CommSecure

COMMSECURE PAYMENTS

Web Interface Documentation

Version: 3.8

Last Modified: 28 June 2002

CommSecure Australia Pty Ltd
ACN: 001 950 670 / ABN: 14 001 950 670
964 Pacific Highway, PYMBLE NSW 2073
Telephone: (02) 9497 4400 Fax: (02) 9497 4499 www.commsecure.com

1. Introduction

1.1. Document Purpose

This is the documentation for the web interface of the CommSecure Payments payment gateway. This document explains how the payment gateway works and how merchants connect to the payment gateway via the internet and make payments.

The intended audience for this document is both decision makers evaluating CommSecure Payments and for management and technical staff who are integrating their web-site to CommSecure Payments.

1.2. Document Overview

Section 2 “Technical Interface Specification” describes what the CommSecure Payments web interface is and how it works.

Section 3 “Setting Up A New Merchant”, Section 4 “Programming at the Merchant Server” and Section 5 “Merchant Tempates” describe the actions required to start making payments via this interface.

The appendix discusses the implications of the impening introduction of CVN and CVV2.

1.3. References

[1] CommSecure Guide to Banking

1.4. Glossary

CVN:	Customer Verification Number
CVV2:	Card Verification Value - 2
HTML:	Hyper-Text Markup Language
HTTP:	Hyper-Text Transfer Protocol
HTTPS:	Hyper-Text Transfer Protocol - Secure
SSL:	Secure Sockets Layer
URL:	Universal Resource Locator

CommSecure Australia
964 Pacific Highway, PYMBLE NSW 2073
Telephone: (02) 9497 4400 Fax: (02) 9497 4499 www.commsecure.com

2. Technical Interface Specification

The CommSecure Payments gateway is a secure, high-availability commerce server for World Wide Web credit card transactions. The web interface to the gateway utilises a unique three-way handshake between the merchant web-site, the customer browser and the CommSecure Payments system. The three-way handshake provides merchants with unrivalled safety and security. The CommSecure Payments gateway interface can provide the following features:

Secure Credit Card Processing: Merchant web-sites direct customers to CommSecure for credit card payment. CommSecure performs all credit card processing on behalf of the merchant. The merchant web-site does not need to receive or process credit cards.

Merchant Specified Look-and-Feel: CommSecure's payment pages are constructed using merchant templates for a consistent look-and-feel between merchant and CommSecure pages.

Secure Credit Card Storage: CommSecure securely stores all credit card transaction details. The merchant does not need to receive or store credit card details.

Transaction Privacy: 128 bit SSL is used to ensure privacy of communications.

Transaction Authentication: Requests to the payment gateway are authorised via a three-way handshake between the customer, the merchant and the gateway.

Transaction Integrity: Digital signatures are used with three-way handshaking to ensure the integrity of transaction messages sent from the gateway. The integrity of messages sent from customer browsers to the gateway is validated by verifying the transaction with the merchant site.

The concept of the three-way handshake is explained in the Section 2.1 "Technical Overview".

2.1. Technical Overview

The CommSecure Payments web interface operates as shown in Figure 1 "CommSecure Payments web interactions".

The customer interacts with the merchant's website to choose goods and/or services. The merchant web-site gathers customer information such as a customer number or customer name, address and so on. The merchant web-site determines the items for which the customer will pay and the payment amount.

When the merchant has sufficient customer information and "shopping trolley" information the customer is directed to the CommSecure Payments clearance web-site via a link or a form. The link or form includes a transaction reference number and the amount the customer will pay.

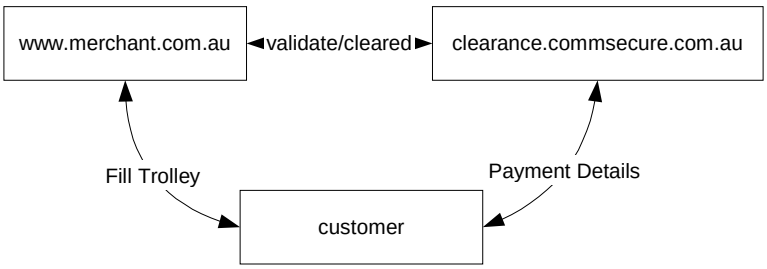

Figure 1 CommSecure Payments web interactions

At the CommSecure Payments clearance web-site the customer is prompted to provide payment details. On receipt of the payment details the CommSecure Payments gateway contacts the merchant with a request to validate that the amount of the payment is correct for the reference number provided. This validation prevents customers changing the payment amount thus ensuring the integrity of the transaction.

On receipt of a positive validation response from the merchant the transaction is sent to the CommSecure Payments gateway and the response from the bank is returned to the merchant's system to confirm when payments have been cleared.

Privacy of the transaction messages is provided by 128 bit SSL. Communications between the customer and the CommSecure Payments system, and between the merchant and the CommSecure Payments system occur over 128 bit SSL.

