



Application Integrated Hosted Payment Page
Integration Guide

Table of Contents

1	Introduction.....	2
1.1	About this Guide.....	2
1.2	Intended Audience.....	2
2	System Overview.....	3
3	Integration.....	4
3.1	General Information.....	4
3.2	Transaction URLs.....	4
3.3	Mandatory Input Fields.....	4
3.4	Transaction Options.....	7
3.5	Receiving Result Parameters.....	10
3.6	Testing.....	10
4	Appendices.....	12
4.1	Appendix 1: Accepted Input Fields.....	12
4.2	Mandatory Fields.....	12
1.2	Appendix 2: Result Fields.....	20

1 Introduction

1.1 About this Guide

This guide provides technical information about integrating and configuring SecurePay's Hosted Payment Page within your shopping cart or application.

The Integrated Hosted Payment Page offers a secure and flexible hosted service that helps you meet your PCI DSS obligations.

1.2 Intended Audience

This document is intended for developers, integrating SecurePay's Hosted Payment Page with their applications or shopping cart.

It is recommended that someone with web site, HTML and application programming experience reads this guide and implements the Hosted Payment Page.

2 System Overview

SecurePay's Integrated Hosted Payment Page provides merchants with the ability to process card payments in a secure environment.

SecurePay partners with the following major banks and financial institutions in the provision of the SecurePay Payment Gateway:

- ANZ (Australia)
- ANZ (New Zealand)
- American Express
- BWA Merchant Services (formerly Bank West)
- Citibank
- Commonwealth Bank
- Diners Club
- Macquarie Bank
- National Australia Bank
- St George
- Westpac

The integrated Hosted Payment Page supports card transactions only.

The integrated Hosted Payment Page supports card transactions only.

It is a fully hosted service and comes with a variety of options to help you integrate such as:

- iFramed or full page templates
- Call-back of results to your application
- Redirect options for receipt page
- FraudGuard
- Surcharging
- Card storage and tokenisation
- 3D Secure

3 Integration

3.1 General Information

3.1.1 Case Sensitivity

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

3.1.2 Sending Data

The Hosted Payment Page system accepts POST or GET data from your application to initiate a transaction, however, POST is the preferred action type.

When using an HTML form, the following “form” tags are used to encapsulate Hosted Payment Page inputs:

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

All INPUT fields must occur between the “form” tags for correct submission of information to the Hosted Payment Page and Test servers.

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

3.1.3 Acceptable Input Fields

Any HTML form tags may be used to submit information to the Hosted Payment Page service.

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 the service.

Most data is normally passed as “hidden” type input fields. Some fields that may be entered by your Customer are typically passed as “text” type input fields.

Form inputs follow the structure:

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

3.2 Transaction URLs

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

3.2.1 Test URL

```
https://payment.securepay.com.au/test/v2/invoice
```

3.2.2 Live URL

```
https:// payment.securepay.com.au/live/v2/invoice
```

3.3 Mandatory Input Fields

3.3.1 Bill Name

The Bill Name defines the flow of the transaction process.

“bill_name” must be set to “transact”

Example: Set the bill name for the default transaction process:

```
<input type="hidden" name="bill_name" value="transact">
```

3.3.2 Merchant ID

The Merchant ID field, “merchant_id”, is mandatory. It is the SecurePay account used 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 ABC00 is your unique account code.

Example: Setting the SecurePay Merchant ID:

```
<input type="hidden" name="merchant_id" value="ABC0010">
```

3.3.3 Transaction Type

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

3.3.3.1 Payment

Payments are real-time, immediately authorised card transactions. Transaction information is submitted by your customer via the Hosted Payments Page to your SecurePay account for immediate processing.

