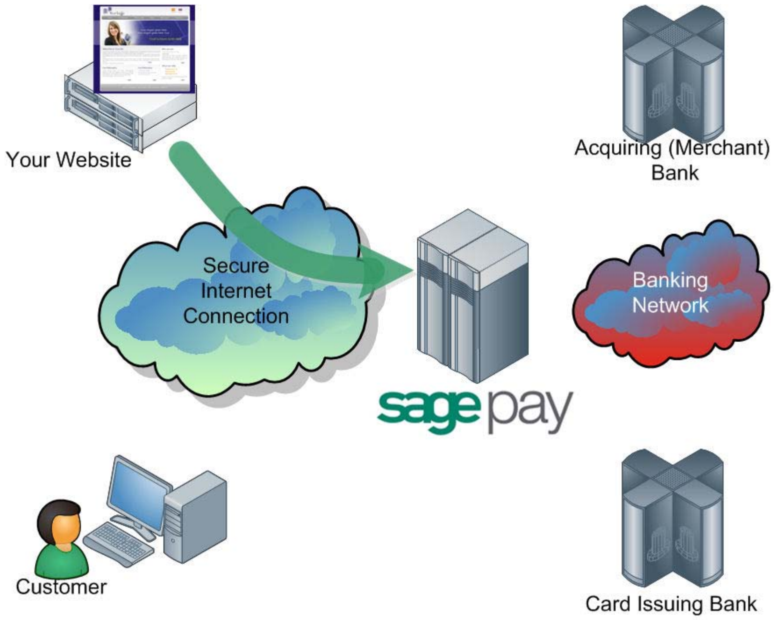
Once complete (either successfully or not), the bank will redirect your customer back to the page supplied in the **TermUrl** field you sent in step 5 above.

Along with this redirection, two fields are also sent. The **MD** value, to uniquely identify the transaction you are being called back about, and the **PARes**, the encrypted and encoded results of your customer's 3D-authentication.

Like the PAReq value sent to your site by Sage Pay in step 5, you should NOT store the PARes file in your database. Also, because it is strongly encrypted, only the Sage Pay MPI can decode this for you, so you should not attempt to modify it or the authentication process will fail.

At this stage the customer is back on your site and you have completion information for the 3D-Authentication process. You now need to send those through to Sage Pay to decode the results and, where appropriate, obtain a card authorisation from your acquiring bank.

Step 7: Your site POSTs the 3D-Secure results to Sage Pay.



The code in your TermUrl call-back page should format a simple HTTPS, server-side POST, which it sends to the Sage Pay Direct 3D-Callback page.

This POST needs to contain the **MD** and **PARes** fields sent back to your site by the cardholder's Issuing Bank (suitably URL Encoded for safe transit across the Web).

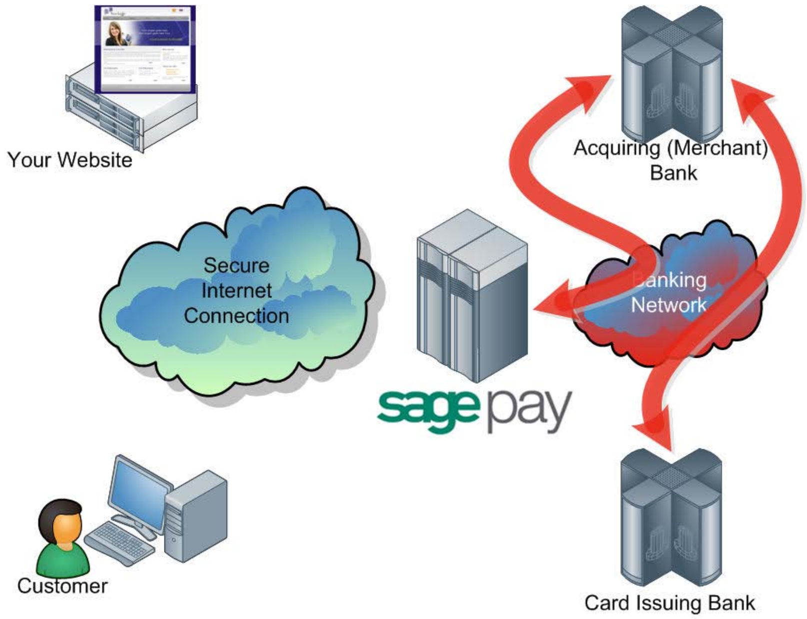
No other information is necessary because the Sage Pay system can use these values to retrieve all the transaction information you originally supplied.

If the decoded PARes indicates that the 3D-Authentication was successful, the Sage Pay gateway goes on to obtain an authorisation (see the next step). If not, the system examines your 3D-Secure rule base to see if authentication 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.

Transactions not sent for authorisation are returned with a **REJECTED** status.

ADDITIONAL NOTE: Similarly to the note in Step 5, the encrypted and encoded results of your customer's 3D-authentication (the PARes) will be returned to you from the Issuing bank in a field called PaRes (lower case "a"), but you must forward this value to Sage Pay in a field called **PARes**.

Step 8: Direct requests 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.

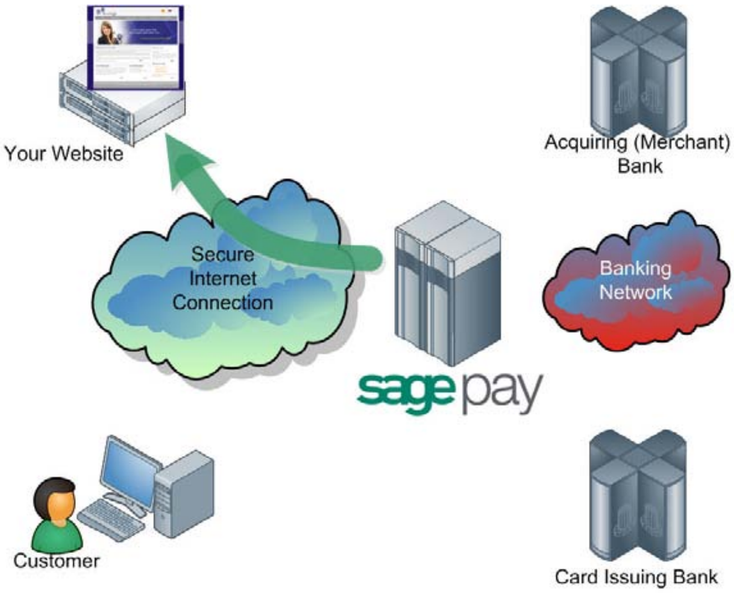
The request is normally answered within a second or so with either an authorisation code, or a failure message. This is obtained directly from the issuing bank by the acquiring bank in real time.

This process happens whilst the script on your server is waiting for a response from our servers. Depending on the response from the acquirer, the Sage Pay gateway prepares either an **OK** response with an Authorisation Code, a **NOTAUTHED** response if the bank declined the transaction or an **ERROR** if something has gone wrong (you will very rarely receive these, since they normally indicate an issue with bank connectivity).

If AVS and CV2 checks are being performed, the results are compared to any rule bases you have set up (see the Sage Pay Rulebase Guide for more information). If the bank has authorised the transaction but the card has failed the AVS and/or CV2 rules you have established, Sage Pay immediately reverse the authorisation on the card and prepares a **REJECTED** response, returning the reason for the failure in the AVSCV2 field.

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* however it may be seen by the customer like the funds have been taken.

Step 9: Sage Pay reply to your server's POST.



Irrespective of the Status being returned, the Sage Pay gateway always replies in the Response section of the POST that your server sent to us. This will either be in response to the Transaction Registration POST for non-3D-authenticated transactions, or in the response to the Terminal URL POST if 3D-Authentication was attempted.

If the transaction was registered successfully, you will always receive the **VPSTxId**, the unique transaction reference mentioned above. You will also receive a **SecurityKey**, a 10-digit alphanumeric code that is used in digitally signing the transaction. Whilst not used in the Direct transaction messages, you do need to know this value if you wish to REFUND the transaction, or perform any other automated actions on it using the Sage Pay Direct interface. Therefore this value should be stored alongside the **VPSTxId**, the order details and the **VendorTxCode**, in your database.

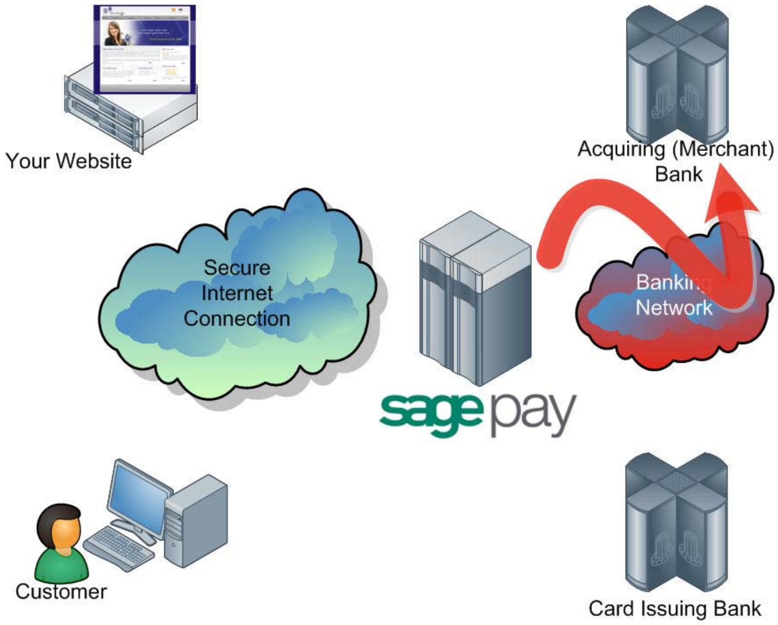
If the transaction was authorised and the Status field contains **OK**, you will also receive a field called **TxAuthNo**. The TxAuthNo field DOES NOT contain the actual Authorisation Code sent by the bank, because it is not unique (although we do store this in our system for you), but contains instead a unique reference number to that authorisation that we call the **VPSAuthCode**. This is the transaction ID sent to the bank during settlement (we cannot use your VendorTxCode because it is too long and might contain unacceptable characters) so the bank will use this value to refer to your transaction if they need to contact you about it. You should store this value in your database along with all the other values returned to you.

The TxAuthNo field is ONLY present if the transaction was authorised by the bank. All other messages are authorisation failures of one type or another (see Appendix A2 for full details of the fields and errors returned by the Direct gatewat) and you should inform your customer that their payment was not accepted.

If you do receive an **OK** status and a **TxAuthNo**, you should display a completion page for your customer thanking them for their order.

Having stored the relevant transaction IDs in your database, your payment processing is now complete.

Step 10: Sage Pay sends settlement batch files to confirm payments.



Once per day, from 12:01am, 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 will 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 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, 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.

Direct Payments using PayPal

The steps involved in using PayPal with Direct are detailed below and summarised at the end in a diagram.

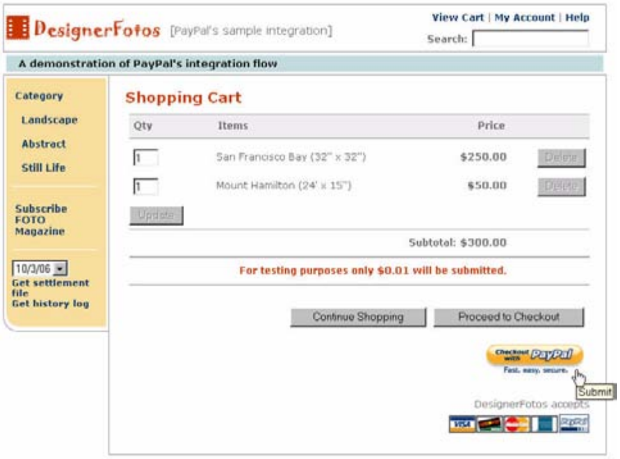
1. The customer shops at your site and fills up a shopping basket with items.

2. At the point the customer wishes to check-out, BEFORE they enter any address or customer details, your site can *optionally* allow the customer to select to pay either with PayPal, or another payment process. This is the "**Express Checkout**" option and should be presented using the orange logo:

   Checkout with PayPal
   Fast, easy, secure.

   To download the PayPal images to use on your own site visit: https://www.paypal.com/express-checkout-buttons

   e.g.

   
   DesignerFotos [PayPal's sample integration]

   View Cart | My Account | Help

   Search:

   A demonstration of PayPal's integration flow

   Category
   Landscape
   Abstract
   Still Life

   Subscribe FOTO Magazine

   10/3/06
   Get settlement file
   Get history log

   Shopping Cart

Qty	Items	Price	
1	San Francisco Bay (32" x 32")	$250.00	Delete
1	Mount Hamilton (24' x 15")	$50.00	Delete

   Update

   Subtotal: $300.00

   For testing purposes only $0.01 will be submitted.

   Continue Shopping   Proceed to Checkout

   This is optional because you may wish to use PayPal as a different "card type" rather than an alternative payment method, (for example, if you wish to collect customer details on your own pages first). This latter option is described later.

   If the customer selects this Express Checkout option, the process jumps ahead to step 6.

