SecurePay | Fat Zebra

Direct Post Integration Guide

Document Control

This is a controlled document

Description	Direct Post Integration Guide	**Created By**	SecurePay
Creation Date	20/12/2011	**Date Updated**	03/06/2025
Current Version	1.14		
Changes	1.14	• Updated logo	
	1.13	• Updated that the Merchant Initiated Transaction functionality is not available yet. **We will advise you of the date and provide instructions ahead of the changes.**	

	1.12	• Updated the following sections
		  ○ 3.2.3 – clarified that Account Verification will only be triggered during “Store Only” if the card provided is Visa or Mastercard
		  ○ A.2.2 – removed the paragraphs relating to Account Verification since it’s already covered in 3.2.3	

	1.11	• Added optional field Customer Code • Updated for Merchant Initiated Transactions following Mastercard and Visa scheme mandates. The below changes are applicable to selected acquiring banks (NAB and fiserv FDMSA). ○ Added optional fields Standing Instruction Type Added new fields in the response – Standing Instruction Type, Standing Instruction ID	
	1.10	• Updated supported TLS versions	
	1.9	• Added a new option ‘avtoken’ to EPS_STORETYPE for account verification • Added Account Verification to Glossary • Added optional field EPS_LSI for 3DS2 payments • Removed references to 3D Secure 1	
	1.8	• Include 3DS2 Authentication • Adding EPS_ORDERID for payments authenticated with 3DS2	

Table of Contents

Introduction

What is Direct Post?

Direct Post is a payment service that accepts Customer data posted directly from a form on your web site. Unlike an API, the data is transmitted directly from the Customer's computer to SecurePay. Once the bank has processed the transaction, Direct Post redirects the Customer back to a result page on your web site for order completion and fulfilment.

About this guide

This guide provides technical information about installing and configuring Direct Post within your website.

It is recommended that someone with HTML and web programming experience reads this guide and implements Direct Post. Talk to your web developer if you require technical assistance with programming.

> ***This guide covers the technical requirements of integrating Direct Post into your web site. An understanding of web programming, such as PHP or .NET, is required.***

Features and Benefits

Direct Post benefits Merchants and developers due to the following features:

- Direct Post integrates seamlessly with an existing website both functionally and aesthetically.
- Direct Post can be integrated with any web-based programming language.
- Direct Post is designed to return data to your system following a transaction to enable seamless tracking of payments and orders.

Card Types Accepted

Direct Post accepts the following card types by default via your SecurePay account:

- Visa
- MasterCard

You may also accept the following cards. However, these must be applied for independently via the contacts shown:

- American Express 1300 363 614
- JCB 1300 363 614

Additional Payment Choices

Direct Post also provides the following payment options. Please contact the Payment Choice provider for details on how to sign-up.

- PayPal

If you wish to accept PayPal transactions via Direct Post please follow the steps below:

1. Inform the SecurePay Support team of your intention to use PayPal.
2. If you don't have a Business PayPal Account, establish an account with PayPal.
3. Login to the SecurePay Merchant Login.
4. Navigate to the following location:
   - Click on [Manage] dropdown list and click on [PayPal Settings].
   - Click on [Change Settings] button.
5. Click on the [Retrieve API Credentials] link on the page.
   - Note: A popup window will appear. Please ensure you have popups enabled in your web browser.
6. Login to PayPal using your credentials.
7. Copy and paste the credentials into [Step 8] and close the popup window.
8. Add the following PayPal credentials obtained from [Step 7] to the Merchant Login PayPal settings page:
   - API Username
   - API Password
   - API Signature
9. Add the company logo URL (Optional). The URL must be publicly accessible and must be securely hosted (HTTPS).
10. Save changes.

Note: Once PayPal has been enabled and configured successfully you can view PayPal transactions processed via Direct Post through the SecurePay Merchant Login.

Feedback

Continuous improvement is one of SecurePay's core values. We welcome any feedback you have on our integration guides to help us improve any future changes to our products.

If you wish to leave feedback, please email us at support@securepay.com.au.

How is Direct Post Implemented?

Important Information Before You Begin

The Direct Post Interface is not an API model, it is a browser redirect model.

Credit card numbers must be submitted by the card holder directly to the payment URL's in this document, and not to your own or a third party server, from an HTML form on your website. This is a bank requirement and must be met before live transaction processing can commence.

Technical Overview

Direct Post is an online, secure credit and charge card transaction system that integrates into a web programming environment, such as PHP or .NET, via a three-step process that ensures transaction amount and response integrity.

1. **Generate a Fingerprint**
   A Fingerprint is generated in your website code by a SHA256 hash comprising your SecurePay Merchant ID and transaction password, your payment amount, transaction reference and a timestamp, 3DS2 Order ID (if applicable) and liability shift indicator (if applicable). This value is then presented on your payment form as a hidden field.

2. **Customer Submits Card Details to Direct Post**
   Your customer enters their credit card details on a secure HTML form on your website and submits them directly to Direct Post which in turn securely processes the transaction.

   Note: When accepting card details on your website you will require an SSL certificate. It is your responsibility to obtain and configure the SSL certificate.

3. **Redirect to Result Page**
   Upon completion of the transaction, Direct Post redirects to your result URL and passes result parameters, including a result Fingerprint to protect the transaction result. Your system checks the Fingerprint, updates your database and displays the receipt to the Customer.

Technical Overview for Payment Choice – PayPal

PayPal uses a secure page, hosted by PayPal and presented to your customer as part of the payment authorization.

1. **Generate a Fingerprint**

   A Fingerprint is generated in your website code by a SHA256 hash comprising your SecurePay Merchant ID and transaction password, your payment amount, transaction reference and a timestamp. This value is then presented on your payment form as a hidden field.

2. **Customer Submits Payment Details to Direct Post**

   Your customer selects the payment option of PayPal. Card details are not submitted at this time. Payment details (including amount, transaction reference, and fingerprint) are submitted to Direct Post via hidden fields along with a payment choice of PayPal.

   The PayPal login page is presented to the customer.

3. **Customer logs into PayPal account and submits payment**

   Your customer logs into their PayPal account, confirms shipping and billing details (as required), and selects the payment tender. Your customer will then submit the payment for processing.

4. **Redirect to Result Page**

   Upon completion of the transaction, Direct Post redirects to your result URL and passes result parameters, including a result Fingerprint to protect the transaction result. Your system checks the Fingerprint, updates your database and displays the receipt to the Customer.

Implementation

1. General Information

1.1. TLS Support

Clients integrating with SecurePay must be configured to use TLS v1.2 or TLS v1.3. TLS versions 1.1 and below, as well as all SSL versions, are not supported.

In addition, insecure ciphers such as Triple-DES (3DES) and RC4 are not accepted. Merchants should configure their webservers to follow a similar TLS profile, only permitting secure cipher suites and TLS v1.2 and above.

More information and additional detail on a secure TLS configuration can be found at the following publications:

- Australian Government's Australian Signals Directorate (ASD) TLS and HTTPS configuration guidelines
- New Zealand Government Communications Security Bureau (NZ GCSB) Information Security Manual
- NIST Guidelines for the Selection, Configuration, and Use of Transport Layer Security

1.2. Case Sensitivity

All field “name” and “value” attributes should be treated as case sensitive.

