



Programming for Netregistry's eCommerce Gateway

DOCUMENT ISSUE AND AMENDMENT STATUS

Revision No.	Issue Date	Author(s)	Revision summary
V1.0	07/09/00	Adam Williams Bernard Schmitz Adam Kennedy	First generation document containing instructions for internally and externally hosted merchants to program transactions through the e-commerce gateway.
V1.0a	12/05/01	Brian McCormack	Modification to Section 1.4 (terms changed)
V2.0	07/04/04	Angelina Potapova	Added new sections: "SECTION 3: Programming in ASP" and "SECTION 4: Programming in PHP"
V3.0	06/10/06	Angelina Potapova	Removed section: "SECTION 3: Programming in ASP". Added sections: "SECTION 3: Programming in ASP/.NET" and "SECTION 5: FAQ"
V4.0	19/07/07	Verity Meagher	Removed Commonwealth Bank (CommWeb) from "SECTION 1.1"
V5.0	11/10/07	Ross Annetts Jamie Mckean Karla Oliveira Verity Meagher	Modified all sections of the document; specifically consolidating duplicated content for externally hosted merchants in previous Sections 2, 3, 4, update FAQ's and Section 1.2.
V6.0	15/10/07	Ross Annetts Jamie Mckean Verity Meagher	Added ASP.NET to Section 3, amended the Transaction Types on page 9.
V6.2	20/11/07	Ross Annetts	Added content to FAQ. Multicurrency. AMEX, DINERS and JCB.

CONTENTS

SECTION 1: INTRODUCTION.....	4
1. What is the e-commerce gateway?.....	4
2. How does one become a merchant?.....	4
3. Benefits of being a merchant.....	4
4. How much does the e-commerce gateway cost to use?.....	4
SECTION 2: STANDARD PROGRAMMING FOR NETREGISTRY PAYMENT GATEWAY	5
1. Method	5
2. Example of a status command result	7
3. Example of a purchase/refund result	7
SECTION 3: ALTERNATIVE METHODS FOR NETREGISTRY HOSTED CUSTOMERS	9
PROGRAMMING IN CLASSIC ASP & ASP.NET	9
1. Transaction types	9
Example Classic ASP/VB script	10
Example ASP.NET script	10
PROGRAMMING IN PHP	11
1. Transaction types	11
PARTY TRANSACTION	11
PROGRAMMING IN CGI/PERL.....	13
1. Transaction types	13
SECTION 4: Frequently Asked Questions.....	16
1. Preauth(orisation).....	16
2. MultiCurrency	17
3. AMEX, DINERS and JCB	17
4. Test credit cards and prefixes	18

SECTION 1: INTRODUCTION

This document lists the different commands available to internally and externally hosted merchants for programming Netregistry's e-commerce gateway.

1. What is the e-commerce gateway?

In simple terms, the e-commerce gateway is a secure network through which merchants may facilitate online credit card transactions with either St-George (Ingenico) or the National Australia Bank (Quest).

2. How does one become a merchant?

a) Purchase one of Netregistry's three basic e-commerce packages. Details of these packages, including costs, may be found at: <http://www.Netregistry.com.au/ecommerce>

Business Hosting – Internally hosted Web site where customers make secure online credit card orders for processing through the business's existing offline system.

Merchant eCommerce – Internally hosted Web site where customers perform secure online credit card transactions, processed in real-time (less than six seconds).

Payment Gateway – Externally hosted Web site where customers perform secure online credit card transactions, processed in real-time (less than six seconds).

b) You will also need to contact either St George Bank or National Australia Bank to set up a merchant account. Once the bank creates this account, please send an email to support2@Netregistry.com.au with the following details:

St George Merchant: Your Merchant ID and Terminal ID

Or

NAB: Your EB Number and Terminal ID

Please note: If you intend to process transactions in currencies other than AUD, request a NAB Multi-Currency account.

3. Benefits of being a merchant

Using Netregistry's e-commerce gateway, merchants gain the ability to perform secure online credit card transactions without the need for an EFTPOS terminal.

Because customers pay for products up-front, merchants can integrate their Web site and sales database so these orders are automatically fulfilled and delivered.

Similarly, two merchants can easily link their Web sites – allowing for cross-promotions, etc.

4. How much does the e-commerce gateway cost to use?

Unlike our competitors, Netregistry charges no per transaction fee – all you pay for is access to the Payment Platform (either by subscribing to "Payment Gateway" or "Merchant eCommerce" packages)

SECTION 2: STANDARD PROGRAMMING FOR NETREGISTRY PAYMENT GATEWAY

1. Method

The following is the standard method for customers with both internally and externally hosted Netregistry payment gateways. It can be used in conjunction with all scripting languages, as it uses a HTTPS post.