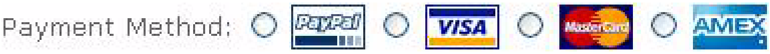
3. Since the customer has not selected Express Checkout (or has not had the option to do so), your site presents the normal customer detail entry screens, requesting name, e-mail address, and billing address in the following format:

   Name (compulsory - 32 chars max)
   Street (compulsory - 100 chars max)
   Street2 (optional - 100 chars max)
   City (compulsory - 40 chars max)
   Zip (compulsory - 20 chars max)
   Country (compulsory – 2 digit ISO 3166-1 code)
   State (compulsory for US Addresses only)
   Phone (optional – 20 characters)

This structure is required to allow PayPal to validate the addresses against those held in their database.

4. Once the customer has entered their address details, they select their card type, as in a normal Direct payment, with the addition of the PayPal Logo.

   (or “PayPal” as a drop down item in a card-type list)

   This is called a “**Mark**” integration (as opposed to “Express Checkout” integration above). From a Sage Pay perspective, the process is almost identical.

5. If the customer selects a method other than PayPal, then the normal Direct process with 3D-authentication continues from this point onwards, as detailed in the Direct payment process above, i.e. the customer enters the card number, expiry date, CV2 etc. and the full server-to-server POST is sent.

   If the customer has selected PayPal, either Mark or Express Checkout, the new process begins at this step.

6. The Direct registration message (see Appendix A1) is sent with the **cardtype** field set to **PAYPAL** (no other card details should be sent). Mark implementations will also require the full **billingnnnnn** AND **deliverynnnn** sections to be completed as detailed above, but Express Checkouts will leave these empty. This POST also includes a **PayPalCallbackURL** field which points to a script on your site to handle the completion process (explained in step 11).

7. The information is POSTed to the Direct Transaction Registration URL and the POST is validated as normal. If all fields are validated and the information is correct, the Sage Pay servers construct a message to send to the PayPal servers; for “Express Checkouts”, as you have not collected customer details on your own pages first, a message is sent to ensure the customer enters their address once they reach the PayPal screens. “Mark” checkouts will already have the address information provided, and therefore the customer will not have the option to select an alternative address once on the PayPal screens.

8. The PayPal servers respond to Sage Pay with a unique token. The transaction is updated in the Sage Pay Database to record this token against the transaction, before returning the Direct response to your servers (Appendix A6).

9. Your site redirects the customer’s browser to the **PayPalRedirectURL** value returned in the Direct response (Appendix A6).

10. The customer logs into PayPal and selects their chosen payment method. For Express Checkouts they will also enter/select their delivery address. For Mark, this address selection is disabled.

11. Once the shopper confirms their details on the PayPal screens, PayPal exchange information with Sage Pay, and then Direct builds a response

message containing the fields listed in Appendix A7. This data is POSTed via the customer's browser to the **PayPalCallbackURL** (which you provided as part of the original Direct POST Appendix A1). This URL is also the place to which the customer's browser is redirected in the event of any errors.

12. Your site can check the information in the message to determine if you wish to proceed with the transaction. If the **AddressStatus** is **UNCONFIRMED**, and the **PayerStatus** is **UNVERIFIED**, for example, you may not wish continue without PayPal Seller Protection. If you do NOT wish to proceed, you should build a Direct PayPal Capture message with the **Accept** field set to **NO** (Appendix A8) and POST it to the Direct PayPal Completion URL. You can then redirect the customer back to select a different payment method at this stage, and begin the Direct process again.

13. If you DO wish to proceed, you should store the delivery address details in your database (if they differ from those supplied), then build a Direct PayPal Capture message with the **Accept** field set to **YES** (Appendix A8) and POST it to the Direct PayPal Completion URL.

14. Direct will validate the POST and, if correct, forward that to PayPal.

15. PayPal will complete the transaction and return the details to Direct. Sage Pay will update the transaction with the required IDs and build a completion response.

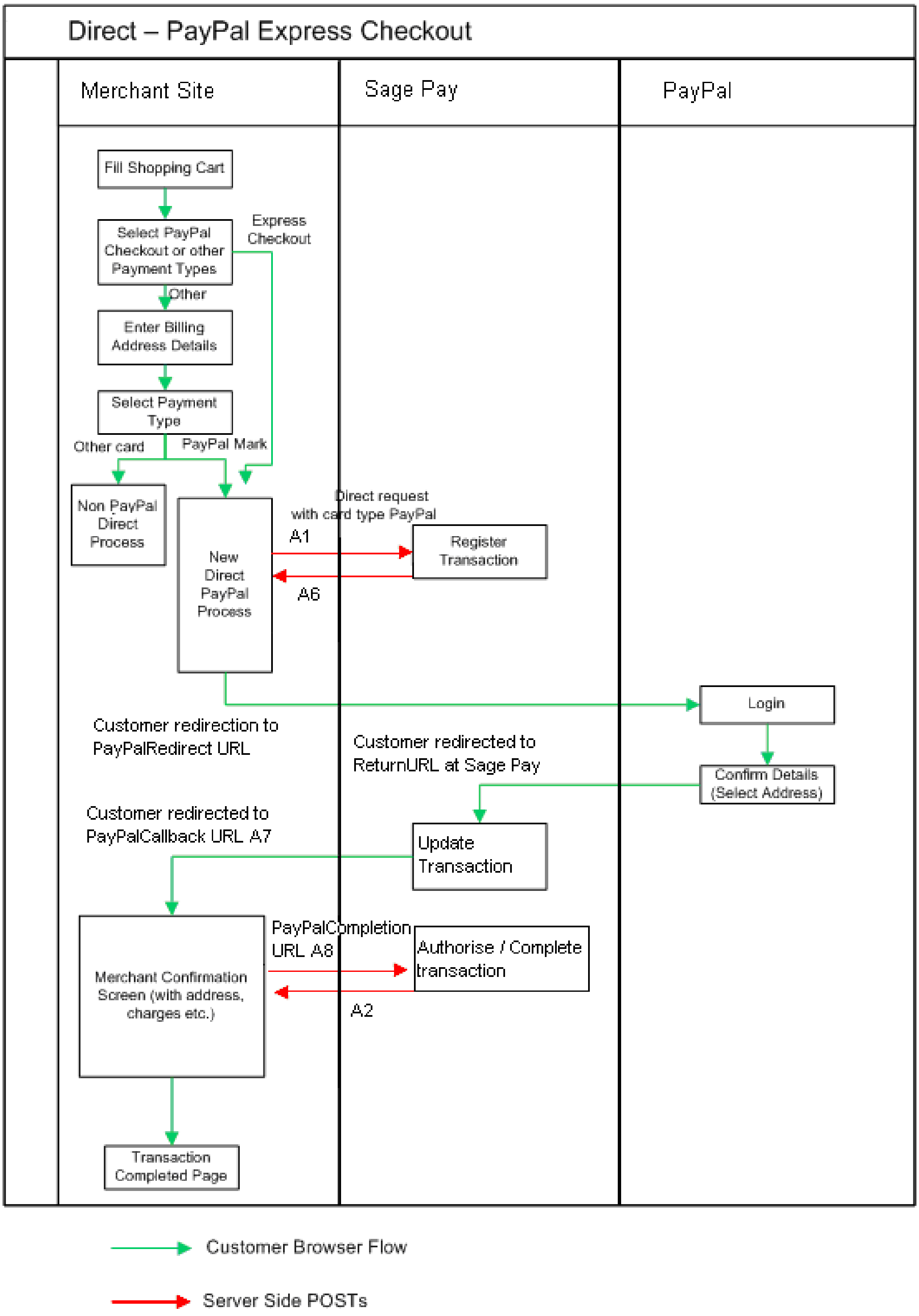
16. Direct replies to the POST sent to the PayPal Completion URL with the Direct completion message (Appendix A2).

17. You display a completion page to the customer.

Please note, if you wish to integrate PayPal using the Direct integration method, you will need to contact our support team (support@sagepay.com) to ensure that the TXTYPE=COMPLETE, (the PayPal callback URL) is enabled on your account.

Direct PayPal Message Flow

The diagram below show the message and customer flow for a Direct PayPal payment (described above).


Direct – PayPal Express Checkout
Merchant Site
Sage Pay
PayPal
Fill Shopping Cart
Select PayPal Checkout or other Payment Types
Express Checkout
Other
Enter Billing Address Details
Select Payment Type
Other card
PayPal Mark
Non PayPal Direct Process
New Direct PayPal Process
Direct request with card type PayPal
A1
Register Transaction
A6
Login
Customer redirection to PayPalRedirect URL
Customer redirected to ReturnURL at Sage Pay
Confirm Details (Select Address)
Customer redirected to PayPalCallback URL A7
Update Transaction
PayPalCompletion URL A8
Authorise / Complete transaction
A2
Merchant Confirmation Screen (with address, charges etc.)
Transaction Completed Page
Customer Browser Flow
Server Side POSTs

Integrating with Direct

Linking your Website to Sage Pay with Direct involves creating one script (or modifying the example provided in the integration kits), which both registers the transaction with our servers and processes the response we send back. If you wish to support 3D-Secure Authentication, you will also need to create or modify a second script to handle the call back from the Issuing Bank. If you wish to integrate with PayPal, additional coding is also required to redirect to a PayPal logon URL. After that, there is a call back to your servers from Sage Pay, and an additional server-to-server POST to confirm the transaction and complete the process.

Stage 1

The 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 Direct, 3D-authentication and PayPal systems.

The 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/VSPDirectGateway.asp

3D-secure callback POSTS should be sent to the following URL:
https://test.sagepay.com/Simulator/VSPDirectCallback.asp

PayPal Completion POSTS should be sent to the following URL:
https://test.sagepay.com/Simulator/paypalcomplete.asp

Stage 2

Once your site is able to talk to the Simulator and process all possible outcomes, an account will be created for you on the 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 run here so you can familiarise yourself with the features available to you.

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

Transactions from your scripts should be sent to the Test Site at:
https://test.sagepay.com/gateway/service/vspdirect-register.vsp

3D-secure callback POSTS should be sent to the following URL:
https://test.sagepay.com/gateway/service/direct3dcallback.vsp

PayPal Completion POSTS should be sent to the following URL:
https://test.sagepay.com/gateway/service/complete.vsp

Stage3

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 and your Merchant Account details have been confirmed, your account will be set up on our Live servers. You then 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 service so you don't charge yourself. If this works successfully, then you are ready to trade online.

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

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

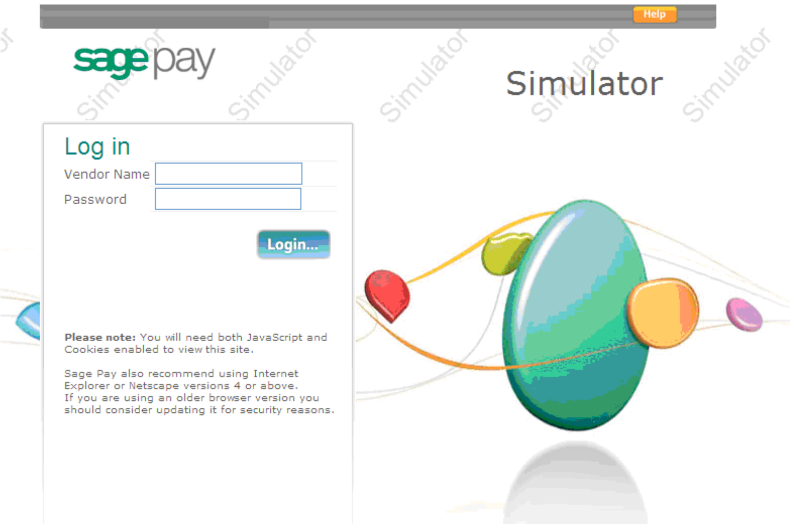
3D-secure callback POSTS should be sent to the following URL:
https://live.sagepay.com/gateway/service/direct3dcallback.vsp

PayPal Completion POSTS should be sent to the following URL:
https://live.sagepay.com/gateway/service/complete.vsp

Stage 1: Integrating with the Simulator

The SAGE PAY Simulator is an expert system that emulates the Sage Pay 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 gateway, allowing you to debug and enhance your code at an earlier stage.

Log into the 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).



Please note: You will need both JavaScript and Cookies enabled to view this site.