1.3. HTML Forms

When using a HTML form, the following “form” tags are used to encapsulate Direct Post inputs:

```
<form method="post" action="https://…">
…
</form>
```

All INPUT fields must occur between the “form” tags for correct submission of information to the Direct Post Live and Test servers.

Direct Post only accepts POST data from an HTML form submitted by your customer on your website to initiate a transaction. Ensure that the “method” attribute is set to “post".

You may also add the “name” attribute or any other form functionality that you require.

1.4. Acceptable Form Input Tags

Any HTML form tags may be used to submit information to Direct Post.

This document deals predominantly with the “input” tag, however, you may use any form tag to create the necessary name/value data pairs that form the information interpreted by Direct Post

Most data are normally passed as “hidden” type input fields. Some fields, such as the card number, are entered by your customer and are typically passed as “text” type input fields.

Form inputs follow the structure:

```
<input type=”field_type” name=”field_name” value=”field_value”>
```

2. Transaction URL’s

Listed below are the live and test URLs for performing several functions

2.1. Test URL

Transactions are created by a HTML form submitted by your customer on your website to:

```
<form method="post" action="https://test.api.securepay.com.au/directpost/authorise">
```

2.2. Live URL

Transactions are created by a HTML form submitted by your customer on your website:

```
<form method="post" action="https://api.securepay.com.au/directpost/authorise">
```

3. Mandatory Fields

3.1. Merchant ID

The Merchant ID field, “EPS_MERCHANT”, is mandatory. It is the SecurePay account to process payments.

SecurePay Customer Support will supply your Merchant ID when your account is activated. The Merchant ID will be of the format “ABC0010”, where ABC is your unique three-letter account code, also used for logging in to the SecurePay Merchant Log In.

Example: Form tags with EPS_MERCHANT input field

```
<form method="post" action="https://…">
<input type="hidden" name="EPS_MERCHANT" value="ABC0010">
…
</form>
```

3.2. Transaction Type

The “EPS_TXNTYPE” defines the payment process. It allows switch of the payment type, as well as addition of optional services such as Store Only, FraudGuard, and 3D Secure 2. It also forms part of the Fingerprint.

3.2.1. Payment

Payments are real-time, immediately authorised card transactions. Transaction information is passed from a payment form to your SecurePay account for immediate processing.

Example: Form fields required to make a card payment

```
Hidden fields:
<input type="hidden" name="EPS_MERCHANT" value="ABC0010">
<input type="hidden" name="EPS_TXNTYPE" value="0">
<input type="hidden" name="EPS_REFERENCEID" value="Test Reference">
<input type="hidden" name="EPS_AMOUNT" value="1.00">
<input type="hidden" name="EPS_TIMESTAMP" value="201106141010">
<input type="hidden" name="EPS_FINGERPRINT"
value="01a1edbb159aa01b99740508d79620251c2f871d">
<input type="hidden" name="EPS_RESULTURL"
value="https://www.myserver.com.au/result.php">

Customer-entered fields:
<input type="text" name="EPS_CARDNUMBER">
<input type="text" name="EPS_EXPIRYMONTH">
<input type="text" name="EPS_EXPIRYYEAR">
<input type="text" name="EPS_CCV">
```

A typical approved result from this transaction would be:

```
https://www.myserver.com.au/result.php?
refid=Test Reference
&rescode=08
&restext=Honour with ID
&txnid=100036
&authid=151678
&settdate=20110617
&sig=MC0CFQCQnNRvziCb1o3q2XPWPljH8qbqpQIUQm9TpDX1NHutXYuxkbUk9
AfV+/M=
```

Refer to "B.1. Standard Result Fields" for more information on transaction results.

3.2.2. Pre-Authorisation

A pre-authorisation is a transaction that reserves funds on a card account. The Merchant can then complete the transaction later and receive the funds. If the pre-authorisation is never completed, it expires, usually after five days. After this, the reserved funds are again available to the card holder.

Pre-authorisations are often used by hotels to reserve funds at booking time and then completed when the guest checks out.

To pre-authorise an amount, submit all the fields exactly as they were for the PAYMENT (0) transaction type above, including the card details, and set:

```
<input type="hidden" name="EPS_TXNTYPE" value="1">
```

Once submitted, the result will be returned to your "EPS_RESULTURL" including the following field:

```
Example: Extra result field from a PREAUTH transaction

preauthid=516376
```

3.2.3. Store Only

When you choose to store a customer's card details in SecurePay's card storage system when a Direct Post transaction is processed, you can optionally choose to store their card details without charging their card. This is known as the Store Only method. Ensure that you have customer's agreement before storing their card.

When you use Store Only, the amount included is ignored and is not stored against the customer's details.

Please note EPS_STANDINGINSTRUCTIONTYPE functionality is not available yet. We will advise you of the date and provide detailed instructions ahead of the changes.*- For Visa and Mastercard cards and selected acquiring banks (NAB and fiserv FDMSA):*

- *An Account Verification will be initiated to verify the card with the bank before it gets stored. This verification step is a mandatory requirement from schemes which will help reduce customer complaints and improve approval rates.*
- *The standing instruction may be optionally provided using EPS_STANDINGINSTRUCTIONTYPE field. If this is passed, the card scheme will send a unique identifier called Standing Instruction ID that links to the payment history between the customer and merchant. SecurePay will store the SIID against the payor or token and will pass the value in subsequent transaction.*

To use Store Only, you must:

- Pass through the EPS_TXNTYPE value of 8 in your requests. This value is defined further in Section A.1.2.

  TYPICAL USE <input type="HIDDEN" name="EPS_TXNTYPE" value="8">

- Generate a fingerprint and pass this through as the EPS_FINGERPRINT value in your requests. This is a protected record of the transaction details and prevents a customer modifying the details when submitting their card information. Your system will need to create a SHA256 hash of the following fields in order, separated by "|". These fields are different to the standard fingerprint fields described in Section A.1.6.

  EPS_MERCHANT|TransactionPassword|EPS_TXNTYPE|EPS_STORETYPE|EPS_REFERENCEID|EPS_TIMESTAMP

  Example:
  ABC0001|abc123|8|payor|Test Reference|20151124041543
  TYPICAL USE <input type="HIDDEN" name="EPS_FINGERPRINT" value="
  95390d3899ad9dfcde342714e765c1de854704cdd841875ca45ee8fcb7de3385">

  When you use Store Only, the following financial transaction result fields are not returned:
  - rescode, restext, txnid, settdate, preauthid

3.3. Payment Reference

The "EPS_REFERENCEID" mandatory field is used to tag orders with an identifier meaningful to you. This may be your invoice number or a unique tracking number produced as part of your own website.

The Reference ID is available to the Result URL and emails, and appears as the Transaction Reference in the SecurePay Merchant login.

It is recommended that the Reference ID is unique to aid in reconciliation.

Example: Defining a reference id

Scenario: Your Company wants to include its invoice numbers with every payment.

```
<input type="hidden" name="EPS_REFERENCEID" value="Invoice#642193">
```

3.4. Transaction Amount

The "EPS_AMOUNT" mandatory field is the amount that will be transacted through your SecurePay account. By default, the currency is AUD (Australian Dollars).