Communications between the customer and the merchant can be sent over SSL or in the clear as per merchant requirements. The merchant can choose to interact with the customer in the clear because credit card details and any other sensitive transaction data is not sent to the merchant.

2.1.1. Secure Credit Card Processing

The CommSecure system removes the need for the merchant to handle sensitive credit card data on their web-site. Instead, CommSecure Payments handles all credit card details and informs the merchant of the outcome of the transaction.

CommSecure Payments' s credit card clearance server is responsible for:

- Providing secure connections to the banks' clearance facilities.
- Providing a robust and reliable processing environment.
- Establishing secure connections (up to 128-bit SSL) with the customer' s browser.
- Providing redundancy between banks for when the primary bank is down.

- Providing redundancy between power grids.
- Providing redundancy between telecommunications carriers.
- Providing secure credit card storage.

2.1.2. Secure Credit Card Storage

Although merchants do not receive or store credit card details these details can be retrieved from the CommSecure Payments system. Transaction information is stored online for at least three months and then archived.

CommSecure provide customisable web-based reports to enable merchants to query individual transactions or ranges of transactions. For security reasons the credit card number cannot be reported in its entirety. Credit card number in reports are 'gutted' to remove significant middle digits. Gutted credit card numbers are still suitable for tracing transactions.

2.1.3. Merchant Look-and-Feel

The look-and-feel of the customer's interactions with the CommSecure web-site are customized with templates provided by the merchant, to provide a seamless customer experience.

CommSecure enables a seamless join between the merchant's site and CommSecure's site, through the use of merchant-supplied *templates*.

These templates are a slightly modified form of HTML. The modifications allow CommSecure to substitute merchant-supplied data and bank responses into template web pages.

This maintains the look and feel of the merchant's web-site. CommSecure provides the security and reliability of payments, while the merchant provides the content.

2.2. Transaction Processing Interactions

This section contains several different "views" of transaction processing. It is intended to provide background material for what follows. The material in the following sections assumes some basic understanding of the CommSecure Payments system.

2.2.1. Customer's View of a Transaction

a) The customer, having selected some purchases, proceeds to the merchant's checkout.

b) The merchant's checkout provides a link for the customer to follow to CommSecure's clearance server.

c) The customer is presented with a GetCardDetails page personalized by the merchant.

d) The customer supplies credit card details to the clearance server.

e) The customer receives a web page showing the result of the transaction,

and usually containing a link back to the merchant's web-site.

2.2.2. Tracing a successful transaction in detail

Following is a more detailed view of a successful transaction, which also explains how to handle any exceptions that might happen.

a) The customer fills in a shopping cart on the merchant's web-site.

b) The customer selects "go to checkout" or some similar link.

c) The merchant's web-site displays a page with a summary of the customer's purchases and a "pay now by credit card" link.

d) The customer follows this link, which is a link to `https://clearance.commsecure.com.au/cgi-bin/GCDTemplate?[details]`.

e) The clearance server sends to the customer a GetCardDetails page. This page is constructed using the details supplied by the previous step, and using the merchant's `GetCardDetails.i` template.

f) The customer fills in the card details (card number, expiry date, customer name) and submits the form via secure HTTPS to the clearance server.

g) The clearance server validates the payment amount and the merchant's reference number through a HTTP call to the merchant's `Validate` URL (usually on the merchant's web server). This confirms that the transaction originated with the merchant, and that no changes were made to the amount. This also can help defeat denial of payment service attacks on the merchant, by throttling the rate of false or nuisance transactions.

h) The clearance server clears the payment through the bank.

i) The clearance server informs the merchant of the successful payment, through a HTTP call to the merchant's `Cleared` URL.

j) The clearance server sends to the customer a "payment accepted" (or declined) page. This page is generated by combining the details received from the bank, the details provided by the merchant in the initial link to the clearance server, and the merchant's `Accepted.i` (or `Declined.i`) template (stored at the CommSecure clearance server).

The sequence diagram in Figure 2 “Sequence Diagram for Successful Transactions” summarizes these interactions:

CommSecure Australia
964 Pacific Highway, PYMBLE NSW 2073
Telephone: (02) 9497 4400 Fax: (02) 9497 4499 www.commsecure.com