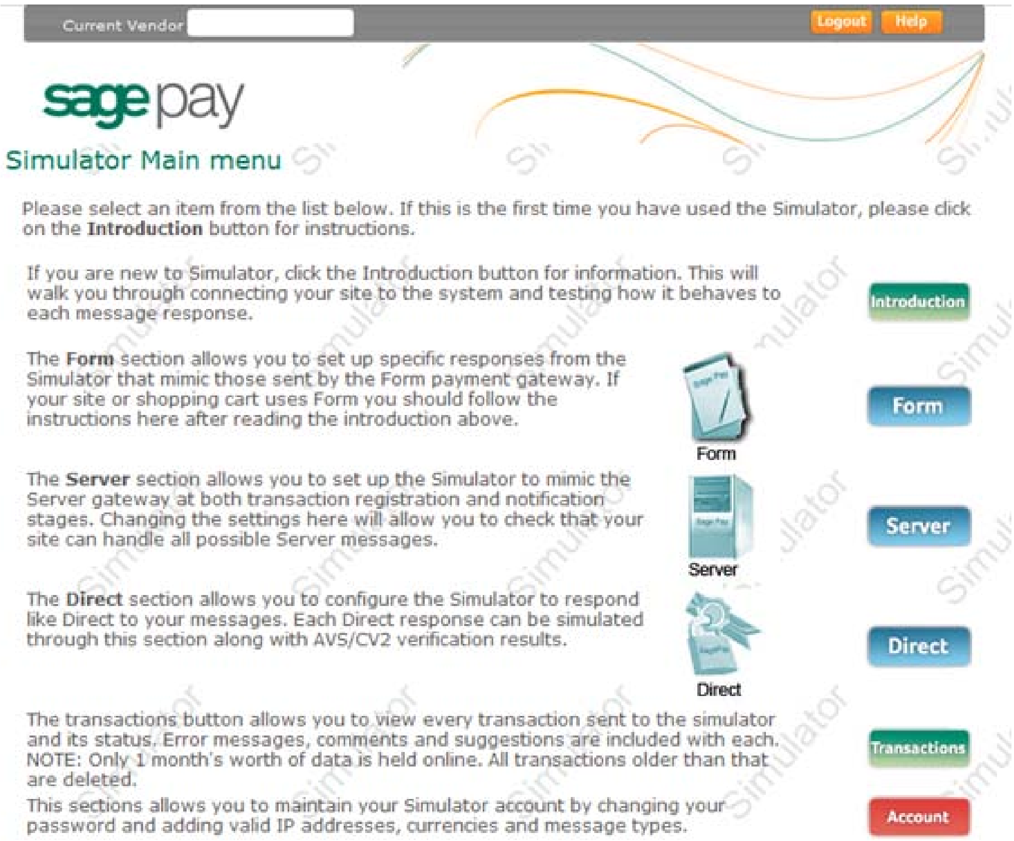
Sage Pay also recommend using Internet Explorer or Netscape versions 4 or above. If you are using an older browser version you should consider updating it for security reasons.

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 Simulator. To register for a Simulator account, please visit our website:

https://support.sagepay.com/apply/requestsimaccount.aspx

If you already have a Test/Live account and would like a Simulator account, contact our Support Team at support@sagepay.com, stating the Vendor Name of your account.

When you log in to the Simulator you will be presented with the main menu screen.



Simulator Main menu

Please select an item from the list below. If this is the first time you have used the Simulator, please click on the **Introduction** button for instructions.

If you are new to Simulator, click the Introduction button for information. This will walk you through connecting your site to the system and testing how it behaves to each message response.

The **Form** section allows you to set up specific responses from the Simulator that mimic those sent by the Form payment gateway. If your site or shopping cart uses Form you should follow the instructions here after reading the introduction above.

The **Server** section allows you to set up the Simulator to mimic the Server gateway at both transaction registration and notification stages. Changing the settings here will allow you to check that your site can handle all possible Server messages.

The **Direct** section allows you to configure the Simulator to respond like Direct to your messages. Each Direct response can be simulated through this section along with AVS/CV2 verification results.

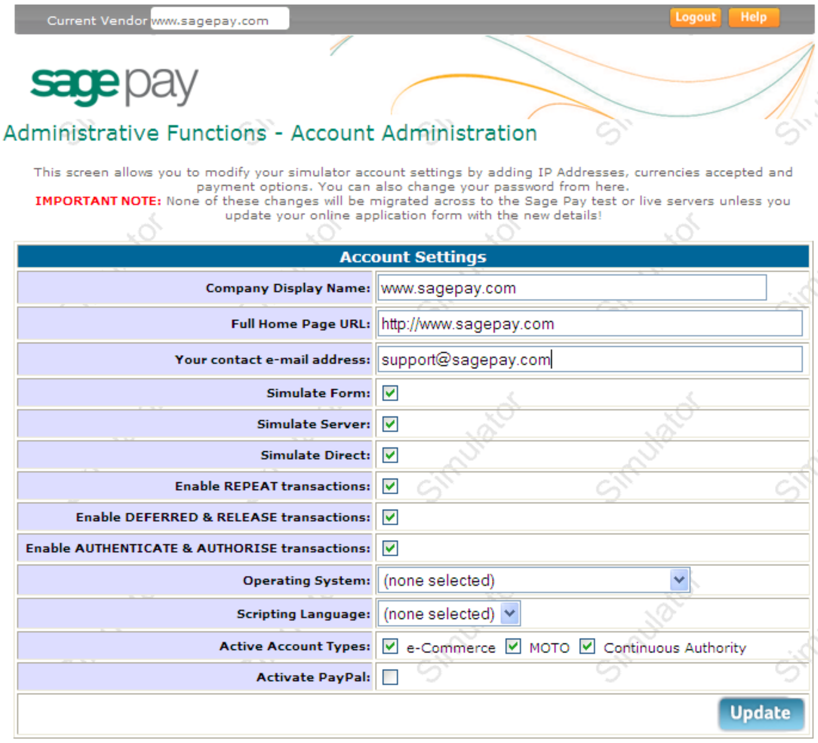
The transactions button allows you to view every transaction sent to the simulator and its status. Error messages, comments and suggestions are included with each. NOTE: Only 1 month's worth of data is held online. All transactions older than that are deleted.

This sections allows you to maintain your Simulator account by changing your password and adding valid IP addresses, currencies and message types.

Extensive help is provided in the Simulator (click the context sensitive Help button on each screen for more details). 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: Simulator account set up

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



Current Vendor www.sagepay.com
Logout
Help
Administrative Functions - Account Administration

This screen allows you to modify your simulator account settings by adding IP Addresses, currencies accepted and payment options. You can also change your password from here.
IMPORTANT NOTE: None of these changes will be migrated across to the Sage Pay test or live servers unless you update your online application form with the new details!

Account Settings	
Company Display Name:	www.sagepay.com
Full Home Page URL:	http://www.sagepay.com
Your contact e-mail address:	support@sagepay.com
Simulate Form:	☑
Simulate Server:	☑
Simulate Direct:	☑
Enable REPEAT transactions:	☑
Enable DEFERRED & RELEASE transactions:	☑
Enable AUTHENTICATE & AUTHORISE transactions:	☑
Operating System:	(none selected)
Scripting Language:	(none selected)
Active Account Types:	☑ e-Commerce ☑ MOTO ☑ Continuous Authority
Activate PayPal:	□

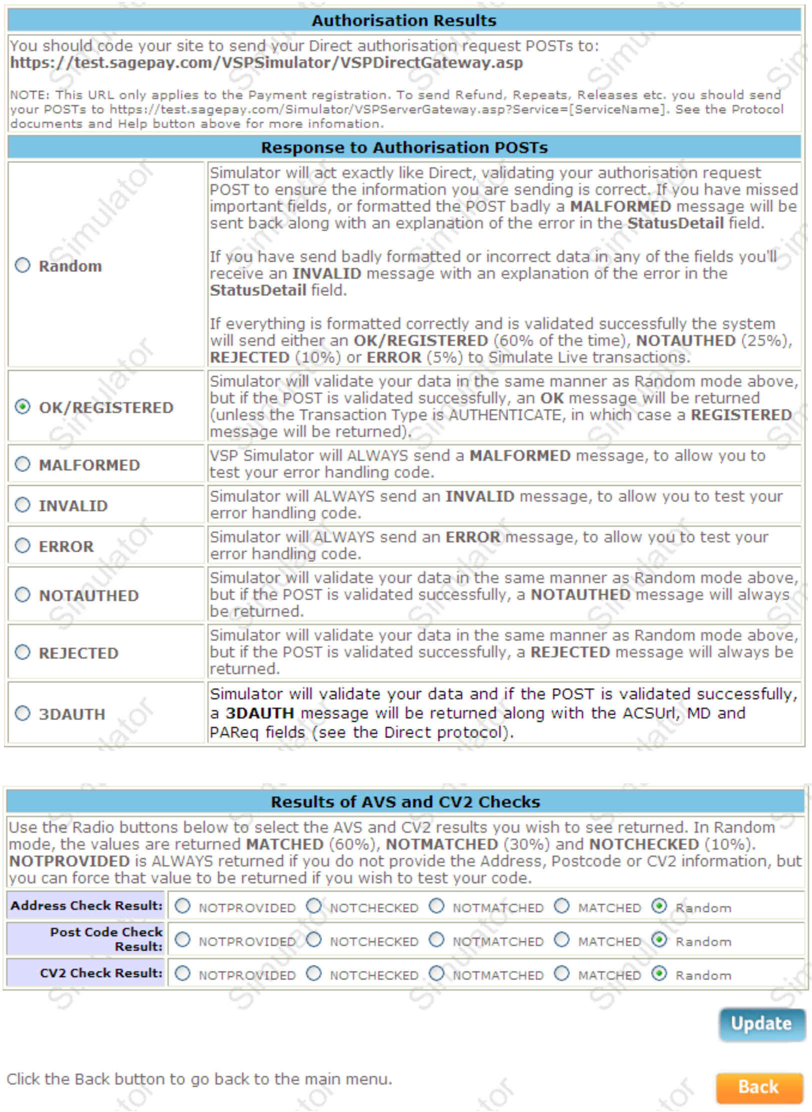
You should ensure that:

- all company details are correct.
- all technical details about web server and platform are correct.
- the Direct box is checked.
- all relevant payment types have been set up, and Activate PayPal has been enabled (if you wish to test your PayPal integration).
- you have at least one payment currency set up (usually GBP unless your site can take multi-currency transactions).
- the IP addresses of your servers are listed.

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

2: Direct integration set up

Click the Direct button in the main menu to open the Direct options page.



Authorisation Results

You should code your site to send your Direct authorisation request POSTs to:
https://test.sagepay.com/VSPSimulator/VSPDirectGateway.asp

NOTE: This URL only applies to the Payment registration. To send Refund, Repeats, Releases etc. you should send your POSTs to https://test.sagepay.com/Simulator/VSPServerGateway.asp?Service=[ServiceName]. See the Protocol documents and Help button above for more infomation.

Response to Authorisation POSTs

○ **Random**	Simulator will act exactly like Direct, validating your authorisation request POST to ensure the information you are sending is correct. If you have missed important fields, or formatted the POST badly a **MALFORMED** message will be sent back along with an explanation of the error in the **StatusDetail** field. If you have send badly formatted or incorrect data in any of the fields you'll receive an **INVALID** message with an explanation of the error in the **StatusDetail** field. If everything is formatted correctly and is validated successfully the system will send either an **OK/REGISTERED** (60% of the time), **NOTAUTHED** (25%), **REJECTED** (10%) or **ERROR** (5%) to Simulate Live transactions.
● **OK/REGISTERED**	Simulator will validate your data in the same manner as Random mode above, but if the POST is validated successfully, an **OK** message will be returned (unless the Transaction Type is AUTHENTICATE, in which case a **REGISTERED** message will be returned).
○ **MALFORMED**	VSP Simulator will ALWAYS send a **MALFORMED** message, to allow you to test your error handling code.
○ **INVALID**	Simulator will ALWAYS send an **INVALID** message, to allow you to test your error handling code.
○ **ERROR**	Simulator will ALWAYS send an **ERROR** message, to allow you to test your error handling code.
○ **NOTAUTHED**	Simulator will validate your data in the same manner as Random mode above, but if the POST is validated successfully, a **NOTAUTHED** message will always be returned.
○ **REJECTED**	Simulator will validate your data in the same manner as Random mode above, but if the POST is validated successfully, a **REJECTED** message will always be returned.
○ **3DAUTH**	Simulator will validate your data and if the POST is validated successfully, a **3DAUTH** message will be returned along with the ACSUrl, MD and PAReq fields (see the Direct protocol).

Results of AVS and CV2 Checks

Use the Radio buttons below to select the AVS and CV2 results you wish to see returned. In Random mode, the values are returned **MATCHED** (60%), **NOTMATCHED** (30%) and **NOTCHECKED** (10%). **NOTPROVIDED** is ALWAYS returned if you do not provide the Address, Postcode or CV2 information, but you can force that value to be returned if you wish to test your code.

Address Check Result:	○ NOTPROVIDED ○ NOTCHECKED ○ NOTMATCHED ○ MATCHED ● Random
Post Code Check Result:	○ NOTPROVIDED ○ NOTCHECKED ○ NOTMATCHED ○ MATCHED ● Random
CV2 Check Result:	○ NOTPROVIDED ○ NOTCHECKED ○ NOTMATCHED ○ MATCHED ● Random

Click the Back button to go back to the main menu.

This page allows you to define the behaviour of the Simulator when it responds to your transaction registrations. By default the system will verify your POST to ensure the contents are correctly formatted and if they are, return a Random Status, along with all relevant fields (StatusDetail, VPSTxID, SecurityKey and so on). If your POST is incorrectly formatted or contains bad data, it will respond with a Status of MALFORMED or INVALID and explain what was wrong in the StatusDetail field.