Example: Form fields required to make a card payment

```
<input type="hidden" name="merchant_id" value="ABC0010">
<input type="hidden" name="primary_ref" value="Test Reference">
<input type="hidden" name="txn_type" value="0">
<input type="hidden" name="amount" value="100">
<input type="hidden" name="fp_timestamp" value="201106141010">
<input type="hidden" name="fingerprint"
value="01aledbb159aa01b99740508d79620251c2f871d">
```

3.3.3.2 Pre-Authorisation

A pre-authorisation is a transaction that reserves funds on a card account. The Merchant can then complete the transaction at a later date 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 transaction type above and set:

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

3.3.4 Transaction Reference

The “primary_ref” mandatory field is used to tag orders with an identifier meaningful to you. This may be your invoice number, or could be a unique tracking number produced as part of your own web site.

The Payment Reference is available to the Result URL and emails, and appears as the Transaction Reference in the SecurePay Merchant Log In.

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

Example: Defining a Transaction Reference

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

```
<input type="hidden" name="primary_ref" value="642193">
```

3.3.5 Transaction Amount

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

It must be passed in the base unit of the currency. For Australian Dollars this is cents. For example, \$1.00 AUD would be passed as "100". 100 Yen would be passed as "100".

Example: Setting the transaction amount

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

```
<input type="hidden" name="amount" value="5320">
```

3.3.6 GMT Timestamp

You must pass a valid Greenwich Mean Time (GMT) timestamp in the field "fp_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="fp_timestamp" value="20110620123505">
```

3.3.7 Fingerprint

The Fingerprint is a protected record of the amount to be paid. It must be generated and then included as an input field to the Hosted Payment Page service. It prevents a customer modifying the transaction details when submitting their card information.

The Fingerprint is a SHA1 hash of the above mandatory fields, plus the SecurePay Transaction Password in this order with a pipe separator "|":

- "merchant_id"
- Transaction Password (supplied by SecurePay Support)
- "txn_type"
- "primary_ref"
- "amount"
- "fp_timestamp"

Example: Setting the fingerprint

Fields joined with a | separator:

```
ABC0010|txnpassword|0|Test Reference|100|20110616221931
```

SHA1 the above string:

```
0433efad55f93acbf343907be7dffa989a7cd69f
```

```
<input type="hidden" name="fingerprint"
value="0433efad55f93acbf343907be7dffa989a7cd69f">
```

For methods of generating a SHA1 hash in your language please visit:

http://code.wikia.com/wiki/SHA_checksum

3.4 Transaction Options

3.4.1 Receipt Page Redirect

Display of the receipt page and button options on the hosted receipt page can be optionally controlled.

By default, the Hosted Page system will display the receipt page to the customer.

To redirect to your own receipt page, set "display_receipt=no".

If the hosted page system displays the receipt (default), you can control the button on the receipt page via the following optional parameters:

return_url – the fully qualified URL to embed in the button link. The system will append result parameters to this link by default.

return_url_text – the text of the button

return_url_target – the target of the button. This can be one of "self", "new", "parent" or "top". "parent" is useful when using the iFrame template to take the browser back to full screen.

Example: Set the button to say "Continue..." and take the browser out of the iFrame when clicked.

```
<input type="hidden" name="return_url" value="http://www.mysite.com.au">
<input type="hidden" name="return_url_text" value="Continue...">
<input type="hidden" name="return_url_target" value="parent">
```

By default, the return_url is used when a Cancel button is clicked by the card holder. This url can be explicitly set by using the "cancel_url" parameter.

3.4.2 Card Storage

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

By setting the field "store=yes", the card will be stored in SecurePay's Payor system using the "primary_ref" as the Payor ID by default.

3.4.2.1 Payor

This is the default card storage method.

With Payor storage, you define the Payor ID to store with the card. Cards and Payor ID's can be edited via the Merchant Login.

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

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

Set "payor" to your required value.

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

```
<input type="hidden" name="primary_ref" value="123456">
<input type="hidden" name="store" value="yes">
<input type="hidden" name="store_type" value="payor">
<input type="hidden" name="payor" value="MyCustomer">
```

3.4.2.2Token

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.

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 "store_type=token".

The Hosted Payments Page will return the token in the result parameters.

Example: Set card storage with type Token