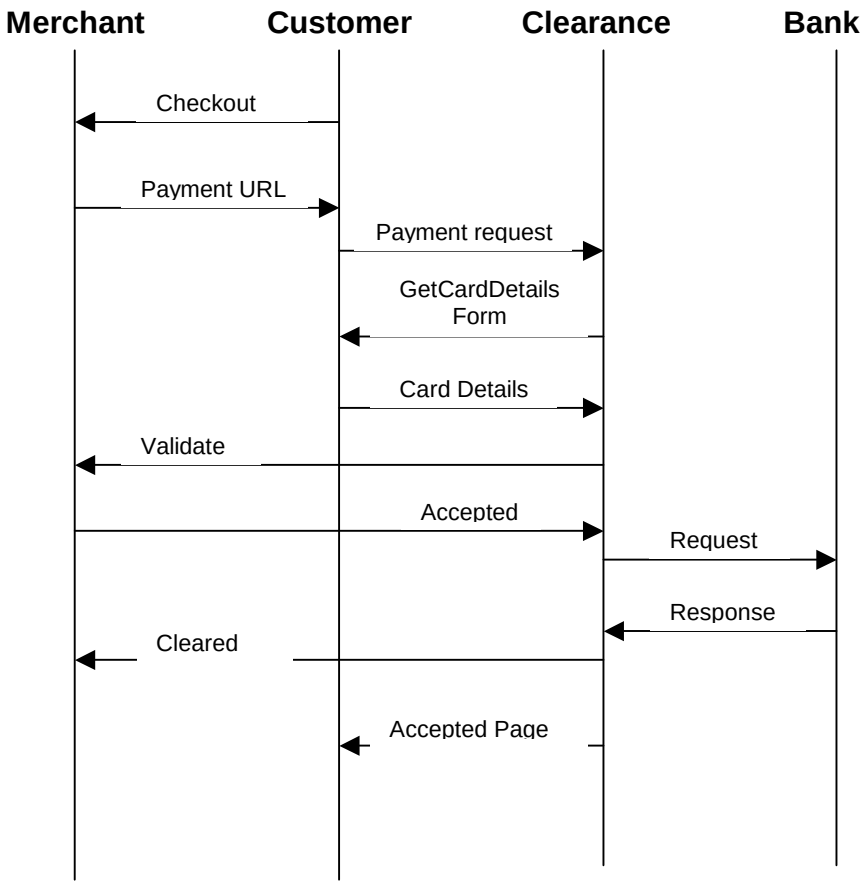

Figure 2 Sequence Diagram for Successful Transactions

CommSecure Australia
964 Pacific Highway, PYMBLE NSW 2073
Telephone: (02) 9497 4400 Fax: (02) 9497 4499 www.commsecure.com

3. Merchant Installation Checklist

A merchant wanting CommSecure credit card clearance hosting must:

1. Set up bank details:

a) Establish a Credit Card Merchant account with a CommSecure-enabled bank.

b) Obtain Merchant number(s) and terminal number(s) as required from the bank.

c) Optionally, set up AMEX and Diners Card clearing facilities and arrange for them to be enabled by the bank.

See [1] for more on these steps.

2. Set up Merchant web/programming:

a) Set up a `Validate` CGI script on the merchant's web server, to validate payment requests (see Section 4.3 “Validate Program”).

b) Set up a `Cleared` CGI script on the merchant's web server, to record payment clearances (see Section 4.3 “Cleared Program”).

c) Obtain and verify a copy of CommSecure's public key. The public key is available as a text file containing two numbers, `exponent` and `modulus`. The CommSecure public key can be obtained from `https://clearance.commsecure.com.au/support/pubkey.txt`, and verified by calling CommSecure technical support on 02 9440 9885.

d) Set up the merchant's "check-out" to produce a link to `https://clearance.commsecure.com.au/cgi-bin/GCDTemplate` (see Section 4.2 “Checkout Program”).

e) Create four HTML templates customized for this merchant. Sample HTML templates are available at `https://clearance.commsecure.com.au/support/templates/`. See Section 5 “Merchant Templates” for a detailed explanation of merchant templates.

CommSecure Australia
964 Pacific Highway, PYMBLE NSW 2073
Telephone: (02) 9497 4400 Fax: (02) 9497 4499 www.commsecure.com

3. Send details to CommSecure:

a) Fill in and sign a **CommSecure Merchant Setup Form**. This will ask you to supply:

- The URLs of the `Validate` and `Cleared` scripts (e.g. `http://www.merchant.com.au/scripts/myValidate`), (Note that URLs are limited to a maximum of 100 characters).
- `AcceptedUrl, DeclinedUrl`: URLs to be used in your templates as links back to your site (see Section 5.3 “Accepted.i, Declined.i”).
- The Merchant numbers and terminal numbers obtained from the bank.
- The HTML templates themselves (normally sent separately by email to `operations@commsecure.com.au`).

b) Send the Merchant Setup Form to CommSecure.

c) CommSecure will supply the merchant with a CommSecure Merchant ID (usually a short name, "`JoesWidgets`"), and with a password to access the transaction reporting web page.

4. Software Certification

a) CommSecure will provide a set of certification tests (see `https://clearance.commsecure.com.au/support/certification`/) to ensure that the Merchant's checkout, `Validate` and `Cleared` programs work properly under all foreseeable circumstances.

b) The merchant will ensure their software works with these certification tests, using the CommSecure test environment. The CommSecure test environment is accessed in the same manner as the production environment but with a different merchant id. The test merchant identifier is the production merchant identifier prefixed by the string “Test”.

c) The Merchant will perform a test purchase transaction with their own credit card and ensure that the money is transferred to their bank account

5. Go live!

CommSecure Australia
964 Pacific Highway, PYMBLE NSW 2073
Telephone: (02) 9497 4400 Fax: (02) 9497 4499 www.commsecure.com

4. Programming at the Merchant Server

4.1. Overview

Merchants need to write three programs, `Checkout`, `Validate` and `Cleared`. These programs are the interface between the merchant's system and the CommSecure Payments system.