This setting is useful in the final stages of testing, but for now you should select **OK/REGISTERED**, to force the Simulator to always provide a TxAuthNo if your data is not MALFORMED or INVALID (or a REGISTERED if your transaction type is an AUTHENTICATE). You can switch it back to Random once you are certain your system can process a successful end-to-end transaction.

You can simulate **NOTAUTHED** responses from the bank, or **REJECTED** messages as if your rule bases were applied. By simulating these messages you can check your own pages respond to the customer in the correct manner. You can also force errors, even if the data in the POST is okay. This is useful when testing upgrades to your scripts and proofing your error handling routines.

For now, select the **OK** setting, update the details, and log out of the Simulator.

3: Registering a payment

If you do not 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 downloads area (www.sagepay.com/help/downloads).

The kits will not quite run out of the box because you have to provide some specific details about your site in the configuration files before a transaction can occur, but they will provide end to end examples of registering the transactions and handling the notification POSTs. Ensure you've completed all configuration in the `includes` file as detailed in the kit instructions, then locate the Transaction Registration script (called `transactionRegistration`).

This script provides a worked example of how to construct the Transaction Registration POST (see Appendix A section A1 in the attached protocol) and how to read the response that comes back (section A2).

If you plan to implement 3D-Secure Authentication, the kit also provides a Terminal URL example page which implements section A3 of the attached protocol.

Check that the payment registration script is sending transactions to the Simulator (rather than the test or live sites) then execute this script. You may need to develop a simple payment page that allows you to enter card details and passes them to this script if this page is not included in your kit. Use the script to send a payment registration to the Simulator. You may wish to modify the script at this stage to echo the results of the POST to the screen, or a file, so you can examine the Status and StatusDetail reply fields to check for errors.

Once your script can successfully register a Payment and you receive a **Status** of **OK**, you should ensure your code stores the **VPSTxId**, **SecurityKey** and **TxAuthNo** fields alongside your uniquely generated **VendorTxCode** and the order details in your own database. You may wish to store the **3DSecureStatus** field if you plan to support 3D-Secure.

Your script should then redirect the customer to a completion page thanking them for their order.

In the real world, the bank will either authorise the transaction (an **OK** response) or fail it (a **NOTATUHED** response), or Sage Pay may reverse an authorisation if your fraud screening rules are not met (a **REJECTED** response). You should log into the Simulator in a separate browser window and change the response type to each of the failure messages in turn so you can write code to handle each message appropriately. Normally NOTAUTHED messages would prompt the user to try another card and REJECTED messages would ask them to check their Address and CV2 details are correct and resubmit, or to try another card. You may wish to store the VPSTxId and SecurityKey of the failed transaction against your VendorTxCode and generate a new VendorTxCode for the retry attempt if you wish to keep a history of the failed transactions as well as the successful one.

You should then use the Simulator to send each type of error message (MALFORMED, INVALID and ERROR) to your payment script to check that all message types are handled correctly. **MALFORMED** messages should only occur during development when the POST may be incorrectly formatted, and **INVALID** messages can be avoided by pre-validating the user input. In the case of **ERROR**, your code should present the customer with a page saying that online payment was not currently available and offering them an alternative contact telephone number for payment or request them to come back later.

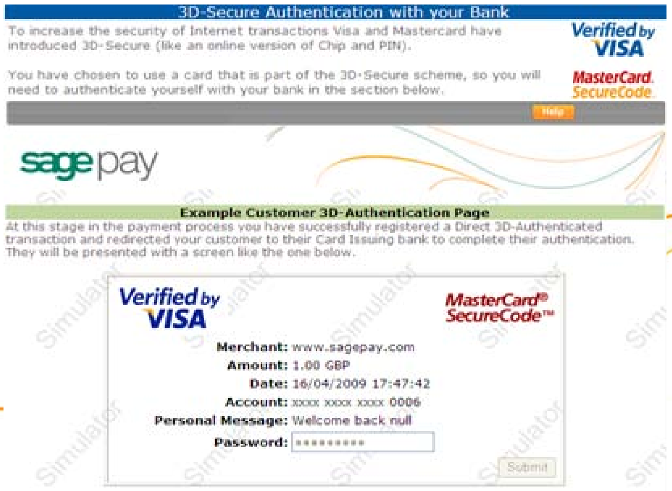
Once your page can handle every type of message returned by the Simulator, you should set the Simulator to Random mode and attach the payment pages onto the end of your e-commerce site to test end-to-end transactions through your site. The Random mode will respond with an OK message 60% of the time, a NOTAUTHED 25% of the time, REJECTED 10% and ERROR 5% to allow you to ensure you site respond correctly in each circumstance.

4: 3D-Authenticated Transactions

If you plan to support Verified by Visa and MasterCard SecureCode, collectively the 3D-Secure authentication system, you should now go on to test that your scripts can handle these messages. You should ONLY do this once your transaction registration script can successfully process non-authenticated transactions as described in section 3 above.

Log into the Simulator, bring up the Direct configuration page and select **3DAUTH** as the response. Click the Update button.

Send a transaction registration POST to the Simulator and rather than receiving an OK status, your script will receive a **3DAUTH** Status instead. A simulated **MD**, **PAReq** and **ACSURL** will be provided and you should ensure that your script builds the simple, automatically-submitting, HTML FORM code (as described in the step by step transaction process earlier in this document) and redirects your browser to the simulated 3D-Authentication page.



3D-Secure Authentication with your Bank

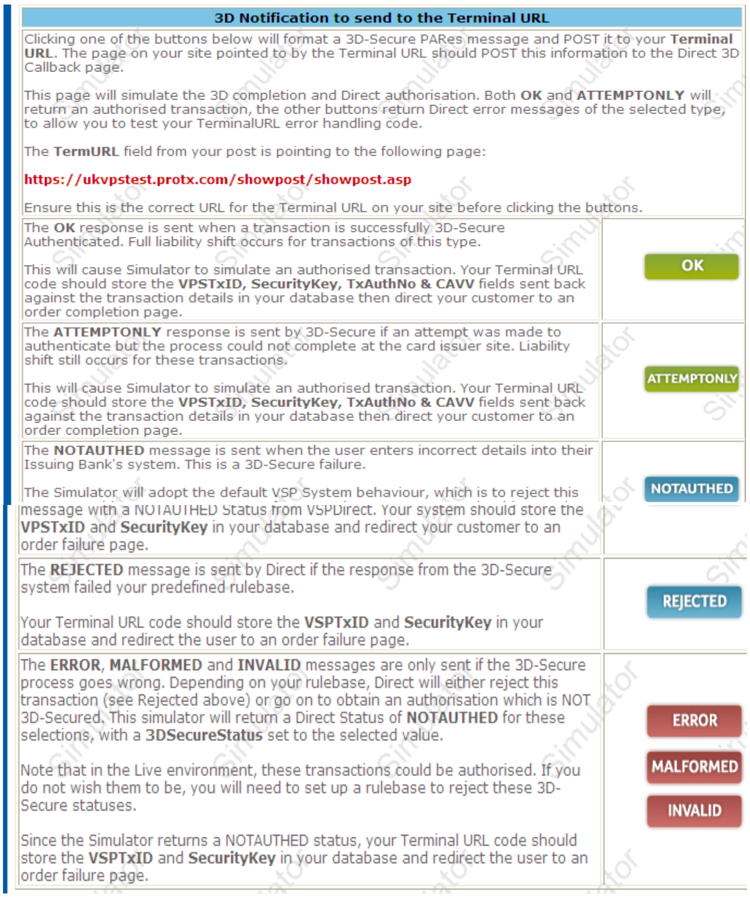
To increase the security of Internet transactions Visa and Mastercard have introduced 3D-Secure (like an online version of Chip and PIN).

You have chosen to use a card that is part of the 3D-Secure scheme, so you will need to authenticate yourself with your bank in the section below.

Example Customer 3D-Authentication Page

At this stage in the payment process you have successfully registered a Direct 3D-Authenticated transaction and redirected your customer to their Card Issuing bank to complete their authentication. They will be presented with a screen like the one below.

Merchant: www.sagepay.com
Amount: 1.00 GBP
Date: 16/04/2009 17:47:42
Account: xxxx xxxx xxxx 0006
Personal Message: Welcome back null
Password: *********



3D Notification to send to the Terminal URL

Clicking one of the buttons below will format a 3D-Secure PARes message and POST it to your **Terminal URL**. The page on your site pointed to by the Terminal URL should POST this information to the Direct 3D Callback page.

This page will simulate the 3D completion and Direct authorisation. Both **OK** and **ATTEMPTONLY** will return an authorised transaction, the other buttons return Direct error messages of the selected type, to allow you to test your TerminalURL error handling code.

The **TermURL** field from your post is pointing to the following page:

https://ukvpstest.protx.com/showpost/showpost.asp

Ensure this is the correct URL for the Terminal URL on your site before clicking the buttons.

The **OK** response is sent when a transaction is successfully 3D-Secure Authenticated. Full liability shift occurs for transactions of this type. This will cause Simulator to simulate an authorised transaction. Your Terminal URL code should store the **VPSTxID, SecurityKey, TxAuthNo & CAVV** fields sent back against the transaction details in your database then direct your customer to an order completion page.	OK
The **ATTEMPTONLY** response is sent by 3D-Secure if an attempt was made to authenticate but the process could not complete at the card issuer site. Liability shift still occurs for these transactions. This will cause Simulator to simulate an authorised transaction. Your Terminal URL code should store the **VPSTxID, SecurityKey, TxAuthNo & CAVV** fields sent back against the transaction details in your database then direct your customer to an order completion page.	ATTEMPTONLY
The **NOTAUTHED** message is sent when the user enters incorrect details into their Issuing Bank's system. This is a 3D-Secure failure. The Simulator will adopt the default VSP System behaviour, which is to reject this message with a NOTAUTHED Status from VSPDirect. Your system should store the **VPSTxID** and **SecurityKey** in your database and redirect your customer to an order failure page.	NOTAUTHED
The **REJECTED** message is sent by Direct if the response from the 3D-Secure system failed your predefined rulebase. Your Terminal URL code should store the **VSPTxID** and **SecurityKey** in your database and redirect the user to an order failure page.	REJECTED
The **ERROR**, **MALFORMED** and **INVALID** messages are only sent if the 3D-Secure process goes wrong. Depending on your rulebase, Direct will either reject this transaction (see Rejected above) or go on to obtain an authorisation which is NOT 3D-Secured. This simulator will return a Direct Status of **NOTAUTHED** for these selections, with a **3DSecureStatus** set to the selected value. Note that in the Live environment, these transactions could be authorised. If you do not wish them to be, you will need to set up a rulebase to reject these 3D-Secure statuses. Since the Simulator returns a NOTAUTHED status, your Terminal URL code should store the **VSPTxID** and **SecurityKey** in your database and redirect the user to an order failure page.	ERROR MALFORMED INVALID

This page displays the Terminal URL you have provided, and you should check that this points to the fully qualified URL of the Callback page provided in your kit. This should begin with https:// (since the Terminal URL must be secured) and provide the full path to the page.

If the URL is correct, you can select one of the buttons to create a Simulated **PARes** message that, when forwarded by your Terminal URL code to the Simulator 3D Callback page, will generate the Direct result of your choosing.

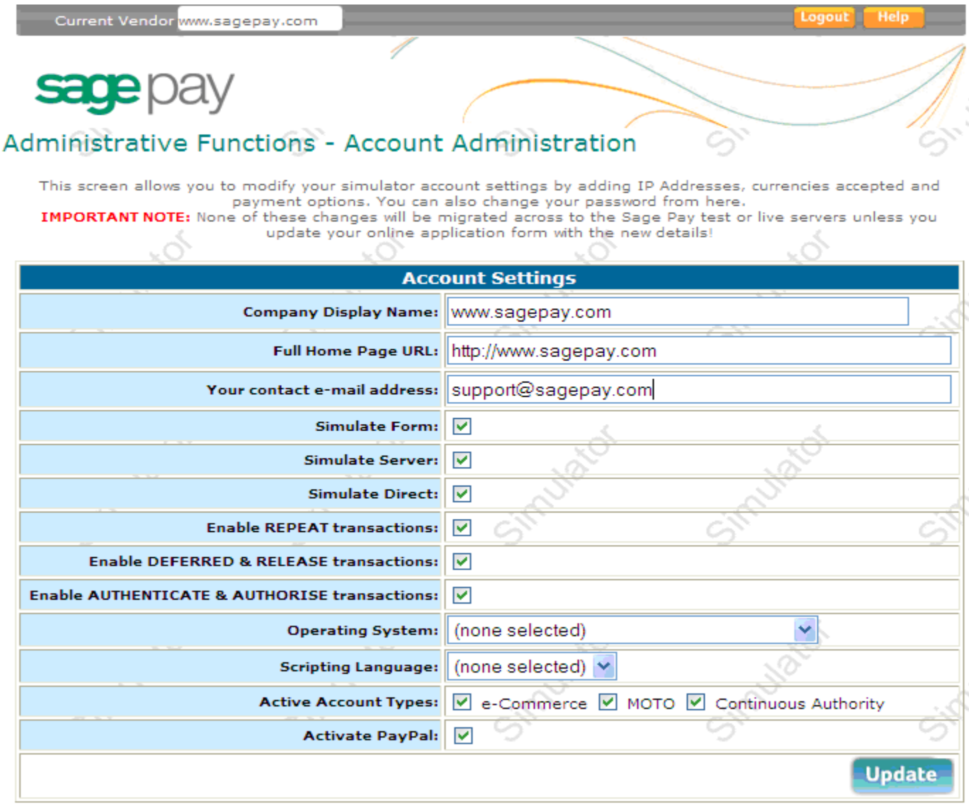
Your Terminal URL code (normally a page called `3DCallback` in the kits) should be modified to store the result fields in your database (as you did for your transaction registration code in section 3 above), including the **3DSecureStatus** field and, for 3D-Authenticated transactions, the **CAVV** field (the unique signature for a validated 3D-Secure transaction).