It is passed in a dollars and cents format. For example, $1.00 would be passed as "1.00".

Example: Setting the transaction amount

Scenario: A customer chooses items from your shopping cart totalling AUD$53.20.

```
<input type="hidden" name="EPS_AMOUNT" value="53.20">
```

3.5. GMT Timestamp

When sending a request to Direct Post to generate a fingerprint or to process a transaction, you must pass a Greenwich Mean Time (GMT) timestamp in the field "EPS_TIMESTAMP" (also known as UTC).

The timestamp used to generate the fingerprint must exactly match the one sent with the associated transaction.

It must be of the format "YYYYMMDDHHMMSS" where:

- YYYY is the current year
- MM is the current two digit month 01 – 12
- DD is the current two digit day 01 – 31
- HH is the current two digit hour in 24-hour format 01 – 24
- MM is the current two digit minute 00 – 59
- SS is the current two digit second 00 – 59

Example: Setting the GMT timestamp

Scenario: Your system has generated a Fingerprint. It is currently 22:35:05 on 20/06/2011 in Sydney (+10 hours from GMT). The time in GMT is 12:35:05 on the same day.

<input type="hidden" name="EPS_TIMESTAMP" value="20110620123505">

3.6. Fingerprint

The Fingerprint is a protected record of the amount to be paid. It must be generated and then included on your customer payment HTML page as a hidden field. It prevents a customer modifying the transaction details when submitting their card information.

The Fingerprint is a SHA256 hash of the above fields plus the SecurePay Transaction Password in this order with a pipe separator "|":

- EPS_MERCHANTID
- Transaction Password (supplied by SecurePay Support)
- EPS_TXNTYPE
- EPS_REFERENCEID
- EPS_AMOUNT
- EPS_TIMESTAMP
- EPS_ORDERID (if your payment is authenticated with 3DS2)
- EPS_LSI (Optional field – if your payment is authenticated with 3DS2)

Example: Setting the fingerprint

Fields joined with a | separator:
ABC0010|abc123|0|Test Reference|1.00|20170307024842

SHA256 the above string:
dae18a7c6b223472c7a7c8116acbdbf1170a033c89f3b9b8f940069c453436de
<input type="hidden" name="EPS_FINGERPRINT" value="
dae18a7c6b223472c7a7c8116acbdbf1170a033c89f3b9b8f940069c453436de">

Example: Setting the fingerprint authenticated with 3DS2

Fields joined with a | separator:
ABC0010|abc123|4|Test Reference|1.00|20211004043120|f8633973-2b76-4517-93cf-828d789e230a

SHA256 the above string:
46bd20f1b5f662769ca600f78c6f5cdb5f159eafa50772354678679ac7c7d934
<input type="hidden" name="EPS_FINGERPRINT" value="
46bd20f1b5f662769ca600f78c6f5cdb5f159eafa50772354678679ac7c7d934">

For methods of generating a SHA256 hash in your language please visit:
https://rosettacode.org/wiki/SHA-256

3.7. Transaction Result URL

Use the field "EPS_RESULTURL" to set the secure page on your website that must receive and interpret the transaction result and display the result to the Customer.

When a transaction is complete (approved or declined), Direct Post redirects the browser to your result page with the transaction result in a series of POST fields. If you redirect Direct Post to another URL, fields may be sent via the GET method. Please handle both GET and POST methods.

The result includes a Fingerprint that you can verify to check the integrity of the transaction result.

The values of EPS_RESULTURL must:

- Be written as fully-qualified URLs. i.e. "https://...".
- Be secure URLs (unless you are testing) from a trusted provider (not self-signed)

Example: Set the Result URL

Scenario: The special result URL "result.php" has been configured to update a database and to display the receipt page.

```
<input type="hidden" name="EPS_RESULTURL"
value="https://www.myserver.com.au/result.php">
```

3.8. Card Information

Each transaction must include the card information submitted by a customer. This is private information and should not be visible to you or your system.

The fields, "EPS_CARDNUMBER", "EPS_EXPIRYMONTH", "EPS_EXPIRYYEAR" and "EPS_CCV" are all required for the transaction.

Visa and MasterCard have the card number and expiry date on the front, and a security number referred to as a CCV2 printed on the signature strip on the back of the card appearing as a three-digit number.

Example: Allow a customer to enter their card information

Scenario: Your system displays a payment page to the customer, complete with amount to pay, requesting input of card information. The following input fields collect that information:

```
<input type="text" name="EPS_CARDNUMBER">
<select name="EPS_EXPIRYMONTH">
        <option value="01">01
        <option value="02">02
        <option value="03">03
        <option value="04">04
        <option value="05">05
        <option value="06">06
        <option value="07">07
        <option value="08">08
```

```
        <option value="09">09
        <option value="10">10
        <option value="11">11
        <option value="12">12
</select>
<select name="EPS_EXPIRYYEAR">
        <option value="2010">2010
        <option value="2012">2011
        <option value="2013">2012
        <option value="2014">2013
        <option value="2015">2014
</select>
<input type="text" name="EPS_CCV">
```

4. Optional Fields

4.1. Currency

If your bank supports multicurrency, you may optionally set the currency of the transaction to one other than AUD

Set EPS_CURRENCY to any ISO three letter currency value.

```
Example: Set the currency to USD

EPS_CURRENCY=USD
```

4.2. Parameter Callback

All result fields are sent to your EPS_RESULTURL. In addition, a callback URL may also be specified to enable separation of the browser process from the update process.

Set EPS_CALLBACKURL similarly to the EPS_RESULTURL.

Fields are sent via the POST method. Following a redirect, fields may be sent via the GET method.

The result fields will then also include a callback_status_code – the HTTP response code from your URL.

Note that your callback URL must not contain multiple redirects or flash content or other content that may prevent Direct Post from successfully making a connection.

Your web host cannot use Server Name Indicators (SNIs) for determining which SSL certificate to serve. This is not supported by SecurePay's systems.

4.3. FraudGuard

If your account has been enabled for FraudGuard, you can set the optional field "EPS_TXNTYPE" to include the FraudGuard option and pass a series of additional payment parameters to the system to help validate your customer.

Note: FraudGuard cannot eliminate fraud. It observes transaction patterns and conservatively judges whether a transaction is likely to be fraudulent. You should always use your own judgement before sending goods or supplying services based on the result of any transaction.

All FraudGuard parameters are described in "FraudGuard Fields".

Example: Sending Fraud Guard parameters with a transaction.

Required (in addition to other required payment fields):
<input type="hidden" name="EPS_TXNTYPE" value="2">
<input type="hidden" name="EPS_IP" value="203.123.456.789">

Optional (any combination is acceptable):
<input type="hidden" name="EPS_FIRSTNAME" value="John">
<input type="hidden" name="EPS_LASTNAME" value="Smith">
<input type="hidden" name="EPS_ZIPCODE" value="2345">
<input type="hidden" name="EPS_TOWN" value="">
<input type="hidden" name="EPS_BILLINGCOUNTRY" value="US">
<input type="hidden" name="EPS_DELIVERYCOUNTRY" value="US">
<input type="hidden" name="EPS_EMAILADDRESS" value="john@email.com">

The field "EPS_IP" is your customer's browser IP address. This can be obtained from your web server as the “Remote IP” address environment variable.