The `Checkout` program is invoked when the customer has selected the goods and/or service to purchase and is ready to pay. The checkout program is responsible for determining the amount to be paid and a reference number for the transaction. The `Checkout` program directs the customer to the clearance server and passes the reference and the amount to the clearance server.

The `Validate` program is called by CommSecure Payments to check that the customer has not altered the reference number or the amount. The `Validate` program is responsible for checking the integrity of these values.

The `Cleared` program is called by CommSecure Payments when the transaction is complete. It is passed the results of the transaction.

See `https://clearance.commsecure.com.au/support/web/example/` for some example `Checkout`, `Validate` and `Cleared` programs that you might choose to modify. Alternatively CommSecure provides a full integration service.

4.2. Checkout Program

The merchant' s checkout program outputs HTML that will allow the customer to send payment request details to CommSecure. A small example of this would be:

```
<a href="https://clearance.commsecure.com.au/cgi-
bin/GCDTemplate?MerchantNum=mNum&RefNum=123abc&Amount=4320">
Pay Securely By Credit Card through CommSecure</a>
```

Creating a small form for the customer to submit could also do this, as illustrated below:

```
<form action="https://clearance.commsecure.com.au/cgi-
bin/GCDTemplate" method=post>
<input type=hidden name=MerchantNum value=mNum>
<input type=hidden name=RefNum value=123abc>
<input type=hidden name=Amount value=4320>
<input type=submit value="Pay securely by Credit Card
through CommSecure">
</form>
```

CommSecure Australia
964 Pacific Highway, PYMBLE NSW 2073
Telephone: (02) 9497 4400 Fax: (02) 9497 4499 www.commsecure.com

The required and optional field names (all are case sensitive) to send to the CommSecure system are tabulated below:

Name		Meaning
MerchantNum	Required	Merchant identifier assigned by CommSecure.
RefNum	Required	Reference identifier generated by the merchant. It is recommended (but not required) that this identifier be unique. This identifier can contain up to 20 characters in the ranges "a-zA-Z0-9_"
Amount	Required	The amount of this transaction, in cents, with no dollar sign, e.g. $43.20 is written "4320".
CustNum	Optional	Up to 16 characters of merchant-supplied data.
MemberNum	Optional	Up to 20 characters of merchant-supplied data.
MerchantData	Optional	Up to 1024 characters of merchant-supplied data.
Description	Optional	Text description of intended purchase.

Table 2: Fields supplied to GCDTemplate

The checkout program should record the Amount and RefNum for later use by the Cleared and Validate programs.

The CustNum, MemberNum, and MerchantData fields do not affect the financial transaction, but they may be used to provide more data to the customer and can be used by the merchant to track the transaction.

4.3. Validate Program

Before dispatching a clearance request to the bank, the clearance server validates with the merchant that the supplied RefNum and Amount originated from the merchant.

This validation is performed by sending a HTTP request to a Validate program on the merchant server. The use of the Validate program is optional but its strongly recommended.

The Validate program is called with two HTTP GET parameters, RefNum and Amount, e.g.:
`http://www.merchant.com.au/cgi-bin/Validate?RefNum=123abc&Amount=4320.`

The Validate program checks that refnum RefNum / Amount pair was generated recently by the checkout program and that the RefNum has not already been used in a transaction.

The Validate program should return as output the word Accept or the word Decline (these are case insensitive).

In the event that the Accept is not found, CommSecure assumes a Decline and rejects the transaction.

An example **Validate** program (in Python) is shown below:

```
#!/usr/local/bin/python
import cgi, payments
form = cgi.parse()
refnum, amount = form[RefNum], form[Amount]
print Content-type: text/plain”
print
if payments.validate(refnum, amount):
 print "Accept"
else:
 print "Decline"
```

4.4. Cleared Program

After the clearance server clears (or fails to clear) a payment with the bank, it informs the merchant of the results by calling the merchant's Cleared program. The Cleared program is optional but its use is strongly recommended.

The program is called with up to seven GET parameters: RefNum, Amount, Response, AuthNo, Audit, ErrorMsg, and Sig.

The RefNum and Amount fields are the RefNum and Amount generated by the checkout software, and recently checked by the Validate program. The Cleared program should check that this is the case.

The Response field is either “A” (for "Accept") or “D” (for "Decline") depending on the banks response.

The AuthNo and Audit fields are passed on if provided by the bank. They should be retained to help resolve queries about the transaction. Note, however, that either or both of these fields may be not provided at all.

The ErrorMsg field is the error message from the bank (e.g., "Card Expired").

The Sig field provides a digital signature of the other message parameters, allowing the merchant to verify that the message comes from CommSecure.

When the Cleared program receives a message with a valid signature and with an "Accepted" Response field, the merchant knows that the payment has been accepted.