You can then direct your customer to the relevant completion page, depending on the Status of the transaction. Like non-authenticated transactions, a Status of OK should redirect the user to a success page, and ERROR, NOTAUTHED, REJECTED, MALFORMED or INVALID to various error handling pages.

5: Direct PayPal transactions

You should **ONLY** begin to test your PayPal integration on the Simulator once you are happy that your site can correctly send and process the messages exchanged between your site and ours for a standard Direct transaction.

Log into the Simulator, click the 'Account' button, and ensure 'Activate PayPal' has been selected in the Account Settings section.



Administrative Functions - Account Administration

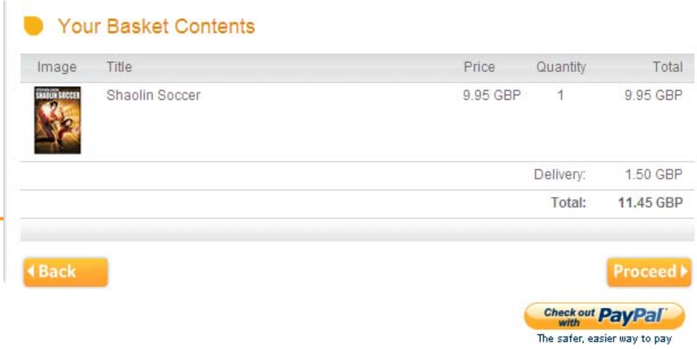
This screen allows you to modify your simulator account settings by adding IP Addresses, currencies accepted and payment options. You can also change your password from here.
IMPORTANT NOTE: None of these changes will be migrated across to the Sage Pay test or live servers unless you update your online application form with the new details!

Account Settings

Company Display Name:	www.sagepay.com
Full Home Page URL:	http://www.sagepay.com
Your contact e-mail address:	support@sagepay.com
Simulate Form:	☑
Simulate Server:	☑
Simulate Direct:	☑
Enable REPEAT transactions:	☑
Enable DEFERRED & RELEASE transactions:	☑
Enable AUTHENTICATE & AUTHORISE transactions:	☑
Operating System:	(none selected)
Scripting Language:	(none selected)
Active Account Types:	☑ e-Commerce ☑ MOTO ☑ Continuous Authority
Activate PayPal:	☑

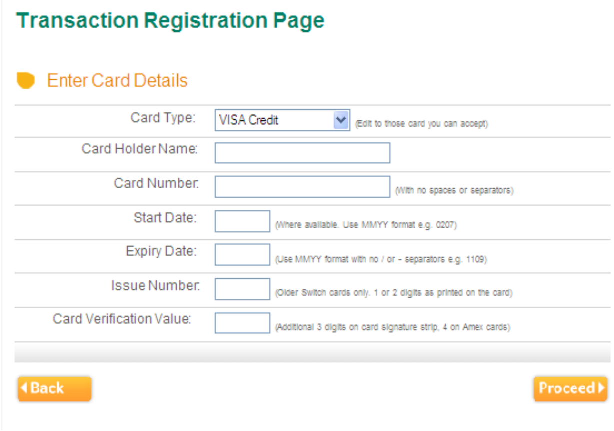
Click the Update button and log out.

Browse to your Direct integration kit and proceed to the checkout page. The kits present both Express Checkout and Mark options:



Your Basket Contents

Image	Title	Price	Quantity	Total
	Shaolin Soccer	9.95 GBP	1	9.95 GBP
			Delivery:	1.50 GBP
			Total:	11.45 GBP



Transaction Registration Page

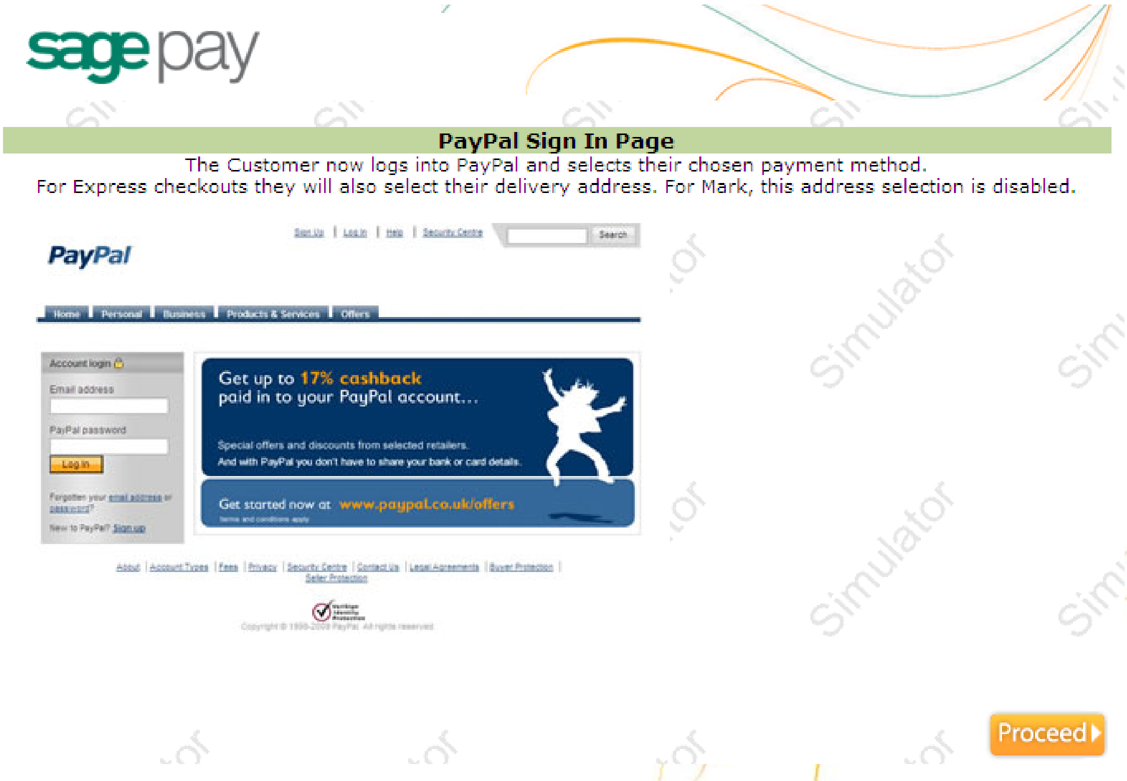
Enter Card Details

Card Type:	VISA Credit	(Edit to those card you can accept)
Card Holder Name:		
Card Number:		(With no spaces or separators)
Start Date:		(Where available. Use MMYY format e.g. 0207)
Expiry Date:		(Use MMYY format with no / or - separators e.g. 1109)
Issue Number:		(Older Switch cards only. 1 or 2 digits as printed on the card)
Card Verification Value:		(Additional 3 digits on card signature strip, 4 on Amex cards)

Whichever option you choose, either Express Checkout or Mark, you should send the Transaction Registration post to the Simulator with the CardType set to PAYPAL.

A Status of “PPREDIRECT” and a simulated PayPalRedirectURL will be provided in the Sage Pay response to your Transaction Registration Post (see below):

Your code should store the VPSTxId and redirect the customer’s browser to the simulated PayPal Sign In page:

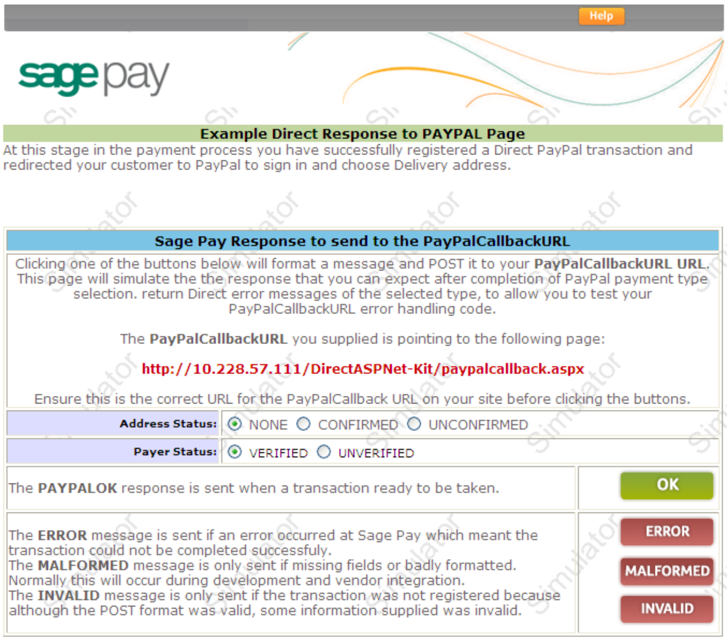


PayPal Sign In Page

The Customer now logs into PayPal and selects their chosen payment method.
For Express checkouts they will also select their delivery address. For Mark, this address selection is disabled.

sagepay
Simulator
PayPal
Sign Up | Log In | Help | Security Centre
Search
Home
Personal
Business
Products & Services
Offers
Account login
Email address
PayPal password
Log In
Forgotten your email address or password?
New to PayPal? Sign up
Get up to 17% cashback paid in to your PayPal account...
Special offers and discounts from selected retailers.
And with PayPal you don't have to share your bank or card details.
Get started now at www.paypal.co.uk/offers
Terms and conditions apply
About | Account Types | Fees | Privacy | Security Centre | Contact Us | Legal Agreements | Buyer Protection | Seller Protection
Copyright © 1999-2009 PayPal. All rights reserved.
Proceed

Click ‘Proceed’ on the PayPal Sign In Page to reach the screen below.



Example Direct Response to PAYPAL Page

At this stage in the payment process you have successfully registered a Direct PayPal transaction and redirected your customer to PayPal to sign in and choose Delivery address.

Sage Pay Response to send to the PayPalCallbackURL

Clicking one of the buttons below will format a message and POST it to your **PayPalCallbackURL URL**. This page will simulate the the response that you can expect after completion of PayPal payment type selection. return Direct error messages of the selected type, to allow you to test your PayPalCallbackURL error handling code.

The **PayPalCallbackURL** you supplied is pointing to the following page:

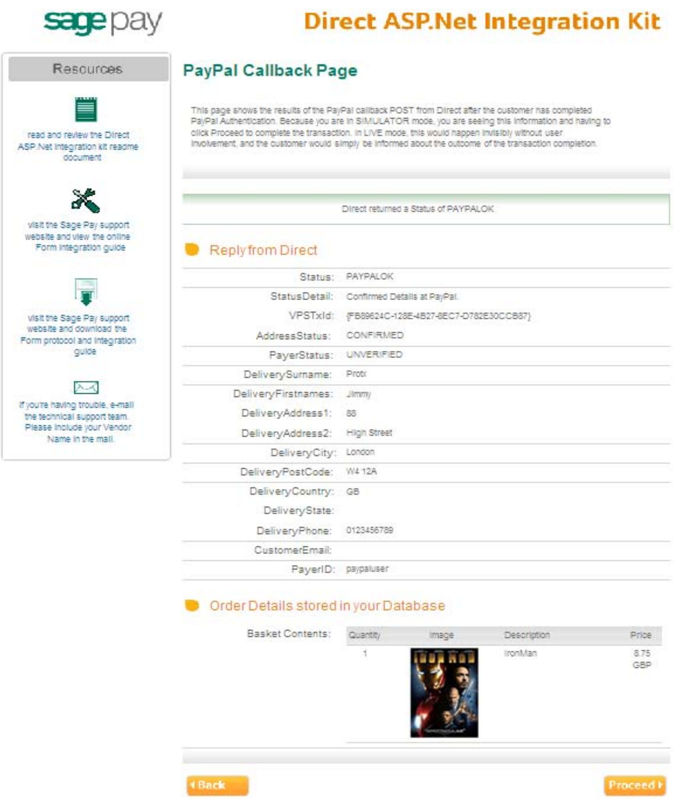
http://10.228.57.111/DirectASPNet-Kit/paypalcallback.aspx

Ensure this is the correct URL for the PayPalCallback URL on your site before clicking the buttons.