If the transaction passes Fraud Guard, you will receive the following result codes:

rescode = Bank response code
restext = Bank response text
…
afrescode = 000

If the transaction does not pass FraudGuard you will receive:

rescode = Error code
restext = Associated error text
afrescode = Value other than 000
afrestext = Associated Fraud Guard result text

4.4. Card Storage

The card number used in the transaction may be optionally stored for subsequent batch or XML transaction triggering.

By setting "EPS_STORE=true", the card will be stored in SecurePay's Payor system using the EPS_REFERENCEID as the Payor ID.

Please note this functionality is not available yet. We will advise you of the date and provide detailed instructions ahead of the changes. - For Visa and Mastercard cards and selected acquiring banks (NAB and fiserv FDMSA), the standing instruction may be optionally provided using EPS_STANDINGINSTRUCTIONTYPE field. If this is passed, the card scheme will send back in the response a unique identifier called Standing Instruction ID (SIID) on the first transaction that links to the payment history between the customer and merchant. SecurePay will store the SIID against the payor or token and will pass the value in subsequent transaction.

4.4.1. Payor

This is the default card storage method.

With Payor storage, you define the Payor ID to store with the card. Customer Code can also be defined which can be a unique (within you organisation) identifier of your customer.

Cards, Payor ID's and Customer Codes can be edited or deleted via the Merchant Login. For any payor created with a Customer Code, it should always be passed along with the Payor ID in all types of requests.

You may also set "EPS_STORETYPE=payor" to use this storage type.

You may optionally pass in an alternative value for the stored Payor ID to override the use of EPS_REFERENCEID.

Set EPS_PAYOR to your required value.

Example: Set card storage with type Payor and my own Payor ID

```
EPS_REFERENCEID=123456
EPS_STORE=true
EPS_STORERTYPE=payor
EPS_PAYOR=MyCustomer
EPS_CUSTOMERCODE=MyCustomerCode
EPS_STANDINGINSTRUCTIONTYPE=1
```

4.4.2. Token

A Token is a string that represents a stored card number. If the card number changes, so does the token, therefore card numbers and tokens cannot be edited, they may only be added or deleted. However, a Customer Code can also be assigned for each token and can be edited through Merchant Portal.

Tokens can be used in 3rd party systems to represent card numbers.

If a card is passed to the system for storage several times, the same token is always returned.

To have SecurePay generate a token for a card or return an existing token for a pre-stored card set “EPS_STORETYPE=token”.

Direct Post will return the token in the result parameters.

```
Example: Set card storage with type Token

EPS_REFERENCEID=123456
EPS_STORE=TRUE
EPS_STORERTYPE=TOKEN
EPS_CUSTOMERCODE=MyCustomerCode
EPS_STANDINGINSTRUCTIONTYPE=1
```

4.4.3. Avtoken

If you are acquiring with NAB or fiserv FDMSA and you want to use SI Type or Customer Code, refer to 4.4.2. Token.

When using “EPS_STORETYPE=avtoken” then account verification will be performed and if passed will then either create and store a new token that represents the card number or return a pre-existing token if the card has been stored previously. Tokens are stored as non-editable Payors. If account verification fails, a decline reason will be returned, and the card will not be tokenised. There is a cost associated with account verification.

```
Example: Set card storage with type Avtoken

EPS_REFERENCEID=123456
EPS_STORE=TRUE
EPS_STORERTYPE=AVTOKEN
```

4.5. Stored Transaction Reference

When triggering a payment from a stored card of either type Payor or Token via batch or API, the Transaction Reference defaults to the Payor ID (or Token). This can be overridden by setting a specific Transaction Reference at the time of storage.

Set the EPS_PAYORREF to store your desired Transaction Reference against the stored card.

This is particularly useful; for tokens, as the token does not necessarily represent your customer. Customer Code can also be used in conjunction with the Transaction Reference.

```
Example: Set card storage with type Token and your own Transaction Reference

EPS_REFERENCEID=123456
EPS_STORE=true
```

```
EPS_STORERTYPE=TOKEN
EPS_PAYORREF=123456
EPS_CUSTOMERCODE=MyCustomerCode
EPS_STANDINGINSTRUCTIONTYPE=1
```

In this example, the Payment Reference ID and the stored Transaction Reference for future triggering are the same.

4.6. 3D Secure 2

3D Secure 2 (aka 3DS2) is an authentication messaging protocol managed by EMVCo used for card-not-present e-commerce transactions to authenticate cardholders with their corresponding card issuers before proceeding with payments.

To utilise this service, your SecurePay account must be enrolled with EMV 3D Secure, and your merchant facility must be integrated to use the 3DS2 feature.

3DS2 authentication must happen before an actual payment or pre-authorisation is sent through Direct Post. Please refer to 3DS2 Integration guide for more details on how to use SecurePay 3DS2 feature.

Once you have successfully authenticated your transaction with 3DS2, you can send 3DS2 authentication details through Direct Post by changing the value of the mandatory EPS_TXNTYPE field to use 3D secure. See EPS_TXNTYPE for more information. In addition, you need to include EPS_ORDERID and may include EPS_LSI, both with values coming from the result of the 3DS2 authentication in the fingerprint. Please see EPS_ORDERID for information on how to use the field.

4.6.1. EPS_LSI

To confirm that you accept the value returned for the Liability Shift Indicator in the **Authentication Outcome**, it is recommended to send an optional EPS_LSI value in the authorisation or preauthorisation request.

If you do not include an EPS_LSI value in the authorisation or preauthorisation request and the liability has not shifted (i.e., you are liable), the transaction will be declined.

To see more details on the field please refer to the 3D Secure 2 Integration guide in the Developer resources on the SecurePay website.

4.7. Payment Choice

Payment Choices are additional payment services that can be accessed via your Direct Post integration.

Direct Post offers the following payment choices:

- PayPal Express Checkout

4.7.1. PayPal

Do note the following integration requirements when selecting PayPal as a Payment Choice via Direct Post:

Set the EPS_PAYMENTCHOICE = “PayPal”.
EPS_TXNTYPE = 0 (Payment) is the only accepted payment type for PayPal.
EPS_CARDNUMBER, EPS_EXPIRYMONTH, EPS_EXPIRYYEAR and EPS_CVV must be left blank or NULL. The customer will be directed to the PayPal login page as part of the payment process

SecurePay FraudGuard and 3D Secure 2 (Verified by Visa and MasterCard SecureCode) cannot be used in conjunction with PayPal payments.

5. Transaction Result

After the transaction has been processed, a set of result parameters will be returned to the URL you defined in EPS_RESULTURL. You may then use these parameters within your defined result URL program to update your application and display the desired outcome to the Merchant.

5.1. Reading the result

Result parameters are returned using the POST method with parameter names as described in Appendix 2: Result Fields

Some parameters will only be returned if a particular feature is used.

5.2. Result Page Redirects

If your website redirects the Direct Post result to another page on your site, Direct Post will automatically follow the redirect. This will occur until Direct Post is no longer redirected.