CommSecure Australia
964 Pacific Highway, PYMBLE NSW 2073
Telephone: (02) 9497 4400 Fax: (02) 9497 4499 www.commsecure.com

Parameter	Meaning
RefNum	Reference number originally generated by merchant's checkout program.
Amount	Amount (in cents) of cleared transaction.
Response	'A' or 'D' (for 'Accepted' or 'Declined')
AuthNo	Authorisation number (optionally) provided by the bank.
Audit	Audit number (optionally) provided by the bank.
ErrorMsg	(Optional) textual error message.
Sig	Digital signature verifying that the other (up to) six fields were provided by CommSecure.

Table 3: Fields provided to Cleared program

4.4.1. Digital Signature

The Sig field must be verified by the `Cleared` program like this:

1 The parameters (except for `Sig`) are combined and URL-encoded into a single string e.g.:
`"RefNum=ab12&Amount=2340&Response=A&AuthNo=439&Audit=34&ErrorMsg=None"`

Fields that have no value are omitted, e.g. if no value were provided for `AuthNo` or `Audit`, the URL-encoded string in our example would be `"RefNum=abc123&Amount=2340&Response=A&ErrorMsg=None"`

The order of the fields is significant. The required ordering is: `RefNum, Amount, Response, AuthNo, Audit, ErrorMsg`

2 An MD5 digest is made of the URL encoded string.

3 The `Sig` field is decoded using base64.

4 The decoded `Sig` is turned into a number by treating each char as a byte (in big endian order). E.g. "abc" would be converted to `ord('a')*256*256 + ord('b')*256 + ord('c')`

5 This number is decrypted using RSA with CommSecure's public key. The decryption is performed as `decrypted = decoded^exponent % modulus.` The CommSecure public key can be obtained from `https://clearance.commsecure.com.au/support/pubkey.txt`, and verified by calling CommSecure technical support on 02 9440 9885.

6 The decrypted result from step 5 should be identical to the digest formed in step 2.

A sample `Cleared` program, including signature verification, is listed on the following pages:

CommSecure Australia
964 Pacific Highway, PYMBLE NSW 2073
Telephone: (02) 9497 4400 Fax: (02) 9497 4499 www.commsecure.com

```
#!/usr/local/bin/python
# $Id: cleared.py,v 1.2 2000/04/03 07:00:36 gary Exp $
# Example cleared cgi script

import cgi, urllib, string, md5, base64

def main():
 fields = form2dict(cgi.FieldStorage())
 checkOriginatedHere(fields['RefNum'], fields['Amount'])
 if hasGoodSig(fields):
      recordTransaction(fields)
 else:
      recordAttack(fields)

def hasGoodSig(fields):
 # Verify that the fields supplied contain a valid
 # signature. 'fields' is a dictionary containing
 # keys of: RefNum, Amount, Response,AuthNo,
 # Audit, ErrorMsg and Sig (Some of these fields
 # are optional).

 # Combine and urlencode the parameters
 # (excluding 'Sig').
 # This will produce something like
 # "RefNum=testRef&Amount=440&Response=A&..."
 params=[]
 PossibleParams=(
      'RefNum','Amount','Response',
      'AuthNo','Audit','ErrorMsg')
 for param in PossibleParams:
      if fields.has_key(param):
           paramValue = urllib.quote_plus(fields[param])
           params.append(param+"="+paramValue)
 data = string.join(params,'&')

 # Create an MD5 digest of the URL parameters,
 # and convert it to a number.
 digest = md5.new(data).digest()
 digestAsNumber = numberFromString(digest)

 # Decode the (base64-encoded) signature,
 # convert it to a number, and decrypt it with RSA
 # and the CommSecure public key.
 sig = base64.decodestring(fields['Sig'])
 sigAsNumber = numberFromString(sig)
 exponent, modulus = getPubKey()
```

CommSecure Australia
964 Pacific Highway, PYMBLE NSW 2073
Telephone: (02) 9497 4400 Fax: (02) 9497 4499 www.commsecure.com

```
    # RSA: decrypted = (plain ^ exponent) % modulus,
    # Python's builtin pow() function does this.
    decryptedSig = pow(sigAsNumber, exponent, modulus)

    return decryptedSig == digestAsNumber

def getPubKey():
    # Return the exponent and modulus of the CommSecure
    # public key.  Stored as two numbers, one number
    # per line, in the file 'CommSecure.pub.key'

    fp = open('CommSecure.pub.key', 'r')
    exponent = long(fp.readline())
    modulus = long(fp.readline())
    return exponent, modulus

def form2dict(form):
    # Given a cgi.FieldStorage object,
    # return simple field->value dictionary
    d = {}
    for k in form.keys():
        try:
            d[k] = form[k].value
        except AttributeError:
            d[k] = form[k][0].value
    return d

def checkOriginatedHere(refnum, amount):
    # Check that the 'refnum, amount' pair supplied
    # was originally produced by our checkout program.
    # Not implemented here.
    pass

def recordTransaction(fields):
    print "Good transaction [%s] received" % fields

def recordAttack(fields):
    print "Invalid signature in [%s] " % fields

def numberFromString(string):
    # Treat each character of 'string' as a byte
    # (in big endian order).
    # Convert to a long and return the converted number.
    num = 0L
    for character in string:
        num = (num << 8) + ord(character)
    return num
```

CommSecure Australia
964 Pacific Highway, PYMBLE NSW 2073
Telephone: (02) 9497 4400 Fax: (02) 9497 4499 www.commsecure.com

```
if __name__ == '__main__':
 main()
```

4.4.2. Checking signatures is not always necessary.

Merchants who dispatch goods immediately (typically electronically), need to check carefully the data provided to the `Cleared` program.