Address Status:	● NONE ○ CONFIRMED ○ UNCONFIRMED
Payer Status:	● VERIFIED ○ UNVERIFIED
The **PAYPALOK** response is sent when a transaction ready to be taken.	OK
The **ERROR** message is sent if an error occurred at Sage Pay which meant the transaction could not be completed successfuly. The **MALFORMED** message is only sent if missing fields or badly formatted. Normally this will occur during development and vendor integration. The **INVALID** message is only sent if the transaction was not registered because although the POST format was valid, some information supplied was invalid.	ERROR MALFORMED INVALID

On this screen, you can choose the message which is returned to your PayPalCallbackURL upon completion of a PayPal payment. For now, we recommend you to select OK to simulate a PAYPALOK response (see Appendix A7 below). Click 'OK'.

The default PayPalCallbackURL code in the kits shows the results of the PayPal Callback Post from Direct after the customer has completed PayPal Authentication, and gives you the option of rejecting the transaction based on the results returned.



read and review the Direct ASP.Net integration kit readme document

visit the Sage Pay support website and view the online Form integration guide

visit the Sage Pay support website and download the Form protocol and integration guide

If you're having trouble, e-mail the technical support team. Please include your Vendor Name in the mail.

PayPal Callback Page

This page shows the results of the PayPal callback POST from Direct after the customer has completed PayPal Authentication. Because you are in SIMULATOR mode, you are seeing this information and having to click Proceed to complete the transaction. In LIVE mode, this would happen invisibly without user involvement, and the customer would simply be informed about the outcome of the transaction completion.

Direct returned a Status of PAYPALOK

Reply from Direct

Status:	PAYPALOK
StatusDetail:	Confirmed Details at PayPal.
VPSTxId:	{FB89624C-128E-4B27-8EC7-D782E30CCB87}
AddressStatus:	CONFIRMED
PayerStatus:	UNVERIFIED
DeliverySurname:	Protx
DeliveryFirstnames:	Jimmy
DeliveryAddress1:	88
DeliveryAddress2:	High Street
DeliveryCity:	London
DeliveryPostCode:	W4 12A
DeliveryCountry:	GB
DeliveryState:	
DeliveryPhone:	0123456789
CustomerEmail:	
PayerID:	paypaluser

Order Details stored in your Database

Basket Contents:

Quantity	Image	Description	Price
1		IronMan	8.75 GBP

This page does not necessarily have to be displayed to your customer; you can choose to either handle the response behind the scenes, or display a page to the shopper confirming the details of the transaction. If required, this page also allows you to change the amount sent in the original POST in Appendix A1 by +/- 15% of the original value (for example, if the delivery price changes as a result of the address selected).

In the example above, the AddressStatus returned was CONFIRMED and the PayerStatus returned was VERIFIED. As the customer has successfully completed the PayPal verification process to help establish the shopper's identity, and their address has been reviewed by PayPal and found highly likely to be that of the User to which it is associated, this is a strong indication that this is a genuine shopper. If you wish to proceed with the transaction, you send a POST to the PayPal Completion URL with a value of **YES** in the **Accept** field (see Appendix A8).

If the address was not confirmed, and the payer not verified, for example, you may not wish continue without **PayPal Seller Protection***. If you do NOT wish to proceed, you would still need to send a POST to the Sage Pay servers to complete the transaction, but enter a value of **NO** in the **Accept** field to cancel the transaction (see Appendix A8).

* **PayPal's Seller Protection Policy** can protect sellers from claims, chargebacks and reversals. PayPal will reimburse you for a specified amount if you meet the requirements set out in their Terms and Conditions of the User Agreement. Please visit the link below for further information about Seller Protection:
https://www.paypal.com/uk/cgi-bin/webscr?cmd=p/gen/ua/policy_spp-outside#spp-policy

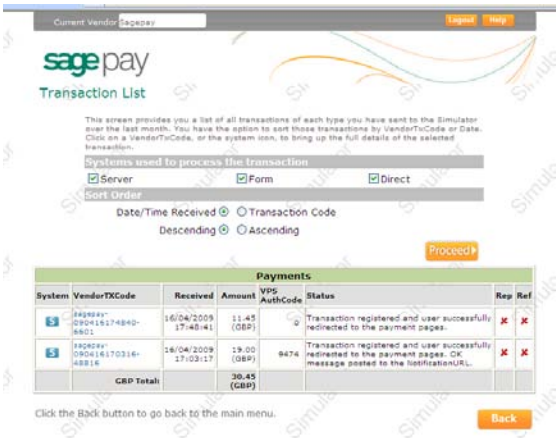
NB: Sage Pay provides example code in several scripting languages to aid developers integrate our Direct product with PayPal. Within the kits there is a file called paypalCallback. This file provides a basic working example of completing a successful PayPal transaction. By default, the Amount of the transaction is unchanged from the original value, and the value returned in the Accept field is always "YES", regardless of the results returned to you in Appendix A7. You can amend your own code and use the results returned to determine whether you wish to proceed with the PayPal transaction.

You can create a Simulated PayPal Completion message that, when POSTed to the Simulator PayPal Completion page, will generate the Direct result of your choosing. By sending the Completion POST with Accept=YES, a Status of OK will be returned in the final POST to your servers. A value of Accept=NO will return a Status of NOTAUTHED.

When you receive this final response from Direct, (see Appendix A2), you should redirect your customer to the relevant completion page on your site, depending on the Status of the transaction. Like standard transactions, a Status of OK should redirect the user to a success page, and ERROR, NOTAUTHED, REJECTED, MALFORMED or INVALID to various error handling pages.

6: Examining your transactions

The 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.



Transaction List

This screen provides you a list of all transactions of each type you have sent to the Simulator over the last month. You have the option to sort those transactions by VendorTxCode or Date. Click on a VendorTxCode, or the system icon, to bring up the full details of the selected transaction.

Systems used to process the transaction

☑ Server ☑ Form ☑ Direct

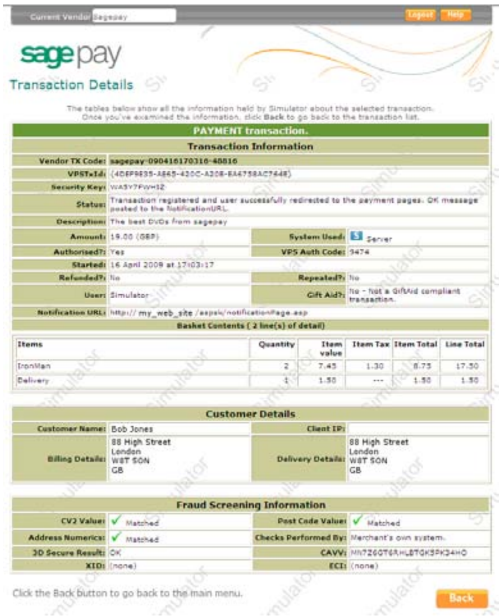
Sort Order

Date/Time Received ● ○ Transaction Code

Descending ● ○ Ascending

Payments							
System	VendorTXCode	Received	Amount	VPS AuthCode	Status	Rep	Ref
S	sagepay-090416174840-6601	16/04/2009 17:48:41	11.45 (GBP)	0	Transaction registered and user successfully redirected to the payment pages.	✗	✗
S	sagepay-090416170316-48816	16/04/2009 17:03:17	19.00 (GBP)	9474	Transaction registered and user successfully redirected to the payment pages. OK message posted to the NotificationURL.	✗	✗
	GBP Total:		30.45 (GBP)				

Click the Back button to go back to the main menu.



Transaction Details

The tables below show all the information held by Simulator about the selected transaction. Once you've examined the information, click **Back** to go back to the transaction list.

PAYMENT transaction.

Transaction Information			
Vendor TX Code:	sagepay-090416170316-48816		
VPSTxId:	{4DEF9E35-A865-420C-A208-EA4758AC744B}		
Security Key:	WA5Y7FWH1Z		
Status:	Transaction registered and user successfully redirected to the payment pages. OK message posted to the NotificationURL.		
Description:	The best DVDs from sagepay		
Amount:	19.00 (GBP)	System Used:	S Server
Authorised?:	Yes	VPS Auth Code:	9474
Started:	16 April 2009 at 17:03:17		
Refunded?:	No	Repeated?:	No
User:	Simulator	Gift Aid?:	No - Not a GiftAid compliant transaction.
Notification URL:	http:// my_web_site /aspsk/notificationPage.asp		

Basket Contents (2 line(s) of detail)

Items	Quantity	Item value	Item Tax	Item Total	Line Total
IronMan	2	7.45	1.30	8.75	17.50
Delivery	1	1.50	---	1.50	1.50

Customer Details			
Customer Name:	Bob Jones	Client IP:	
Billing Details:	88 High Street London W8T 5ON GB	Delivery Details:	88 High Street London W8T 5ON GB

Fraud Screening Information			
CV2 Value:	✓ Matched	Post Code Value:	✓ Matched
Address Numerics:	✓ Matched	Checks Performed By:	Merchant's own system.
3D Secure Result:	OK	CAVV:	MN7Z6OT6RHLBTGK5PK34HO
XID:	(none)	ECI:	(none)

Click the Back button to go back to the main menu.

You can also see from this screen which transactions have been subsequently refunded or used as the basis for repeat payments (see 7 below).

Once your site can handle all Direct status types, on both your transaction registration and 3D-Secure Terminal URL pages, then you've completed your basic Direct integration and can move on to testing your site against the real Direct, firstly on the Test Server (see the next main section). If, however, you wish to link in additional processes, such as Refunds or Repeats, or the ability to Release or Abort Deferred transactions, you should continue with step 7 below.

7: Additional Transaction Types

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. Direct 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 Direct will reply with 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 for that transaction).

NB: For PayPal transactions, you can use the Authenticate and Authorise Payment Type but the transaction will only ever be **REGISTERED** (because the transaction will never be 3D Secured). 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.

If, however, the card *is* part of the 3D-Secure scheme, Direct will reply with a Status of **3DAUTH** along with the MD, PAReq and ACSURL. You must then redirect the customer 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 your TermUrl along with the PARes and MD. These you'll forward to the Direct 3D Callback page and Sage Pay will decode the response for you.

If the customer has not passed authentication, your rule base is consulted to check if they can proceed for authorisation anyway. If not, Direct replies with a Status of **REJECTED**. If the customer failed authentication but can proceed, Direct replies with a **REGISTERED** Status. If the user passed authentication with their bank and a CAVV/UCAF value is returned, Direct sends 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 (see the "Server and Direct Shared Protocols" document for details of these messages).

To charge the customer when you are ready to fulfil the order, your site will need to send an **AUTHORISE** request. You can Authorise any amount up to 115% of the value of the original 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 greater 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 send a **CANCEL** message to our Server to archive away the Authentication and prevent any further Authorisations being made against the card. This happens automatically after 90 days.

Both AUTHORISE and CANCEL operations can also be performed in the *My Sage Pay* area.

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 send a **VOID** message to our servers to prevent the transaction ever being settled (see the "Server and Direct Shared Protocols" document for more detail), thus saving you your transaction charges and the customer from ever being charged. You can also VOID transactions through the *My Sage Pay* Admin 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 send a **REFUND** message to our servers, or use the *My Sage Pay* Admin screens to do the same.

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. Again, the

REFUND protocol can be found in the "Server and Direct Shared Protocols" document.

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

The Sage Pay Simulator and Additional Transaction Types

The Sage Pay Simulator can handle all the additional transaction types discussed above. It will accept PAYMENT, AUTHENTICATE and DEFERRED transactions at the registration stage, plus it has services that emulate those of the real servers when you send REFUND, RELEASE, ABORT, REPEAT, AUTHORISE, CANCEL and VOID messages to it.

The additional transaction types, however, **do not** have a user configurable interface associated with them. By default they are all set to Automatic mode, so they will respond with an OK unless the data you send would generate a MALFORMED or INVALID response.

For information regarding registering additional transaction types using HTTPS POSTS, please refer to the Server and Direct Shared Protocols Guide, which can be obtained from the developer's area on our website:

www.sagepay.com/help/downloads

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. 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 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* Admin 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 the Server integrated payment method rather than the Simulator. In many kits this is done simply by changing the *strConnectTo* string in the includes file 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/vspdirect-register.vsp

For other transaction types, the final vspdirect-register.vsp section would be changed to refund.vsp, release.vsp, void.vsp etc. Please refer to the Server and Direct Shared Protocols Guide.

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 over the page. 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.

Card Type	Card Number	Issue	CV2	Address	PostCode
Visa (VISA)	4929000000006		123	88	412
MasterCard (MC)	5404000000000001		123	88	412
Visa Debit / Delta (DELTA)	4462000000000003		123	88	412
UK Maestro / International Maestro (MAESTRO)	5641820000000005 300000000000000004	01 N/A	123 123	88 88	412 412
American Express (AMEX)	374200000000004		123	88	412
Visa Electron (UKE)	4917300000000008		123	88	412
JCB (JCB)	3569990000000009		123	88	412
Diner's Club (DINERS)	36000000000008		123	88	412
Laser (LASER)	6304990000000000044		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 Simulator process because it creates real PAReq and PARes messages. It does not talk to the real Visa and MasterCard systems though, so no live authentications can occur.