Direct Post will POST result parameters the first time it calls your server, but Direct Post will send result parameters using the GET method based on RFC 2616 standards after being redirected.

6. Testing

As you build your system, you can test functionality when necessary, by submitting parameters to the test URL found in "2 Transaction URL’s". You can generate a fingerprint and then complete the transaction by using the card details listed below.

6.1. Test Card Number, Type and Expiry

Use the following information when testing transactions:

Card Number : 4444333322221111
Card Type : VISA
Card CCV : 123
Card Expiry : 08 / 24 (or any date greater than today)

6.2. Simulating Approved and Declined Transactions

You can simulate approved and declined transactions by submitting alternative payment amounts.

If the payment amount ends in 00, 08, 11 or 16, the transaction will be approved once card details are submitted. All other options will cause a declined transaction.

Payment amounts to simulate approved transactions:

$1.00
$1.08
$105.00
$105.08
(or any total ending in 00, 08)

Payment amounts to simulate declined transactions:

$1.51
$1.05
$105.51
$105.05
(or any totals not ending in 00, 08)

Note that when using the live URL for payments, the bank determines the transaction result, independent of the payment amount.

Glossary

Account Verification	Payment Account Validation enables a real-time account check with the card issuer if a particular account is valid and in good standing.
CSC	Cardholder Security Code. This is an extra code printed on the back of a Visa or MasterCard, typically shown as the last three digits on the signature strip. It is used during a payment as part of the cardholder authentication process. You may also know it as the Cardholder Verification Value (CVV), Card Verification Code (CVC), or the Personal Security Code. American Express and Diner Club Cards use a 4-digit Security Code in much the same manner.
3DS2	An authentication messaging protocol managed by EMVCo used for card-not-present e-commerce transactions to verify that the customer making the purchase is the legitimate cardholder before the actual payment is done. The issuer decides the authentication type to proceed based on the information provided. The issuer may prompt the customers to answer challenge questions if the transaction is deemed as high-risk.
FORM	The HTML tag used to mark the start and end of the area of your payment page that passes name/value data pairs to Direct Post.
HTML	Hypertext Markup Language. The language interpreted by web browsers. This is the language used to create your Direct Post payment form.
Hyperlink	A shortcut to another function within the system, accessed by clicking on an underlined label.
Input Field	HTML tags that define Form input fields. Used to submit information to Direct Post from your order form.
LSI	Liability Shift Indicator determines whether you or the card issuer is liable for fraudulent chargebacks after a customer completes 3DS2 authentication.
Log Date/Time	The date and time that the transaction was processed via Direct Post. Log Date and Time helps to tie a transaction back to your business system and assists in searching (via SecurePay's Transaction Search) for transactions which occurred during a specific period. Refer also to Settlement Date.
Merchant ID	Your SecurePay Merchant ID used to direct which account payments are made.
Merchant Number	Your bank's merchant number.
MOTO	An acronym for Mail Order/Telephone Order. MOTO is now a general term used to describe any process of processing a credit or charge card transaction by manual entry of the card details.

Payment	A transaction which both reserves card holder funds and transfers those funds to the merchants account in a single step. Refer also to Pre-authorisation and Complete.
Pre-authorisation	A transaction which reserves card holder funds but does transfer not those funds to the merchants account until a follow up Complete transaction is performed. Refer also to Complete and Payment.
Response Code	A numeric code associated with a transaction to indicate a specific transactions processing result. Transactions which are successfully passed through the banking system are returned with a two-digit response code allocated by the banking system. Transactions which were rejected during FraudGuard processing or which encountered technical problems and therefore were not successfully returned by the banking system will be allocated a 3 digit response code by SecurePay. A full list of response codes is available for download from the SecurePay website.
Settlement Date	The date on which funds associated with successful transactions are transferred to the merchant's account. Settlement is usually same day for transactions which have been processed by your bank before 6-10:00 pm AEST and next day for transactions processed after that time. Settlement for American Express, Diners and JCB cards will vary depending on your relationship with these organisations. Searching by Settlement Date helps to tie a transaction back to your bank statement. Refer also to Log Date/Time.
SSL	Secure Sockets Layer. The mechanism used to encrypt form data submitted from a browser.
Standing Instructions	Standing instructions describe a pre-existing arrangement with the cardholder to debit payments from their stored card in exchange for services.
Standing Instruction Type	Standing instructions can be of three types – Recurring, Instalment and Unscheduled (UCOF). **Please note this functionality is not available yet. We will advise you of the date and provide detailed instructions ahead of the changes.**
Standing Instruction ID	A unique identifier that will be sent by the card scheme on the first transaction that links to the payment history between the customer and merchant. **Please note this functionality is not available yet. We will advise you of the date and provide detailed instructions ahead of the changes.**
Transaction Password	This password is sent in transaction requests along with your Merchant ID to authenticate your account. It is not your online login password, however, it can be changed via your online login. Be aware that changing this password may prevent transactions from being processed unless you also update it in your programs.
Transaction Reference	A meaningful business reference such as customer name, customer number, order number, reservation number etc which you allocate to your transaction at the time of processing. Transactions processed by SecurePay are immediately recorded in the secure database which is accessed by the Merchant Log In.

	Transaction Reference (or any part of it) is an important search criterion within the Merchant Log In.
Transaction Source	The point of origination of this transaction. Valid Transaction Sources are: Online, Direct Post, IVR, Batch, Periodic, and Administration. Each of these is individually explained in more detail in this Glossary.
Transaction Type	The type of processing requested by this transaction. Valid Transaction Types are Payment and Pre-authorisation. Each of these is individually explained in more detail in this Glossary.

Appendices

Appendix A: Accepted Input Fields

Mandatory	Conditional	Optional	FraudGuard	3D Secure 2
EPS_MERCHANT EPS_TXNTYPE EPS_AMOUNT EPS_REFERENCEID EPS_TIMESTAMP EPS_FINGERPRINT EPS_RESULTURL	EPS_CARDNUMBER EPS_EXPIRYMONTH EPS_EXPIRYYEAR EPS_CCV	EPS_CURRENCY EPS_REDIRECT EPS_CALLBACKURL EPS_PAYMENTCHOICE **Card Storage** EPS_STORE **Card Storage (Optional)** EPS_STORETYPE EPS_PAYOR EPS_PAYORREF EPS_STANDINGINSTRUCTIONTYPE EPS_CUSTOMERCODE	EPS_IP **FraudGuard (Optional)** EPS_FIRSTNAME EPS_LASTNAME EPS_ZIPCODE EPS_TOWN EPS_BILLINGCOUNTRY EPS_DELIVERYCOUNTRY EPS_EMAILADDRESS	EPS_ORDERID EPS_LSI

A.1. Mandatory Fields

A.1.1. EPS_MERCHANT

Class	Mandatory
Format	Alpha-numeric, length 7
Description	A unique identifier for the Merchant within the Payment Gateway. This Merchant identifier value is an alphanumeric string allocated to you by SecurePay. This merchant identifier value is not the same as the merchant number provided by your bank.
Typical Use	<input type="hidden" name="EPS_MERCHANT" value="ABC0010">