If the merchant does not provide instant dispatch, they may choose to simply use the log report provided from CommSecure to fulfill orders. The log data is available from `https://clearance.commsecure.com.au/cgi-bin/Report.cgi`. Reports can only be accessed using a password supplied by CommSecure.

The merchant could use the data from this page to determine which transactions were accepted, and to ship goods based on that data.

CommSecure Australia
964 Pacific Highway, PYMBLE NSW 2073
Telephone: (02) 9497 4400 Fax: (02) 9497 4499 www.commsecure.com

5. Merchant Templates

5.1. Introduction

The merchant stores at the Clearance Server templates to control the "look and feel" of their customers' experience at the Clearance Server. These templates must be named `GetCardDetails.i`, `Accepted.i`, `Declined.i`, and `Error.i`. Each template is a HTML file which contains some tokens that the Clearance Server will replace with values. Inside the template file, the tokens have the format %%FieldName%%, e.g.:

```
Welcome to %%MerchantName%%'s CommSecure-hosted credit-
card handling.
<P>Please fill in your credit card details to pay
%%Amount%%
```

Tokens can occur in two formats: %%`FieldName`%%, which is replaced directly by the field value, and %%**+**`FieldName`%%, which is replaced by the URL-encoded version of the field value. For example, if the field `ClientData` contains the value `"hello there"`, %%`ClientData`%% will be replaced by `"hello there"`, while %%+`ClientData`%% will be replaced by `"hello%20there"`. The URL-encoded format should be used whenever a field is intended to be used in a URL or as form input, e.g. `<input type=hidden name=MerchantData value="%%+Merchant Data%%">`

CommSecure Australia
964 Pacific Highway, PYMBLE NSW 2073
Telephone: (02) 9497 4400 Fax: (02) 9497 4499 www.commsecure.com

Where a field is used in a template but no value has been supplied for that field, the field is replaced in the output HTML with `"[not supplied]"`, e.g.

```
Thank you for your purchase of %%Description%%
```

would be replaced by

```
Thank you for your purchase of [not supplied]
```

Some sample template files that can be adapted to your needs are available at `https://clearance.commsecure.com.au/support/templates/`

5.2. GetCardDetails.i

The `GCDTemplate` program combines this file with the CGI parameters it receives, and with other parameters supplied by the CommSecure system, to produce a form for the customer to fill in their credit card details.

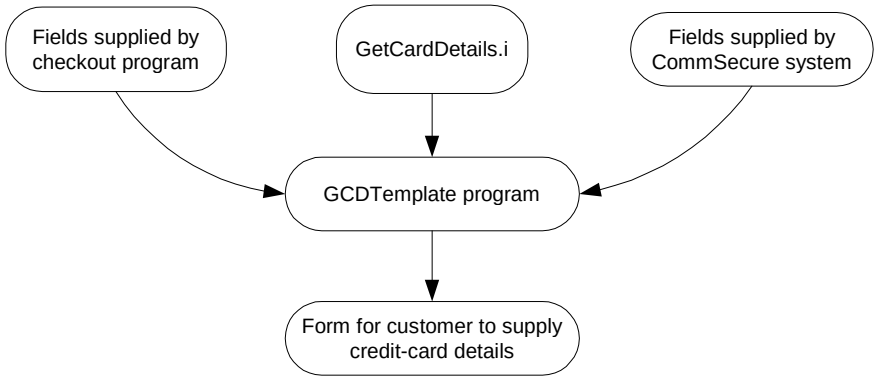

Figure 3 GCDTemplate operation

The fields that can be used in the `GetCardDetails.i` template include:

- The fields supplied by the checkout program (see Table 2: Fields supplied to GCDTemplate).
- The fields supplied by the CommSecure system (`AuthoriseScript, BaseDir, DollarAmount and MerchantName` explained below).

5.2.1. AuthoriseScript

The only requirement of the `GetCardDetails.i` template is that it must contain a form that will ask the customer for some credit card details.

The form must have its `ACTION` attribute set to `%%+AuthoriseScript%%`. The value of `AuthoriseScript` will be set to the URL of the program that will further process the credit-card payment request.