```
<input type="hidden" name="primary_ref" value="123456">
<input type="hidden" name="store" value="yes">
<input type="hidden" name="store_type" value="TOKEN">
```

3.4.2.3Stored 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 "payor_ref" field 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.

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

```
<input type="hidden" name="primary_ref" value="123456">
<input type="hidden" name="store" value="yes">
<input type="hidden" name="store_type" value="TOKEN">
<input type="hidden" name="payor_ref" value="123456">
```

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

3.4.3 Currency

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

Set the field "currency" to any ISO three letter currency value.

Example: Set the currency to USD

```
<input type="hidden" name="currency" value="USD">
```

3.4.4 Surcharging

Surcharging may be applied across all card types (Visa, MasterCard, etc.), or set individually per card type

Surcharging can be either or both:

- A percentage rate of the invoice total (E.g. 1%)
- A flat fee (E.g. \$2.00) The fee is added on top of any rate.

To activate surcharging set "surcharge=yes". This will not apply any surcharging unless rates and/or fees have been set. It will turn on optional result parameters.

To set a surcharge rate of 1%, set "surcharge_rate=1".

Example: Set a surcharge rate of 1% for all cards

```
<input type="hidden" name="surcharge" value="yes">
```

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

Individual rates and fees may be also set per card.

Example: Set a surcharge rate of 1% for Amex only

```
<input type="hidden" name="surcharge" value="yes">
```

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

3.4.5 Template

There are several options to control how the hosted pages are displayed to your customers. This is controlled by the optional "template" parameter.

Set "template=default" to use the full screen payment option. If the template parameter is omitted, this is the default.

Set "template=iframe" to use the iFrame version. This version displays only the input fields, confirmation text and form buttons.

Tip: use the Return URL text and target options to control the payment flow from within the iFrame.

Example: Set the template to iFrame

```
<input type="hidden" name="template" value="iframe">
```

3.4.6 Look and Feel

Several options can be passed to the system to change the look and feel. These include the header logo, the page title and the primary reference name

Example: Set the logo, title and primary reference values

```
<input type="hidden" name="page_header_image" value="https://www.mysite.com.au/header.jpg">
```

```
<input type="hidden" name="title" value="Payment Page">
```

```
<input type="hidden" name="primary_ref_name" value="Order Number">
```

For more look and feel options, please see the Appendices.

3.4.7 FraudGuard

FraudGuard is an optional service offered by SecurePay.

It can be activated by setting the "txn_type" field. See the Appendices for more details.

Please contact SecurePay Sales for pricing and more information.

3.4.8 3D Secure

The 3D Secure service is a combined offering of Verified by Visa and MasterCard SecureCode.

It can be activated by setting the "txn_type" field. See the Appendices for more details.

Please contact SecurePay Sales for pricing and more information.

3.5 Receiving Result Parameters

The system will send back to your URL's a predefined set of result parameters.

Some parameters will only be returned if an option is activated, such as "store".

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

Result parameters are available to you via two methods:

- Return URL – When you specify a result URL, result parameters are appended to the URL. This occurs for both receipt page redirect and for the return button on the hosted receipt page.
- Call-back URL – The Hosted Page System contacts your system in the background and sends result parameters

In addition, you can configure a unique cancel URL which overrides the return_url when the Cancel button is clicked on the page.

All result parameters are listed in the Appendices.

3.6 Testing

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

3.6.1 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 / 13 (or any date greater then today)
```

3.6.2 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 (100)

\$1.08 (108)

\$105.00 (10500)

\$105.08 (10508)

(or any total ending in 00, 08)

Payment amounts to simulate declined transactions:

\$1.51 (151)

\$1.05 (105)

\$105.51 (10551)

\$105.05 (10505)

(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.