A.1.2. EPS_TXNTYPE

Class	Mandatory		
Format	Numeric		
Description	Used to determine the processing type for an individual transaction. May be one of the following:		
	0	PAYMENT	A card payment/purchase transaction. Note: This is the only accepted type for PayPal payments.
	1	PREAUTH	Used to pre-authorise an amount on a card. The result parameters include the “preauthid” which must be stored and used when completing the pre-authorisation
	2	PAYMENT with FRAUDGUARD	A card payment/purchase transaction with the optional FraudGuard service
	3	PREAUTH with FRAUDGUARD	A card preauthorisation transaction with the optional FraudGuard service
	4	PAYMENT with 3D Secure 2	A card payment/purchase transaction with the optional 3DS2 service
	5	PREAUTH with 3D Secure 2	A card preauthorisation transaction with the optional 3DS2 service
	6	PAYMENT with FRAUDGUARD and 3D Secure 2	A card payment/purchase transaction with the optional FraudGuard and 3DS2 services
	7	PREAUTH with FRAUDGUARD and 3D Secure 2	A card preauthorisation transaction with the optional FraudGuard and 3DS2 services
	8	STORE ONLY	This will store the card details without taking a payment or preauthorisation. See section 3.2.3. for more details
Typical Use	<input type="hidden" name="EPS_TXNTYPE" value="0">		

A.1.3. EPS_AMOUNT

Class	Mandatory
Format	Numeric, two decimal places, from 0.01 to 99999999.99
Description	The total amount of the purchase transaction. This value must be a positive decimal value of dollars and cents. Please be careful to correctly specify the amount as Direct Post has no method of determining whether an amount has been correctly specified.

Typical Use	<input type="hidden" name="EPS_AMOUNT" value="107.95">

A.1.4. EPS_REFERENCEID

Class	Mandatory
Format	String, min length 1, max length 60
Description	A string that identifies the transaction. This string is stored by SecurePay as the Transaction Reference. This field is typically a shopping cart id or invoice number and is used to match the SecurePay transaction to your application.
Typical Use	<input type="hidden" name="EPS_REFERENCEID" value="My Reference">

A.1.5. EPS_TIMESTAMP

Class	Mandatory
Format	String, format "YYYYMMDDHHMMSS" in GMT (UTC)
Description	The GMT time used for Fingerprint generation. This value must be the same submitted to generate a fingerprint as submitted with the transaction. SecurePay validates the time to within one hour of current time. The time component must be in 24-hour time format.
Typical Use	<input type="hidden" name="EPS_TIMESTAMP" value="20110620122453">

A.1.6. EPS_FINGERPRINT

<table>
<tr><td>Class</td><td colspan="2">Mandatory</td></tr>
<tr><td>Format</td><td colspan="2">String, length up to 64</td></tr>
<tr><td rowspan="5">Description</td><td colspan="2">A SHA256 hash of</td></tr>
<tr><td>For EPS_TXNTYPE 0-7</td><td>EPS_MERCHANT|TransactionPassword|EPS_TXNTYPE|EPS_REFERENCEID|EPS_AMOUNT|EPS_TIMESTAMP</td></tr>
<tr><td>For EPS_TXNTYPE 4-7 (with 3DS2)</td><td>EPS_MERCHANT|TransactionPassword|EPS_TXNTYPE|EPS_REFERENCEID|EPS_AMOUNT|EPS_TIMESTAMP|EPS_ORDERID</td></tr>
<tr><td>For EPS_TXNTYPE 8</td><td>EPS_MERCHANT|TransactionPassword|EPS_TXNTYPE|EPS_STORETYPE|EPS_REFERENCEID|EPS_TIMESTAMP</td></tr>
<tr><td colspan="2">Where the EPS_ prefixed fields are sent in the request and “TransactionPassword” is obtained from SecurePay Support and maybe changed via the SecurePay Merchant Log In.</td></tr>
</table>

Typical Use	<input type="hidden" name="EPS_FINGERPRINT" value="dae18a7c6b223472c7a7c8116acbdbf1170a033c89f3b9b8f940069c453436de">

A.1.7. EPS_CARDNUMBER

Class	Conditional
Format	Numeric, min length 12, max length19
Description	The card number used in the transaction. The card number is **not required** for the following Payment Choices: • PayPal
Typical Use	<input type="text" name="EPS_CARDNUMBER" value="4444333322221111">

A.1.8. EPS_EXPIRYMONTH

Class	Conditional
Format	String, min length 1, max length 2
Description	The month in which the card expires. This may only contain an integer value between 1 and 12, inclusive, corresponding to the month of the year. The expiry month and expiry year together must form a date that is at least the current month. Transactions that contain an expiry date in the past will be rejected. A leading zero is allowed. The card number is **not required** for the following Payment Choices: • PayPal
Typical Use	<input type="text" name="EPS_EXPIRYMONTH" value="06">

A.1.9. EPS_ EXPIRYYEAR

Class	Conditional
Format	String, length 2 or 4
Description	The year in which the card expires. This should ideally be a 2 digit year value. The expiry month and expiry year together must form a date that is later than the current date. Transactions that contain an expiry date in the past will be rejected. Four-digit years are accepted, with the first two digits ignored. E.g. 1911 will be treated as 2011 The card number is **not required** for the following Payment Choices:

	• PayPal
Typical Use	<input type="text" name="EPS_EXPIRYYEAR" value="15">

A.1.10. EPS_CCV

Class	Conditional
Format	Numeric, length 3 or 4
Description	The Card Check Value (CCV) field should contain the three-digit value that is printed on the back of the card itself, or the four digit value printed on the front of American Express cards. When sending transactions to the Payment Gateway test facility, any 3- or 4-digit value will be accepted. This field may be referred to elsewhere as a Card Verification Value (CVV2) or a Card Verification Code (CVC), most notably in information provided by banks or card providers. The card number is not required for the following Payment Choices: • PayPal
Typical Use	<input type="text" name="EPS_CCV" value="999">

A.1.11. EPS_RESULTURL

Class	Mandatory
Format	String, fully qualified URL
Description	The URL on the Merchant website that accepts transaction result data as POST elements. The result page may be almost any form of web page, including static HTML pages, CGI scripts, ASP pages, JSP pages, PHP scripts, etc, however cookies or other forms of additional information will not be passed through the Payment Gateway. The EPS_RESULTURL must be a URL for a publicly visible page on a web server within a domain that is delegated to a public IP number. Internal machine names, such as "localhost", Windows-style machine names, and privately translated IP numbers will fail.
Typical Use	<input type="hidden" name="EPS_RESULTURL" value="http://www.myserver.com.au/result.asp">

A.1.12. EPS_CURRENCY

Class	Optional

Format	String, length, ISO 4217 three letter currency code
Default	AUD
Description	Used to set the transaction currency sent to the bank for processing. You must have a bank merchant facility that accepts currencies other than AUD before using this feature. Set the currency to any ISO 4217 three letter currency code. E.g. USD, NZD, GBP, etc.
Typical Use	<input type="hidden" name="EPS_CURRENCY" value="NZD">