CommSecure Australia
964 Pacific Highway, PYMBLE NSW 2073
Telephone: (02) 9497 4400 Fax: (02) 9497 4499 www.commsecure.com

The form must be sure to prompt the user for the fields NameOnCard, CardNumber, CVN and ExpiryDate. See the Appendix for details on the CVN. Thus a minimal GetCardDetails.i template would include:

```
<form ACTION="%%+AuthoriseScript%%" METHOD=POST>
<br>Name on Card: <input name="NameOnCard">
<br>Card Number: <input name="CardNumber">
<br>Card Verification Number (optional): <input
name=”CVN”>
<br>Expiry Date: <input name="ExpiryDate">
</form>
```

The GCDTemplate program will automatically pass on the fields supplied by the checkout program (e.g. Description) by inserting <input type=hidden...> fields just before the end of the form, e.g.

```
...
<input type=hidden name="Description"
value="%%+Description%%">
</form>
```

The <input type=hidden...> lines are added automatically by the GCDTemplate program without requiring anything in the GetCardDetails.i template.

5.2.2. BaseDir

BaseDir is used to reference images. It is set to a URL from which the GetCardDetails.i file (and other associated files) can be found.

For example, if a GetCardDetails.i file wants to refer to two images "clear.gif" and "blueSquare.gif" that are to be kept in the same directory as the GetCardDetails.i file, the way to do it is:

```
<img src="%%+BaseDir%%/clear.gif">, <img
src="%%+BaseDir%%/blueSquare.gif">
```

Note that we use the %%+FieldName%% syntax, because the field value is being used as a URL, so we may need to URL-encode it.

Although absolute URLs will always work, e.g: <img src=”http://www.yoursite.com/img/logo.gif”>, this method is discouraged because most moden Web Browsers will alert the customer to “Insecure Content” on the secured page.

CommSecure recommends keeping all template images in an “images” directory. In this case, the above HTML code would look like:

```
<img src="%%+BaseDir%%/images/clear.gif">, <img
src="%%+BaseDir%%/images/blueSquare.gif">
```

5.2.3. DollarAmount

This variable is set to the amount to be paid, formatted as `"DD.CC"`

5.2.4. MerchantName

This field is set by the CommSecure system to the full name of the merchant, as registered in the CommSecure Merchant Setup Form.

5.2.5. CommSecure logo

So that customers are aware that their private information is being protected, every template file must include this snippet of html:

```
<a
ref="http://www.commsecure.com.au/htdocs/privacy.html">
<img
src="https://clearance.commsecure.com.au/images/csIcon.g
if">
</a>
```

CommSecure Australia
964 Pacific Highway, PYMBLE NSW 2073
Telephone: (02) 9497 4400 Fax: (02) 9497 4499 www.commsecure.com

5.2.6. Summary of template variables available in GetCardDetails.i

Variable	Supplied by	Details
MerchantNum	Merchant checkout	CommSecure merchant identifier
RefNum	Merchant checkout	Reference number for this payment request
Amount	Merchant checkout	Amount (in cents) of this payment request.
CustNum	Merchant checkout (optional)	Up to 16 chars of merchant-supplied data
MemberNum	Merchant checkout (optional)	Up to 20 chars of merchant-supplied data
MerchantData	Merchant checkout (optional)	Up to 1024 chars of merchant-supplied data
Description	Merchant checkout (optional)	Text description of intended purchase
AuthoriseScript	CommSecure	URL of script to process credit-card details
BaseDir	CommSecure	URL base for relative links
DollarAmount	CommSecure	Amount formatted in DD.CC format
MerchantName	CommSecure	Long name associated with MerchantNum

5.2.7. Example GetCardDetails.i

```
<html><head>
<title>CommSecure Payment for %%MerchantName%%</title>
<base href="%%+BaseDir%%">
</head>
<body>

<!-- CS logo and link -->
<a
href="http://www.commsecure.com.au/htdocs/privacy.html">
```

CommSecure Australia
964 Pacific Highway, PYMBLE NSW 2073
Telephone: (02) 9497 4400 Fax: (02) 9497 4499 www.commsecure.com

```
<img
src="https://clearance.commsecure.com.au/images/csIcon.g
if">
</a>

<!-- Tell customer what's going on -->
Please enter credit details for your purchase of
%%Description%%,
for total value of $%%DOLLARAMOUNT%%,
from %%MerchantName%%.

<!-- Ask for credit-card details -->
<form ACTION="%%+AuthoriseScript%%" METHOD=POST>
<br>Name on Card: <input name="NameOnCard">
<br>Card Number: <input name="CardNumber">
<br>Card Verification Number (optional): <input
name=”CVN”>
<br>Expiry Date (MMYY): <input name="ExpiryDate">
</form>
</body></html>
```

5.3. Accepted.i, Declined.i

Either `Accepted.i` or `Declined.i` is used as a template to show the success (or failure) of the attempted transaction.