4 Appendices

4.1 Appendix 1: Accepted Input Fields

Mandatory	Transaction	Surcharging	Flow
bill_name	currency	surcharge	display_receipt
merchant_id		surcharge_rate	return_url
txn_type		surcharge_fee	return_url_text
amount	Card Storage	surcharge_rate_v	return_url_target
primary_ref	store	surcharge_fee_v	callback_url
fp_timestamp		surcharge_rate_m	cancel_url
fingerprint	Card Storage (Optional)	surcharge_fee_m	cancel_url_text
	store_type	surcharge_rate_a	cancel_url_target
	payor	surcharge_fee_a	confirmation
	payor_ref	surcharge_rate_d	Look and Feel
		surcharge_fee_d	template
		surcharge_rate_j	primary_ref_name
		surcharge_fee_j	page_header_image
			page_footer_image
			page_title
			card_types
			page_style_url

4.2 Mandatory Fields

4.2.1.1 *bill_name*

CLASS:	Mandatory
FORMAT:	Fixed value "transact"
DESCRIPTION:	Defines the transaction process.
TYPICAL USE:	<code><input type="hidden" name="bill_name" value="transact"></code>

4.2.1.2 *merchant_id*

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 <u>not</u> the same as the merchant number provided by your bank.
TYPICAL USE:	<input type="hidden" name="merchant_id" value="ABC0010">

4.2.1.3 *txn_type*

CLASS:	Mandatory
FORMAT:	Numeric
DESCRIPTION:	Used to determine the processing type for an individual transaction. May be one of the following: <ul style="list-style-type: none"> • "0" - PAYMENT: A card payment/purchase transaction. • "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: A card payment/purchase transaction with the optional 3D Secure service • "5" - PREAUTH with 3D Secure: A card preauthorisation transaction with the optional 3D Secure service • "6" - PAYMENT with FRAUDGUARD and 3D Secure: A card payment/purchase transaction with the optional FraudGuard and 3D Secure services • "7" - PREAUTH with FRAUDGUARD and 3D Secure: A card preauthorisation transaction with the optional FraudGuard and 3D Secure services
TYPICAL USE:	<input type="hidden" name="txn_type" value="0">

4.2.1.4 *amount*

CLASS:	Mandatory
FORMAT:	Numeric, integer, from 1 to 99999999
DESCRIPTION:	The total amount of the purchase transaction. This value must be a positive integer in the base unit of the currency, such as cents for AUD. Please be careful to correctly specify the amount as the system has no method of determining whether an amount has been correctly specified.
TYPICAL USE:	<input type="hidden" name="amount" value="10795">

4.2.1.5 *primary_ref*

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="primary_ref" value="My Reference">

4.2.1.6 *fp_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="fp_timestamp" value="20110620122453">

4.2.1.7 fingerprint

CLASS:	Mandatory
FORMAT:	String, length up to 60
DESCRIPTION:	A SHA1 hash of merchant_id TransactionPassword txn_type primary_ref amount fp_timestamp Where the "TransactionPassword" is obtained from SecurePay Support and maybe changed via the SecurePay Merchant Log In. All other fields must be exactly as sent.
TYPICAL USE:	<input type="hidden" name="fingerprint" value="01a1edbb159aa01b99740508d79620251c2f871d">

4.2.2 Transaction Fields

4.2.2.1 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. The format of the amount must match the currency. E.g. Yen has no decimal place.
TYPICAL USE:	<input type="hidden" name="currency" value="NZD">

4.2.3 Card Storage Fields

4.2.3.1 store

CLASS:	Mandatory for Card Storage
FORMAT:	String, values "yes" or "no"
DEFAULT:	no
DESCRIPTION:	"yes" to enable card storage
TYPICAL USE:	<input type="hidden" name="store" value="yes">

4.2.3.2 *store_type*

CLASS:	Optional
FORMAT:	String, values "PAYOR" or "TOKEN"
DEFAULT:	PAYOR
DESCRIPTION	Type PAYOR will store the card in the SecurePay Payor database. The "primary_ref" field will be used as the Payor ID unless overridden with "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.
TYPICAL USE:	<input type="hidden" name="store_type" value="payor">