A.1.13. EPS_REDIRECT

Class	Optional
Format	String, values “FALSE” or “TRUE”
Default	FALSE
Description	Directs the system to redirect to the EPS_RESULTURL. Result parameters are appended to the URL as a GET string. Validate the result fingerprint to ensure integrity of the bank response. Use the EPS_CALLBACK if separate database update and page redirect URL’s are required.
Typical Use	<input type="hidden" name="EPS_REDIRECT" value="TRUE">

A.1.14. EPS_CALLBACKURL

Class	Optional
Format	String, fully qualified URL
Description	The URL on the Merchant website that accepts transaction result data as POST elements for the purpose of updating a client database or system with the transaction response. The page is not displayed in the browser. The result page may be almost any form of web page, including static HTML pages, CGI scripts, ASP pages, JSP pages, PHP scripts. The EPS_CALLBACKURL must be a URL for a publicly visible page on a web server within a domain that is delegated to a public IP number. Internal machine names, such as "localhost", Windows-style machine names, and privately translated IP numbers will fail.
Typical Use	<input type="hidden" name="EPS_CALLBACKURL" value="http://www.myserver.com.au/read_callback_result.asp">

A.1.15. EPS_PAYMENTCHOICE

Class	Optional

Format	String, max length 30
Description	Used to select additional payment choices. Set the value for the selected Payment Choice: • For PayPal payments, set to “PayPal” For standard payment options, this field must be NULL or omitted.
Typical Use	<input type="hidden" name="EPS_PAYMENTCHOICE " value="PayPal” >

A.2. Card Storage Fields

A.2.1. EPS_STORE

Class	Mandatory for Card Storage
Format	String, values “FALSE” or “TRUE”
Default	FALSE
Description	TRUE to enable card storage
Typical Use	<input type="hidden" name="EPS_STORE" value="true">

A.2.2. EPS_ STORETYPE

Class	Optional if EPS_TXNTYPE=STORE ONLY
Format	String, values “PAYOR” or “TOKEN” or “AVTOKEN"
Default	PAYOR
Description	Type PAYOR will store the card in the SecurePay Payor database. The EPS_REFERENCE field will be used as the Payor ID unless overridden with EPS_PAYOR. Type TOKEN will either create and store a new token that represents the card number or return a pre-existing token if the card has been stored previously. Tokens are stored as non-editable Payors. Type AVTOKEN will first perform account verification and if passed will then either create and store a new token that represents the card number or return a pre-existing token if the card has been stored previously. Tokens are stored as non-editable Payors. If account verification fails, a decline reason will be returned, and the card will not be tokenised. There is a cost associated with account verification.
Typical Use	<input type="hidden" name="EPS_STORETYPE" value="payor">

A.2.3. EPS_PAYOR

Class	Optional if EPS_STORETYPE=PAYOR
Format	String, length up to 20
Default	If not specified, EPS_REFERENCEID is used
Description	The Payor ID to store with the Payor. This will become the Transaction Reference for future triggered payments against that Payor unless overridden with EPS_PAYORREF
Typical Use	<input type="hidden" name="EPS_PAYOR" value="MyPayorID">

A.2.4. EPS_PAYORREF

Class	Optional
Format	String, length up to 30
Description	Sets the Transaction Reference for future triggered payments. If not set, the system will log the Payor as the Transaction Reference when a payment is triggered.
Typical Use	<input type="hidden" name="EPS_PAYORREF" value="MyTransactionReference">

A.2.5. EPS_STANDINGINSTRUCTIONTYPE

Class	Optional
Format	Numeric, values “1”, “2” or “3” • “1” for Recurring • “2” for Instalment • “3” for Unscheduled Credential On File (UCOF)
Default	If no value is passed and • EPS_TXNTYPE = Store Only or • EPS_TXNTYPE = 0-Payment **and** EPS_STORE = true Then this field will default to “3” which is Unscheduled Credential on File (UCOF). Otherwise, it will pass the same value received in the request to the bank.
Description	**Please note this functionality is not available yet. We will advise you of the date and provide detailed instructions ahead of the changes.** This field is only applicable to Visa and Mastercard cards only and on selected acquiring banks (NAB and fiserv FDMSA). Standing instructions can be of three types: • Recurring – type of payment that is processed at fixed, regular intervals for an ongoing service e.g. gym membership.

	• Instalment – type of payment that is processed at fixed incremental payments made towards a single purchase at regular intervals and stop once the balance is paid off e.g. four $100 monthly payments towards a sofa that costs $400. • Unscheduled Credential On File (UCOF) – type of payment that is used for adhoc payment triggered using a stored card e.g. an auto top-up of an account whenever the amount falls below a certain threshold.
Typical Use	<input type="hidden" name="EPS_STANDINGINSTRUCTIONTYPE" value="1">

A.2.6. EPS_CUSTOMERCODE

Class	Optional
Format	String, length up to 30
Description	An optional field available for merchants to identify a customer and is associated with a Payor ID. For any payor created with a customer code, it should always be passed along with the payor ID in all types of requests (add, edit or delete). A Payor ID and Customer Code must be a unique combination.
Typical Use	<input type="hidden" name="EPS_CUSTOMERCODE" value=" MyCustomerCode">

A.3. FraudGuard Fields

FraudGuard is SecurePay's system for fraud minimisation. FraudGuard is an additional feature that must be enabled by SecurePay before utilisation.

Merchants using this feature are required to include the following fields with all transactions sent to the SecurePay system.

A.3.1. EPS_IP

Class	Mandatory when EPS_TXNTYPE includes FraudGuard
Format	String, length up to 15
Description	Payee's IPV4 IP Address – should be obtained from the card holder's browser. Typically a programmatic environment variable such as remote IP.
Typical Use	<input type="hidden" name="EPS_IP" value="203.123.456.789">

A.3.2. EPS_FIRSTNAME

Class	Optional
Format	String, length less than 30

Description	Payee's first name
Typical Use	<input type="text" name="EPS_FIRSTNAME">

A.3.3. EPS_LASTNAME

Class	Optional
Format	String, length less than 30
Description	Payee's last name
Typical Use	<input type="text" name="EPS_LASTNAME">

A.3.4. EPS_ZIPCODE

Class	Optional
Format	String, length less than 30
Description	Payee's zip/post code
Typical Use	<input type="text" name="EPS_ZIPCODE">

A.3.5. EPS_TOWN

Class	Optional
Format	String, length less than 30
Description	Payee's town
Typical Use	<input type="text" name="EPS_TOWN">

A.3.6. EPS_BILLINGCOUNTRY

Class	Optional
Format	String, length 2, ISO 4217 currency code
Description	Payee's Country two letter code
Typical Use	<input type="text" name="EPS_BILLINGCOUNTRY">

A.3.7. EPS_DELIVERYCOUNTRY

Class	Optional
Format	String, length 2, ISO 4217 currency code

Description	Order delivery country two letter code
Typical Use	<input type="text" name="EPS_DELIVERYCOUNTRY">

A.3.8. EPS_EMAILADDRESS

Class	Optional
Format	String, length less than 30
Description	Payee's email address
Typical Use	<input type="text" name="EPS_EMAILADDRESS">