The fields available to be used in these templates include:

- The fields supplied by the merchant's checkout (see Table 2: Fields supplied to GCDTemplate)
- The `BaseDir, DollarAmount` and `MerchantName` fields supplied to `GetCardDetails.i`
- `AcceptedUrl, DeclinedUrl`- URLs provided by the merchant to be used as a link back to the merchant's site
- Extra fields containing details about the transaction with the bank, tabulated below:

Field	Meaning
`AuthNo`	Bank's Authorisation number. Only supplied if provided by the bank, which it often isn't.
`TransId`	The Transaction Identifier. A large string of characters identifying this transaction. Usable as a receipt number.
`Response`	"Accepted" or "Declined".

ErrorMsg	The error message, if any.

Table 4: Bank reply supplied fields

The total list of fields available in Accepted.i or Declined.i is thus:

Variable	Supplied by	Details
MerchantNum	Merchant checkout	CommSecure merchant identifier
RefNum	Merchant checkout	Reference number for this payment request
Amount	Merchant checkout	Amount (in cents) of this payment request.
CustNum	Merchant checkout (optional)	Up to 16 chars of merchant-supplied data
MemberNum	Merchant checkout (optional)	Up to 20 chars of merchant-supplied data
MerchantData	Merchant checkout (optional)	Up to 1024 chars of merchant-supplied data
Description	Merchant checkout (optional)	Text description of intended purchase
BaseDir	CommSecure	URL base for relative links
DollarAmount	CommSecure	Amount formatted in DD.CC format
MerchantName	CommSecure	Long name associated with MerchantNum
AcceptedUrl	CommSecure	URL supplied by Merchant in setup form
DeclinedUrl	CommSecure	URL supplied by Merchant in setup form
AuthNo	Bank (optional)	Bank's authorisation number.
TransId	CommSecure	Unique identifier for this transaction
Response	CommSecure	"Accepted" or "Declined".
ErrorMsg	CommSecure	The error message, if any.

CommSecure Australia
964 Pacific Highway, PYMBLE NSW 2073
Telephone: (02) 9497 4400 Fax: (02) 9497 4499 www.commsecure.com

5.3.1. Example Accepted.i Template

A simple `Accepted.i` might include a section looking like this:

```
…
...
<!-- CS logo and link -->
<a
href="http://www.commsecure.com.au/htdocs/privacy.html">
<img
src="https://clearance.commsecure.com.au/images/csIcon.g
if">
</a>

<H1>Payment Accepted</H1>
<P>Thank you, your payment for $%%DollarAmount%%
for purchase reference %%RefNum%% was successful.
Please note the bank's Authorisation Number %%AuthNo%%
in case you want to make further queries.
<P>Thank you for shopping with %%MerchantName%%.
<p>Please <a href="%%+AcceptedUrl%%">Return to our
site</a>
...
```

5.3.2. Example `Declined.i` *Template*

```
...
<!-- CS logo and link -->
<a
href="http://www.commsecure.com.au/htdocs/privacy.html">
<img
src="https://clearance.commsecure.com.au/images/csIcon.g
if">
</a>

<H1>Payment Declined</H1>
<P>We are sorry, but your payment for $%%DollarAmount%%
for purchase reference [%%RefNum%%] was declined.
<p>The reason supplied was: %%ErrorMsg%%
<p>Please <a href="%%+DeclinedUrl%%">Return to our
site</a> and try again?
...
```

5.4. Error.i

This template is used when anything goes wrong during processing of the transaction.

This includes a system problem at CommSecure, or missing or malformed

template files.

All merchants should provide such a page. It should provide a contact email address for reporting problems, as well as a link back to the merchant's web-site.

The fields available for use in the `Error.i` template are:

Field	Meaning
`BaseDir`	URL base for relative links
`ErrorMsg`	Textual description of error
`MerchantName`	Full name of merchant

5.4.1. Sample `Error.i`

```
    ...
    <!-- CS logo and link -->
    <a
    href="http://www.commsecure.com.au/htdocs/privacy.html">
    <img
    src="https://clearance.commsecure.com.au/images/csIcon.g
    if">
    </a>
    <H1>System Error</H1>
    <P>We are sorry, but your payment to %%MerchantName%%
    has failed.
    <p>The error was: %%ErrorMsg%%
    <p>Please contact us at <a
    href="mailto:help@merchant.com">help@merchant.com</a>.
...
```

Apendix

CVN is an additional security code. It is a number that is up to 4 digits long and is printed separately to the embossed card number. Visa and Mastercard cards will have the CVN (called a CVV2) printed in the signature area of their card. American Express cards will have the CVN printed above and to the right of the imprinted card number on the front.

The CVN (Card Verification Number) is a number that is printed (but not embossed) on the card. The CVN is not stored on the magnetic stripe and the manual credit card machines cannot pick them up (because they are not embossed). The CVN will be used for phone and online transactions and should prove that the person is the actual card holder, rather than someone who has either read the magnetic stripe or has stored a copy of the manual credit card voucher. This is in the process of being phased in.

CommSecure Australia
964 Pacific Highway, PYMBLE NSW 2073
Telephone: (02) 9497 4400 Fax: (02) 9497 4499 www.commsecure.com