4.2.3.3 *payor*

CLASS:	Optional if store_type=PAYOR
FORMAT:	String, length up to 20
DEFAULT:	If not specified, "primary_ref" 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 "payor_ref"
TYPICAL USE:	<input type="hidden" name="payor" value="MyPayorID">

4.2.3.4 *payor_ref*

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="payor_ref" value="MyTransactionReference">

4.2.4 *Surcharge Fields*

4.2.4.1 *surcharge*

CLASS:	Mandatory for Surcharging
FORMAT:	String, values "yes" or "no"
DEFAULT:	no
DESCRIPTION	"yes" to enable surcharge calculation and result parameters
TYPICAL USE	<input type="hidden" name="surcharge" value="yes">

4.2.4.2 *surcharge_rate*

CLASS:	Optional
--------	----------

FORMAT:	Numerical, 0.0001 to 99.9999
DEFAULT:	0
DESCRIPTION	Percentage surcharge to apply across all card types. May be overridden by applying individual card type rates.
TYPICAL USE	<input type="hidden" name="surcharge_rate" value="1.00">

4.2.4.3 *surcharge_fee*

CLASS:	Optional
FORMAT:	Numerical, integer, 0 to 999999 cents (or base currency unit)
DEFAULT:	0
DESCRIPTION	Fee amount to be added to the payment amount for all card types.
TYPICAL USE	<input type="hidden" name="surcharge_fee" value="100">

4.2.4.4 *surcharge_rate_v/m/a/d/j*

CLASS:	Optional
FORMAT:	Numerical, 0.01 to 99.9
DEFAULT:	0
DESCRIPTION	Percentage surcharge to apply to a specific card type. Overrides "surcharge_rate".
TYPICAL USE	<input type="hidden" name="surcharge_rate_v" value="1">

4.2.4.5 *surcharge_fee_v/m/a/d/j*

CLASS:	Optional
FORMAT:	Numerical, 0 to 999999 cents (or base currency unit)
DEFAULT:	0
DESCRIPTION	Fee amount to be added to the payment amount for a specific card type. Overrides "surcharge_fee".
TYPICAL USE	<input type="hidden" name="surcharge_fee_v" value="100">

4.2.5 *Transaction Flow Fields*

4.2.5.1 *display_receipt*

CLASS:	Optional
FORMAT:	String, values "yes" or "no"
DEFAULT:	yes
DESCRIPTION	Define which system displays the receipt page to the card holder. "Yes" means SecurePay displays the receipt page. To redirect to your own receipt page set this value to "no" and use the return_url parameter.

TYPICAL USE	<input type="hidden" name="display_receipt" value="no">
-------------	---

4.2.5.2 return_url

CLASS:	Optional
FORMAT:	String, fully-qualified URL
DESCRIPTION:	<p>The URL of the page on the Merchant web site that accepts transaction result data as POST elements.</p> <p>The 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.</p> <p>The "return_url" 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.</p> <p>If "display_receipt" is set to "yes", this is the URL of the button on the hosted result page.</p> <p>If "display_receipt" is set to "no", this is the redirect URL following a transaction.</p>
TYPICAL USE:	<input type="hidden" name="return_url" value="http://www.myserver.com.au/result.asp">

4.2.5.3 return_url_text

CLASS:	Optional
FORMAT:	String, length up to 30 characters
DEFAULT:	none
DESCRIPTION	Defines the text on the hosted result page button that takes the card holder to the next step in the process.
TYPICAL USE	<input type="hidden" name="return_url_text" value="Continue">

4.2.5.4 return_url_target

CLASS:	Optional
FORMAT:	String, values "self", "top", "parent" or "new"
DEFAULT:	none
DESCRIPTION	Defines the target of the hosted result page button that takes the card holder to the next step in the process. Useful for iFramed pages to take the process out of the iFrame.
TYPICAL USE	<input type="hidden" name="return_url_target" value="parent">