At the Simulated Authentication screens, to successfully authenticate the transaction, enter "**password**" (without the quotes) into the password box. Any other phrase will fail the authentication.

At the 3D-secure callback stage you'll need to change your POST to go to:

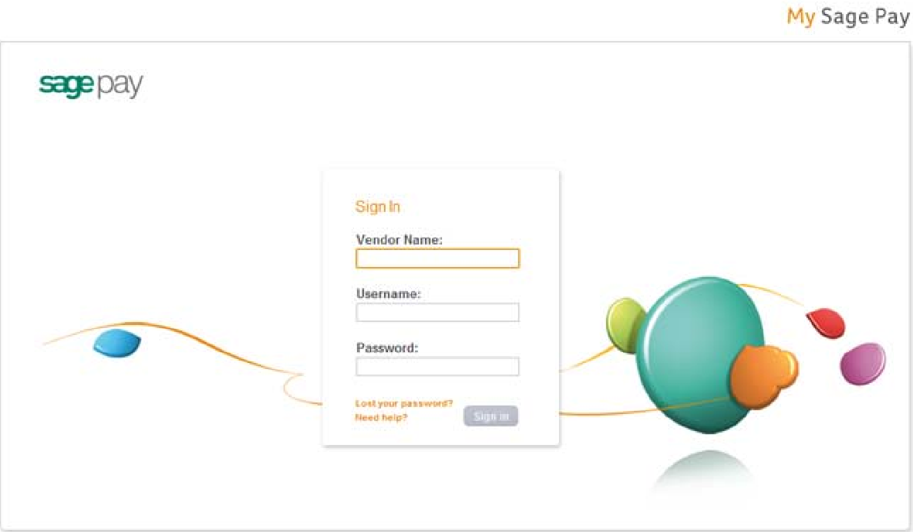
https://test.sagepay.com/gateway/service/direct3dcallback.vsp

Once you've checked you can process an end-to-end transaction and, where appropriate, can successfully process 3D-Authentication, and tested any additional transaction types you have set up (such as Refunds and Releases) then you are almost ready to go live. Before doing so, however, you should log into your unique *My Sage Pay* account on the test servers to view your transactions and familiarise yourself with the interface.

The Test Server My Sage Pay interface

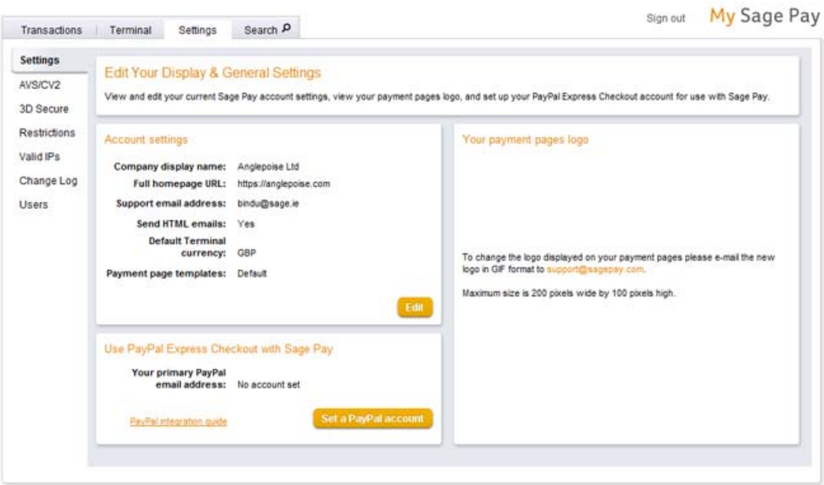
A Test Server version of the My Sage Pay interface is available to you whilst using your test account to view your transactions, refund payments, release deferred payments, void transactions etc. You should familiarise yourself with this system on the Test Server before you go live so you know how to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

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



When you log in to the *My Sage Pay* screens 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 Admin password as supplied to you by Sage Pay when your test account was set up.
- Click **Login** and you will see the settings section (below).



Edit Your Display & General Settings

View and edit your current Sage Pay account settings, view your payment pages logo, and set up your PayPal Express Checkout account for use with Sage Pay.

Account settings

Company display name: Anglepoise Ltd
Full homepage URL: https://anglepoise.com
Support email address: bindu@sage.ie
Send HTML emails: Yes
Default Terminal currency: GBP
Payment page templates: Default

Use PayPal Express Checkout with Sage Pay

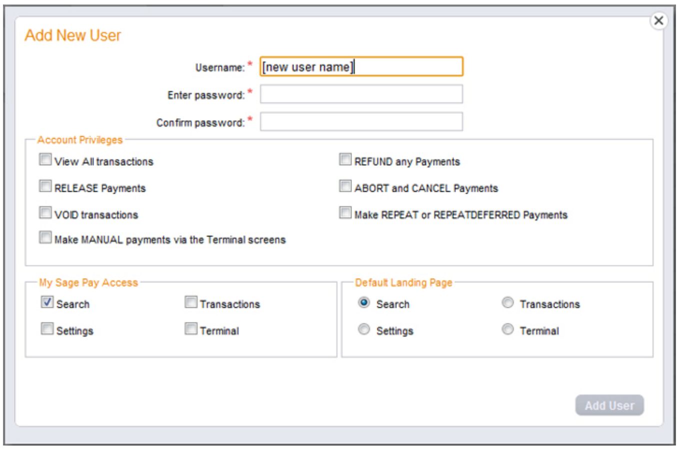
Your primary PayPal email address: No account set

Your payment pages logo

To change the logo displayed on your payment pages please e-mail the new logo in GIF format to support@sagepay.com.

Maximum size is 200 pixels wide by 100 pixels high.

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.



Add New User

Username: * [new user name]

Enter password: *

Confirm password: *

Account Privileges

□ View All transactions
□ REFUND any Payments
□ RELEASE Payments
□ ABORT and CANCEL Payments
□ VOID transactions
□ Make REPEAT or REPEATDEFERRED Payments
□ Make MANUAL payments via the Terminal screens

My Sage Pay Access

☑ Search
□ Transactions
□ Settings
□ Terminal

Default Landing Page

● Search
○ Transactions
○ Settings
○ Terminal

To use those functions, and to protect the administrator account, you need to create new users for yourself and others. Click on the user tab on the left to create a new users and you will be presented the following screen (right).

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

Now click the sign out button in the top right hand corner 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 e-mail 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 e-mail to the above and request a password retrieval link, stating the Vendor Name and Merchant Number of the account.

Detailed information on using the My sage Pay admin area can be found in the online help centre (www.sagepay.com/help) or you can watch a video demo available in the demo area (www.sagepay.com/help/demos). Play with the system until you are comfortable with it though; you cannot inadvertently charge anyone or damage anything whilst on the test server.

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 and you will start to be billed for using our gateway.

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:

www.sagepay.com/help/faq/processes_to_go_live_how_to_start_accepting_payments_from_your_customers

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/vspdirect-register.vsp

(for other transaction types, the vspdirect-register.vsp section would be changed to refund.vsp, void.vsp, release.vsp etc.)

The 3D-Secure Callback URL becomes:

https://live.sagepay.com/gateway/service/direct3dcallback.vsp

The PayPal Completion URL becomes:

https://live.sagepay.com/gateway/service/complete.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.

Finally 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 the Sage Pay.

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 Tellmemore@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 **0845-111-44 55**, 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 www.sagepay.com/system_monitor.asp or our system twitter feed: @System_SagePay

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

Appendix A - The Sage Pay Direct 2.23 Protocol

This section details the Sage Pay Direct Protocol. It details the contents of the POSTs and responses, 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:

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

This is performed via a **HTTPS POST** request, sent to the initial Direct Payment URL. The details should be URL encoded Name=Value fields separated by '&' characters.

Request format (continued overleaf)

Name	Format	Values	Comments
VPSProtocol	Numeric. Fixed 4 characters.	**2.23** in this release	Default or incorrect value is taken to be 2.23.
TxType	Alphabetic Max 15 characters.	**PAYMENT**, **DEFERRED** or **AUTHENTICATE** ONLY	See companion document "Server and Direct Shared Protocols" other transaction types (such as Refund, Releases, Aborts and Repeats). The value should be in capital letters.
Vendor	Alphanumeric Max 15 characters.	Your unique 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.
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	Alphabetic 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 for you reference in the My Sage Pay screens
CardHolder	Alphabetic Max 50 characters	The card holder's name. Aa á ␣ /\ & . - '	This should be the name displayed on the card. **Not required if CardType=PAYPAL**
CardNumber	Numeric Max 20 characters	The credit or debit card number with no spaces. 0-9	The full card number is required. **Not required if CardType=PAYPAL**

Request format (continued overleaf)

Optional: StartDate	Numeric 4 characters	The Start date (required for some Maestro and Amex) in **MMYY** format. 0-9	The start date MUST be in MMYY format i.e. 0699 for June 1999. No / or – characters should be included. **Not required if CardType=PAYPAL**
ExpiryDate	Numeric 4 characters	The Expiry date (required for ALL cards) in **MMYY** format. 0-9	The expiry date MUST be in MMYY format i.e. 1206 for December 2006. No / or – characters should be included. **Not required if CardType=PAYPAL**
Optional: IssueNumber	Numeric Max 2 characters	The card Issue Number (some Maestro cards only) 0-9	The issue number MUST be entered EXACTLY as it appears on the card. e.g. some cards have Issue Number “4”, others have “04”. **Not required if CardType=PAYPAL**
Optional: CV2	Numeric Max 4 characters	The extra security 3 digits on the signature strip of the card, or the extra 4 digits on the front for American Express Cards 0-9	**NB: If AVS/CV2 is ON for your account this field becomes compulsory. Not required if CardType=PAYPAL**
CardType	Alphabetic Max 15 characters	**VISA, MC, DELTA, MAESTRO, UKE, AMEX, DC, JCB, LASER, PAYPAL** NB: “SWITCH” is still accepted for UK Maestro but you should now use “MAESTRO” for both UK and Int. cards	**MC** is MasterCard. **UKE** is Visa Electron. **MAESTRO** should be used for both UK and International Maestro. **AMEX**, **DC** (DINERS) and **PAYPAL** can only be accepted if you have additional merchant accounts with those acquirers.
BillingSurname	Alphabetic Max 20 characters	Customer’s surname Aa á ␣ /\ & . - '	In Protocol 2.23, unlike previous protocols, the Billingxxxxx columns are compulsory. N.B: All fields must contain a value including the Post Code field even if the customer does not have a post code. Providing a blank field will cause an error.
BillingFirstnames	Alphabetic 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 ␣ -	

Request format (continued overleaf)

BillingCountry	Alphabetic Max 2 characters	ISO 3166-1 country code of the cardholder's billing address ISO3166	
Optional*: BillingState	Alphabetic 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	Alphabetic Max 20 characters	Customer's surname Aa á ␣ /\ & . - '	In Protocol 2.23, unlike previous protocols, the Deliveryxxxx columns are compulsory. N.B: All fields must contain a value including the Post Code field even if the customer does not have a post code. Providing a blank field will cause an error.
DeliveryFirstnames	Alphabetic 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	Alphabetic Max 2 characters	ISO 3166-1 country code of the cardholder's delivery address ISO3166	
Optional*: DeliveryState	Alphabetic Max 2 characters	State code for US customers only* US	
Optional: DeliveryPhone	Alphanumeric Max 20 characters	Phone number at delivery address 0-9 - Aa + ␣ ()	

Request format (continued overleaf)

Optional: PayPalCallbackURL	Alphanumeric Max 255 characters	Full qualified domain name of the URL to which customers are redirected upon completion of a PayPal transaction RFC1738	Must begin http:// or https:// **Only required if the CardType=PAYPAL.**
Optional: CustomerEMail	Alphanumeric Max 255 characters	The customer's e-mail address. NOTE: If you wish to use multiple e-mail addresses, you should add them using the **:** (colon) character as a separator. e.g. me@mail1.com:me@mail2.com RFC532n	The current version of Direct does not send confirmation e-mails to the customer. This field is provided for your records only.
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: GiftAidPayment	Flag	**0** = This transaction is not a Gift Aid charitable donation (default) **1** = This payment is a Gift Aid charitable donation and the customer has AGREED to donate the tax.	Only of use if your vendor account is Gift Aid enabled. Setting this field means the customer has ticked a box on your site to indicate they wish to donate the tax.
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: ClientIPAddress	Numeric Max 15 characters	The IP address of the client connecting to your server making the payment. 0-9 .	This should be a full IP address which you can obtain from your server scripts. We will attempt to Geolocate the IP address in your reports and fraud screening.
Optional (3D-Secure only): 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 only (if your account is 3D-enabled) and apply rules for authorisation. **2** = Do not perform 3D-Secure checks for this transaction only and always authorise. **3** = Force 3D-Secure checks for this transaction (if your account is 3D-enabled) 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. NB: If 3D Secure is ON for your account this field becomes compulsory. **This field is ignored for PAYPAL transactions**

Request format (continued…)

Optional: AccountType	Alphanumeric 1 character	**E** = Use the e-commerce merchant account (default). **C =** Use the continuous authority merchant account (if present). **M =** Use the mail order, telephone order account (if present).	This optional flag is used to tell the SAGE PAY System which merchant account to use. If omitted, the system will use E, then M, then C by default. **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:

- The line breaks above are included for readability only. No line breaks are needed; the only separators should be the colons.
- 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: Sage Pay Response to the Registration or Callback POSTs

This is the plain text response part of the POST originated by your servers in A1 (or A4 for 3D-Secure transactions). Encoding will be as Name=Value fields separated by carriage-return-linefeeds (CRLF).