A.4. 3D Secure 2 Fields

A.4.1. EPS_ORDERID

Class	Mandatory when EPS_TXNTYPE includes 3D Secure 2 and ONLY when 3DS2 is used
Format	String, length of 36
Description	Order ID is a uniquely generated ID by SecurePay to track 3DS2 authentication requests done for this payment request. Before submitting your payment request authenticated with 3DS2, you must generate a fingerprint with all the mandatory fields and add the 3DS2 order ID value through EPS_ORDERID field. Please note: If the payment is authenticated using 3DS2 and EPS_ORDERID field is provided but is left as blank in the fingerprint, then transaction would be rejected as invalid fingerprint. If the parameter is not provided, payment will be treated as normal payment.
Typical Use	Setting a fingerprint for payment with 3DS2: • EPS_MERCHANT|TransactionPassword|EPS_TXNTYPE|EPS_REFERENCEID|EPS_AMOUNT|EPS_TIMESTAMP|EPS_ORDERID Fields joined with a | separator: • ABC0010|abc123|4|Test Reference|1.00|20211004043120|f8633973-2b76-4517-93cf-828d789e230a • Example Fingerprint Value: 46bd20f1b5f662769ca600f78c6f5cdb5f159eafa50772354678679ac7c7d934

A.4.2. EPS_LSI

Class	Optional field when EPS_TXNTYPE includes 3D Secure 2 and ONLY when EPS_ORDERID is used.
Format	String, length of 1
Description	3D Secure 2 (3DS2) Liability shift indicator is a field returned to you by SecurePay via "liabilityShiftIndicator" during 3DS2 authentication. It may have

	a value either Y (Liability for relevant Chargeback reason codes is shifted to the issuer) or N (liability stays with merchant). If you wish to submit your payment with liabilityShiftIndicator = N, include EPS_LSI in the fingerprint with the value you received from liabilityShiftIndicator. The value sent in EPS_LSI must match the value SecurePay have stored for the 3DS2 authentication. Please note, EPS_LSI is sent along with EPS_ORDERID field.
Typical Use	Setting a fingerprint for payment with 3DS2 with liability shift indicator: EPS_MERCHANT|TransactionPassword|EPS_TXNTYPE|EPS_REFERENCEID|EPS_AMOUNT|EPS_TIMESTAMP|EPS_ORDERID|EPS_LSI Fields joined with a | separator: ABC0010|abc123|4|Test Reference|1.00|20211004043120|f8633973-2b76-4517-93cf-828d789e230a|N Example Fingerprint Value: 318c3bc22be788ab6492081b48c97564208564f34fff111214d4e18f558b8457

Appendix B: Result Fields

B.1. Standard Result Fields

Field	Description
summarycode	The one-digit summary of the transaction result 1 = Approved 2 = Declined by the bank 3 = Declined for any other reason Use “rescode” and “restext” for more detail of the transaction result.
rescode	The primary indicator of the transaction result. Bank response or internal error code numbers used to determine the transaction result. Rescode's of 00, 08 and 11 indicate approved transactions, while all other codes represent declines. A full list of response codes is available for download from the SecurePay website.
restext	The associated text for each "rescode". For bank response codes 00 – 99, this field is generated by the bank's payment systems. All other codes have the "restext" generated by SecurePay.
refid	The value of the EPS_REFERENCEID parameter from the transactions request. This value is returned to the Merchant's processing system to allow matching of the original transaction request.
txnid	The bank transaction ID. This string is unique at least per terminal, per bank and per settlement date. This value is required to be re-entered along with other details of the original payment when processing refunds.
settdate	The bank settlement date. This is the date the funds will be settled into the merchant's account. The date will correspond to today's date until the bank's cut-off time (typically 6-11pm), then roll to the following business day. The settlement date is returned in the format "YYYYMMDD".
preauthid	The bank pre-authorisation ID returned by the payment gateway. This value is used when sending a pre-authorisation complete transaction via XML or Batch.
pan	The masked card number of format first six…last three. E.g. 444433…111
expirydate	The four-digit expiry date entered by the customer. E.g. 0813
merchant	The EPS_MERCHANT value used for the transaction
timestamp	The GMT (UTC) time used for the response fingerprint of the format "YYYYMMDDHHMMSS". This value must be used when generating a string to compare to the response “fingerprint” value to validate the response. The time component must be in 24-hour time format.
fingerprint	A string used to validate the transaction output. A SHA256 hash of the following fields in order, separated by “|”:

	For EPS_TXNTYPE 0-7: merchant, transaction password, reference, amount, timestamp, summarycode For Example: ABC0010|mytxnpasswd|MyReference|1000|20170307030036|1 is SHA256 hashed to give: 0db5d20452e47d6adfc9ec326d0d47730128a6dec4612e50f7656337a83954f9 **For EPS_TXNTYPE 8:** merchant, transaction password, store type, reference, timestamp, summarycode For Example: ABC0010|mytxnpasswd|payor|Test Reference|20170307030036|1 is SHA256 hashed to give: 0e0949c77c10e980aa311a06f12593a5d17fc08b337c15965fedfc789265973e
callback_status_code	The HTTP status code of the callback to the EPS_CALLBACKURL. This can be used to determine if Direct Post was able to successfully contact your web server.

B.2. FraudGuard Result Fields

FraudGuard fields are returned in addition to the Standard Result Fields if the input field EPS_TXNTYPE includes FraudGuard.

Field	Description
afrescode	FraudGuard code if EPS_TXNTYPE includes FraudGuard. Returns “400” if the transaction passes FraudGuard tests. Returns a different string depending on the type of fraud detected.
afrestext	FraudGuard response text. Used if the “afrescode” is not 000. Contains a description of the FraudGuard result.

B.3. Card Storage Result Fields

Card storage fields are returned in addition to the Standard Result Fields if the input field EPS_STORE=TRUE.

Field	Description
strescode	Storage code Returns “800” if the Payor or Token was successfully stored. Returns a different string if the storage failed. The “strestext” describes the failure reason.
strestext	Storage response text. Contains a description of the storage result.

payor	If EPS_STORETYPE is set to “PAYOR” (default), the EPS_PAYOR field will be returned in this result field.
token	If EPS_STORETYPE is set to “TOKEN”, the system-generated token will be returned in this field. If the card has never been stored before, this will be a new value. If the card has been stored previously, the stored value will be returned. If EPS_STORETYPE is set to “AVTOKEN”, after account verification has been successful, a system-generated token will be returned in this field. If the card has never been stored before, this will be a new value. If the card has been stored previously, the stored value will be returned.
customercode	If EPS_CUSTOMERCODE is provided, the same value will be returned in this field.
standinginstructiontype	The same EPS_STANDINGINSTRUCTIONTYPE value will be returned in this field. Applicable to Visa and Mastercard cards only and on selected acquiring banks (NAB and fiserv FDMSA)
standinginstructionid	If EPS_STORE is set to “True”, the card scheme will send back in the response a unique identifier called Standing Instruction ID (SIID) on the first transaction that links to the payment history between the customer and merchant. SecurePay will store the SIID against the payor and will pass the value in subsequent transaction. Applicable to Visa and Mastercard cards only and on selected acquiring banks (NAB and fiserv FDMSA)