4.2.5.5 callback_url

CLASS:	Optional
FORMAT:	String, fully-qualified URL
DESCRIPTION:	<p>The URL of the page on the Merchant web site that accepts transaction result data as POST elements.</p> <p>The 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.</p>

	The "callback_url" 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="callback_url" value="http://www.myserver.com.au/result.asp">

4.2.5.6 *cancel_url*

CLASS:	Optional
FORMAT:	String, fully-qualified URL
DEFAULT:	Uses the return_url if not present.
DESCRIPTION:	<p>The URL of the page on the Merchant web site that accepts transaction result data as POST elements when the card holder clicks a Cancel button on a hosted page.</p> <p>The 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.</p> <p>The "cancel_url" 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.</p> <p>Note: some result parameters may not be populated depending on the point of cancellation of the process.</p>
TYPICAL USE:	<input type="hidden" name="cancel_url" value="http://www.myserver.com.au/result.asp">

4.2.5.7 *confirmation*

CLASS:	Optional
FORMAT:	String, values "yes" or "no"
DEFAULT:	yes
DESCRIPTION:	<p>By default, a Confirmation page is displayed to the cardholder, after the card details have been entered. This is to allow the cardholder to review the details (and edit as required) prior to initiating the payment.</p> <p>Setting the "confirmation" field to "no" will bypass this default confirmation step.</p>
TYPICAL USE:	<input type="hidden" name="confirmation" value="no">

4.2.6 *Look and Feel Fields*

4.2.6.1 *template*

CLASS:	Optional
FORMAT:	String, values "default" or "iframe"
DEFAULT:	default
DESCRIPTION:	Defines which hosted page template is displayed to card holders.
TYPICAL USE:	<input type="hidden" name="template" value="iframe">

1.1.1.1*primary_ref_name*

CLASS:	Optional
FORMAT:	String, length up to 30 characters
DEFAULT:	Invoice Number
DESCRIPTION	Defines the label on the hosted pages for the Primary Reference field.
TYPICAL USE	<code><input type="hidden" name="primary_ref_name" value="Order Number"></code>

1.1.1.2*page_header_image*

CLASS:	Optional
FORMAT:	String, fully-qualified URL. HTTPS method only. Must end in one of gif, jpeg, jpg, png. Valid images only.
DEFAULT:	none
DESCRIPTION	URL of an image to be used as the header of the hosted pages.
TYPICAL USE	<code><input type="hidden" name="page_header_image" value="http://www.myserver.com.au/result.asp/header.jpg"></code>

1.1.1.1*page_footer_image*

CLASS:	Optional
FORMAT:	String, fully-qualified URL. HTTPS method only. Must end in one of gif, jpeg, jpg, png. Valid images only.
DEFAULT:	none
DESCRIPTION	URL of an image to be used as the footer of the hosted pages.
TYPICAL USE	<code><input type="hidden" name="page_footer_image" value="http://www.myserver.com.au/result.asp/header.jpg"></code>

1.1.1.2*page_title*

CLASS:	Optional
FORMAT:	String, length up to 30 characters
DEFAULT:	Invoice Payment
DESCRIPTION	Defines the title text on the hosted pages.
TYPICAL USE	<code><input type="hidden" name="page_title" value="Pay Your Invoice"></code>

1.1.1.3*card_types*

CLASS:	Optional
FORMAT:	String, bar " " separate values from VISA, MASTERCARD, AMEX, DINERS, JCB
DEFAULT:	Visa, MasterCard

DESCRIPTION	Defines the card types accepted. This must be a subset of the card types you accept.
TYPICAL USE	<code><input type="hidden" name="card_types" value="VISA MASTERCARD"></code>