Response format (continued overleaf)

Name	Format	Values	Comments
VPSProtocol	Numeric Fixed 4 characters.	Version number of the protocol of the system. This release will return **2.23**	This will match the protocol version supplied in A1.
Status	Alphanumeric Max 15 characters.	**OK** – The transaction was authorised by the bank. **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. **NOTAUTHED** – The transaction was not authorised by the acquiring bank. No funds could be taken from the card. A NOTAUTHED status is also returned for PayPal transactions in response to the PayPal Completion Post (if Accept=NO was sent to complete PayPal transaction, see Appendix A8). **REJECTED** – The SAGE PAY System rejected the transaction because of the rules you have set on your account. **3DAUTH** – The customer needs to be directed to their card issuer for 3D-Authentication. **GO TO APPENDIX A3**. **PPREDIRECT** – Customer needs to be redirected to PayPal **GO TO APPENDIX A6** **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** – An error occurred at Sage Pay which meant the transaction could not be completed successfully.	If the status is not **OK**, the **StatusDetail** field will give more information about the status. Please notify Sage Pay if a Status report of **ERROR** is seen, together with your VendorTxCode and the StatusDetail text. **3DAUTH** is only returned if 3D-Authentication is available on your account AND the directory services have issued a URL to which you can progress. A Status of 3DAUTH only returns the StatusDetail, 3DSecureStatus, MD, ACSURL and PAReq fields. The other fields are returned with other Status codes only. See Appendix 3. **AUTHENTICATED** and **REGISTERED** statuses are only returned if the TxType is **AUTHENTICATE**.

Response format (continued overleaf)

StatusDetail	Alphanumeric Max 255 characters	Human-readable text providing extra detail for the Status message.	Always check StatusDetail if the Status is not **OK**
VPSTxId	Alphanumeric 38 characters	Sage Pay ID to uniquely identify the Transaction on our system.	You should store this value and quote it to us if you have a query about the transaction. Not present when Status is **3DAUTH.**
SecurityKey	Alphanumeric 10 characters	Security key which Sage Pay uses to generate an MD5 Hash to sign the transaction.	Should be kept secret from the Customer but stored in your database. Not present when Status is **3DAUTH or PPREDIRECT**
TxAuthNo	Numeric Long integer	The Sage Pay authorisation code (also called **VPSAuthCode**) for this transaction.	Only present if Status is **OK**.
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 on your account 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 **3DAUTH, AUTHENTICATED** or **REGISTERED,** or **PPREDIRECT**
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 **3DAUTH, AUTHENTICATED** or **REGISTERED,** or **PPREDIRECT**
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 **3DAUTH, AUTHENTICATED** or **REGISTERED,** or **PPREDIRECT**
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 **3DAUTH, AUTHENTICATED** or **REGISTERED,** or **PPREDIRECT**

Response format (continued…)

3DSecureStatus	Alphanumeric Max 20 characters	**OK –** The 3D-Authentication step completed successfully. If the Status field is OK too, then this indicates that the authorized transaction was also 3D-authenticated and a CAVV will be returned. Liability shift occurs. **NOAUTH –** This means the card is not in the 3D-Secure scheme. **CANTAUTH -** This normally means the card Issuer is not part of the scheme. **NOTAUTHED** – The cardholder failed to authenticate themselves with their Issuing Bank. **ATTEMPTONLY –** The cardholder attempted to authenticate themselves but the process did not complete. A CAVV is returned, therefore a liability shift may occur for non-Maestro cards. Check your Merchant Agreement. **NOTCHECKED** - No 3D Authentication was attempted for this transaction. Always returned if 3D-Secure is not active on your account. **INCOMPLETE** – 3D-Secure authentication was unable to complete (normally at the card issuer site). No authentication occurred. **MALFORMED,INVALID,ERROR** – These statuses indicate a problem with creating or receiving the 3D-Secure data. These should not occur on the live environment.	This field holds the results of the 3D-Authentication steps carried out by Direct and the Sage Pay MPI. If 3D-Secure is not active on your account, this field will always contain the value **NOTCHECKED**.
CAVV	Alphanumeric Max 32 characters	The encoded result code from the 3D-Secure checks. Holds the Visa CAVV or the MasterCard UCAF depending on the card type used in the transaction.	Only present if the 3DSecureStatus field is **OK** AND the Status field is **OK**

A3: Sage Pay Response to the Registration POST (3D Secure)

If 3D-Authentication is available on your account and the Card AND the Card Issuer are (or can be) part of the scheme, this is the plain text response part of the POST originated by your servers in A1. Encoding will be as Name=Value fields separated by carriage-return-linefeeds (CRLF).

Response format:

Name	Format	Values	Comments
Status	Alphanumeric Max 15 characters.	**3DAUTH** – The customer needs to be directed to their card issuer for 3D-Authentication.	**3DAUTH** is only returned if 3D-Authentication is available on your account AND the directory services have issued a URL to which you can progress. A Status of 3DAUTH only returns the StatusDetail, 3DSecureStatus, MD, ACSURL and PAReq fields.
StatusDetail	Alphanumeric Max 255 characters	Human-readable text providing extra detail for the Status message.	Always check StatusDetail if the Status is not **OK**
3DSecureStatus	Alphanumeric Max 20 characters	**OK –** If the Status field is 3DAUTH, this means the card is part of the scheme.	If a Status of 3DAUTH is returned at this stage, the only value you will receive for the 3DSecureStatus is OK.
MD	Alphanumeric Max 35 characters	A unique reference for the 3D-Authentication attempt.	Only present if the Status field is **3DAUTH**.
ACSURL	Alphanumeric Max Unlimited, but practical limit should be 7,500 characters	A fully qualified URL that points to the 3D-Authentication system at the Cardholder's Issuing Bank.	Only present if the Status field is **3DAUTH**.
PAReq	Alphanumeric Max Unlimited, but practical limit should be 7,500 characters	A Base64 encoded, encrypted message to be passed to the Issuing Bank as part of the 3D-Authentication.	Only present if the Status field is **3DAUTH**. NOTE: When forwarding this value to the ACSURL, pass it in a field called **PaReq** (note the lower case a). This avoids issues with case sensitive ACSURL code.

A4: 3D-Authentication Results POST from your Terminal URL to Sage Pay.

This is performed via a **HTTPS POST** request, sent to the Direct 3D-Secure Callback URL. The details should be URL encoded Name=Value fields separated by '&' characters.

Request format

Name	Format	Values	Comments
MD	Alphanumeric Max 35 characters	A unique reference for the 3D-Authentication attempt. Aa 0-9	This will match the MD value passed back to your site in response to your transaction registration POST.
PARes	Alphanumeric Max Unlimited, but practical limit should be 7,500 characters	A Base64 encoded, encrypted message sent back by Issuing Bank to your Terminal URL at the end of the 3D-Authentication process. You will receive this value back from the Issuing Bank in a field called PaRes (lower case a"), but should be passed to Sage Pay as PARes. BASE64	This field must be passed back to Direct along with the MD field to allow the Sage Pay MPI to decode the result.

A5: Response to 3D Callback POSTs

The response from the 3D Callback service is identical to that of the initial registration POST. See section A2 above.

A6: Sage Pay Response to the Registration POST (for PayPal transactions)

If you supplied PayPal as a CardType in A1 above and PayPal is active on your account, this response is returned from the server. Encoding will be as Name=Value fields separated by carriage-return-linefeeds (CRLF).

Response format

Name	Format	Values	Comments
VPSProtocol	Numeric. Fixed 4 characters.	**2.23** in this release.	Current system version
Status	Alphanumeric Max 15 characters.	**PPREDIRECT** – The customer needs to be redirected to PayPal for authorisation.	
StatusDetail	Alphanumeric Max 255 characters	Will contain the text: "Please redirect the customer to the PayPal URL"	
VPSTxId	Alphanumeric 38 characters	Sage Pay ID to uniquely identify the Transaction on our system.	You should store this value and quote it to us if you have a query about the transaction.
PayPalRedirectURL	Alphanumeric Max 255 characters	A fully qualified domain name URL to which you should redirect the customer. Contain the PayPal token which should not be stripped out.	Forward the customer's browser top frame to this URL to continue with the transaction.

A7: Sage Pay Callback after PayPal Authentication

After redirecting your customer to the PayPalRedirectURL in step A6 above, this message sent to your PayPalCallbackURL, along with the customer, after they have completed their PayPal authentication and payment method selection.

It provides all relevant information about the transaction to allow you to decide if you wish to proceed with the payment (see A8 below). The information will be in the form of URL encoded Name=Value fields separated by '&' characters.

Request format (continued overleaf)

Name	Format	Values	Comments
VPSProtocol	Numeric. Fixed 4 characters.	**2.23** in this release.	Default or incorrect value is taken to be 2.23.
Status	Alphanumeric Max 15 characters.	**PAYPALOK** – The customer has selected a payment type and the transaction is ready to be taken. **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. **ERROR** – An error occurred at Sage Pay which meant the transaction could not be completed successfully.	StatusDetail contains more information if the Status is not PAYPALOK
StatusDetail	Alphanumeric Max 255 characters	Human readable status or error information	
VPSTxId	Alphanumeric 38 characters	Sage Pay ID to uniquely identify the Transaction on our system.	You will need to use this when confirming the PayPal transaction
AddressStatus	Alphanumeric Max 20 characters	Either **NONE, CONFIRMED** or **UNCONFIRMED**	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**	

DeliverySurname	Alphanumeric Max 20 characters	Customer’s surname	These fields form the delivery address selected on the PayPal screens, or provided as part of your registration POST if you are overriding the address (i.e. using a Mark integration rather than Express)
DeliveryFirstnames	Alphanumeric Max 20 characters	Customer’s first names	
DeliveryAddress1	Alphanumeric Max 100 characters	First line of delivery address	
Optional: DeliveryAddress2	Alphanumeric Max 100 characters	Second line of delivery address	
DeliveryCity	Alphanumeric Max 40 characters	City component of the address	
DeliveryPostCode	Alphanumeric Max 10 characters	The Post/Zip code of the Card Holder’s delivery address	
DeliveryCountry	Alphanumeric Max 2 characters	ISO 3166-1 country code of the cardholder’s delivery address	
Optional*: DeliveryState	Alphanumeric Max 2 characters	State code for US customers only	These fields form the delivery address selected on the PayPal screens, or provided as part of your registration POST if you are overriding the address.
Optional: DeliveryPhone	Alphanumeric Max 20 characters	Phone number at delivery address	
CustomerEMail	Alphanumeric Max 255 characters	Customer e-mail address registered at PayPal	
PayerID	Alphanumeric Max 15 characters	Unique PayPal User Reference ID	

A8: Complete PayPal transaction

If you wish to complete a PayPal transaction (whose details are returned in A7 above), you must send a completion POST the Sage Pay servers. **Please note, you will have to contact our support team to enable the COMPLETE TxType on your account.**

This is performed via an **HTTPS POST** request, sent to the Direct PayPal Completion URL. The details should be URL encoded Name=Value fields separated by '&' characters.

The response to this POST is the normal Direct transaction response detailed in A2.

Request format (continued overleaf)

Name	Format	Values	Comments
VPSProtocol	Numeric. Fixed 4 characters.	**2.23** in this release.	Default or incorrect value is taken to be 2.23.
TxType	Alphabetic Max 15 characters.	**COMPLETE** only	
VPSTxId	Alphanumeric 38 characters	Sage Pay ID to uniquely identify the Transaction on our system. { } Aa 0-9 -	You will need to use this when confirming the PayPal transaction
Amount	Numeric. 0.01 to 10,000.00	Amount for the Transaction containing minor digits formatted to 2 decimal places where appropriate. 0-9 , .	The amount can vary from the original POST in A1 by +/- 15% of the original amount (for example, if delivery prices change as a result of the address selected).
Accept	Alphanumeric 3 characters	**YES** or **NO** only	**YES** if you wish to proceed with the PayPal transaction, **NO** if you wish to cancel based on the information returned.

A9: Direct Full URL Summary

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

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

3D-Secure Callback	
Simulator:	https://test.sagepay.com/Simulator/VSPDirectCallback.asp
TEST System:	https://test.sagepay.com/gateway/service/direct3dcallback.vsp
Live System:	https://live.sagepay.com/gateway/service/direct3dcallback.vsp

PayPal Completion	
Simulator:	https://test.sagepay.com/Simulator/paypalcomplete.asp
TEST System:	https://test.sagepay.com/gateway/service/complete.vsp
Live System:	https://live.sagepay.com/gateway/service/complete.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).