1.1.1.1*page_style_url*

CLASS:	Optional - NOT YET IMPLEMENTED
FORMAT:	String, fully-qualified URL
DESCRIPTION:	The URL of the page on the Merchant web site that provides styling for the hosted page. Styles override the default page styles. Images embedded by be fully qualified URL's. All JavaScript and malicious code will be removed.
TYPICAL USE:	<code><input type="hidden" name="callback_url" value="http://www.myserver.com.au/result.asp"></code>

1.2 Appendix 2: Result Fields

1.2.1 Standard Result Fields

1.2.1.1*summarycode*

The one digit summary of the transaction result:

- 1 = Approved
- 2 = Declined by the bank
- 3 = Declined for any other reason
- 4 = Cancelled by user

Use "rescode" and "restext" for more detail of the transaction result.

1.2.1.2*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 web site.

1.2.1.3*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.

1.2.1.4*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.

1.2.1.5*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.

1.2.1.6*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".

1.2.1.7*pan*

The masked card number of format first six...last three. E.g. 444433...111

1.2.1.8*expirydate*

The four digit expiry date entered by the customer. E.g. 0813

1.2.1.9*merchant*

The EPS_MERCHANT value used for the transaction

1.2.1.10 *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.

1.2.1.11 *amount*

The amount in the base unit of the currency, typically cents for Australian Dollars. E.g. \$104.23 will be 10423.

1.2.1.12 *fingerprint*

A string used to validate the transaction output.

A SHA1 hash of the following fields in order, separated by "|":

merchant, transaction password, reference, amount, timestamp, summarycode

For Example:

ABC0010|mytxnpasswd|MyReference|1000|201105231545|1

is SHA1 hashed to give:

3f97240c9607e86f87c405af340608828d331e10

1.2.1.13 *cardtype*

The card type used for the transaction. This will be one of:

- "Visa"
- "MasterCard"
- "American Express"
- "Diners"
- "JCB"

1.2.2 Preauthorisations

Preauth fields only returned when `txn_type` is set to `preauth` on input.

1.2.2.1 `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.

1.2.3 FraudGuard Result Fields

FraudGuard fields are returned in addition to the Standard Result Fields if your account is enabled for FraudGuard by SecurePay' Support team and the "`bill_name`" includes the FraudGuard option.

1.2.3.1 `afrescode`

FraudGuard code if `txn_type` includes FraudGuard. Returns "400" if the transaction passes FraudGuard tests. Returns a different string depending on the type of fraud detected.

1.2.3.2 `afrestext`

FraudGuard response text. Used if the "`afrescode`" is not 000. Contains a description of the FraudGuard result.

1.2.4 Card Storage Result Fields

Card storage fields are returned in addition to the Standard Result Fields if the input field "`store`" "`yes`" on input.

1.2.4.1 `stsummarycode`

The one digit summary of the storage result:

1 = Successful

2 = Unsuccessful

4 = Cancelled by user

Use "`strescode`" and "`strestext`" for more detail of the storage result.

1.2.4.2 `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.

1.2.4.3 `strestext`

Storage response text. Contains a description of the storage result.

1.2.4.4 `payor`

If "`store_type`" is set to "`payor`" (or absent - default), the "`payor`" field will be returned in this result field.

1.2.4.5 `token`

If "`store_type`" 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.

1.2.5 Surcharge Result Fields

If “surcharge” is set to “yes” on input or the fields are set in your account, the following additional fields are returned:

1.2.5.1baseamount

The amount passed to the system, prior to the addition of the surcharge, in the base unit of the currency, typically cents for Australian Dollars. E.g. \$104.23 will be 10423. This amount added to the “suramount” equals the “amount”.

1.2.5.2suramount

The surcharge amount in the base unit of the currency, typically cents for Australian Dollars. E.g. \$104.23 will be 10423. This amount added to the “baseamount” equals the “amount”.

1.2.5.3surrate

The surcharge rate in percent used to calculate the surcharge. E.g. 1% = “1”.

1.2.5.4surfee

The surcharge fee added to the amount amount in the base unit of the currency, typically cents for Australian Dollars. E.g. \$104.23 will be 10423.