The HTTPS method comprises of two parts:

a) Your Web site

Your Web site is responsible for collecting data from the browser in a secure manner, filtering and validating it. From inside your site, you make an HTTPS call to an External Access Server (EAS).

b) external2.pl

This access program validates the login, validates all data passed to it and makes the call to the next tier down. It then formats a response to the calling program in one of a standard set of formats

The stages of a transaction:

1. HTML Form calls your Web site
2. Your Web site validates the input data
3. Your Web site makes an external https call to an EAS
4. EAS validates login
5. EAS validates input (card number, exploit protection, etc.)
6. EAS directly contacts Internal Access Server (IAS) to process card
7. IAS contacts banks and receives bank response
8. IAS returns response to EAS.
9. EAS formats reply to the caller's response type and replies
10. Your Web site displays response to user, and handles order completion

The URL to use is <https://4tknox.au.com/cgi-bin/themerchant.au.com/ecom/external2.pl>

The fields are:

Field	Description	Example
LOGIN	Your Merchant ID (MID) number, a forward slash "/" and your external access password.	LOGIN=24/wwjkvfmdqj7xj74y
AMOUNT	Amount in dollars	AMOUNT=12.00
CCNUM	Credit Card number	CCNUM=5444471216129177
CCEXP	Credit card expiry date (format: mm/yy)	CCEXP=01/08

Programming for Netregistry's eCommerce Gateway

COMMENT	Optional comment for transaction log table	COMMENT=Refund%20error%20in%20payment.
TXNREF	16-digit transaction reference number	TXNREF=0007311009230043
COMMAND	Command you want to perform (purchase, refund, status, preauth, completion) If you don't send the COMMAND field, a "purchase" will be assumed	COMMAND=refund
PREAUTHNUM	Preauthorisation number (See Preauth(orisation) section for more details)	PREAUTHNUM=123456

COMMAND variables and their required fields are:

Variable	Fields
purchase	AMOUNT, CCNUM, CCEXP
refund	TXNREF, AMOUNT
status	TXNREF
preauth	CCNUM, CCEXP, AMOUNT
completion	PREAUTHNUM, CCNUM, CCEXP, AMOUNT Note: "AMOUNT" must be the same as the PreAuthorisation amount.

All commands can use the COMMENT field.

All commands **require** the LOGIN field.

Examples:

```
https://4tknox.au.com/cgi-bin/themerchant.au.com/ecom/external2.pl?LOGIN=1000/blahblah&COMMAND=purchase&AMOUNT=1234.56&CCNUM=4111111111111111&CCEXP=01/01&COMMENT=An%20example
```

```
https://4tknox.au.com/cgi-bin/themerchant.au.com/ecom/external2.pl?LOGIN=1000/blahblah&COMMAND=refund&AMOUNT=1234.56&TXNREF=1234567890123456&COMMENT=An%20example%20refund
```

```
https://4tknox.au.com/cgi-bin/themerchant.au.com/ecom/external2.pl?LOGIN=1000/blahblah&COMMAND=status&TXNREF=0007310935130098
```

Programming for Netregistry's eCommerce Gateway

The return values are as below. The first line will be approved, declined or failed. Only if the transaction is approved will the transaction go through. If it is declined, then the bank declined the transaction for some reason. Any other response is a failure. The word complete is returned for a status request.

The second line holds the RRN for a Quest transaction. It is blank for an Ingenico transaction.

For Quest transactions, a receipt is returned. It is terminated by a period "." on a single line. Ingenico does not return a receipt.

After the "." there are several fields which may be of interest. `txn_ref` is a unique transaction reference number. This should be kept as a reference. `response_text` and `reponse_code` may also be of interest.

The only fields of interest are `status`, `bank_ref`, `txn_ref`, `response_code`, `response_text`, `card_type`, `card_desc` and `result`. All other fields returned by Quest and Ingenico will be removed in the future.

2. Example of a status command result

complete

```
.  
card_number=411111xxxx1111  
settlement_date=31/07/00  
response_text=INVALID TRANSACTION  
amount=100  
status=complete  
txnref=0007311428202312  
bank_ref=000731000024  
card_desc=VISA  
response_code=12  
card_expiry=01/01  
MID=24  
card_type=6  
time=2000-07-31 14:28:20  
command=purchase  
result=0
```

3. Example of a purchase/refund result

```
declined  
000782000024  
Transaction No: 00332546  
-----  
TYRELL CORPORATION  
  
MERCH ID          99999999  
TERM ID           Y9TB99  
COUNTRY CODE AU  
31/07/00          14:32  
RRN               000782000024  
VISA  
411111-111  
CREDIT A/C        01/01  
  
AUTHORISATION NO:  
DECLINED 12  
  
PURCHASE          $1.00
```

Programming for Netregistry's eCommerce Gateway

TOTAL AUD \$1.00

PLEASE RETAIN AS RECORD
OF PURCHASE

(SUBJECT TO CARDHOLDER'S
ACCEPTANCE)

.
training_mode=0
pld=0
approved=0
settlement_date=31/07/00
transaction_no=332546
status=declined
version=V1.0
operator_no=22546
refund_mode=0
merchant_index=24
response_code=12
receipt_array=ARRAY(0x8221b9c)
cashout_amount=0
account_type=CREDIT A/C
rrn=000782000024
response_text=INVALID TRANSACTION
txn_ref=0007311458332546
card_no=4111111111111111
total_amount=100
card_desc=VISA
card_expiry=01/01
card_type=6
result=0

SECTION 3: ALTERNATIVE METHODS FOR NETREGISTRY HOSTED CUSTOMERS

PROGRAMMING IN CLASSIC ASP & ASP.NET

1. Transaction types

Both internally and externally hosted merchants have the ability to make purchases, provide refunds and obtain transaction statistics.

These functions are programmed using the commands `purchase`, `refund` and `status`. (Note that these can also be carried out through our admin site [<https://theconsole.Netregistry.com.au/>].)

Transactions are processed using `Payment.NRGatewayClient` object, which was written specifically for transaction processing, for internally hosted customers. Check the example classic ASP script on the next page. The line which is commented out will allow you to check status of a given transaction reference number.

The breakdown of required specified values is as follows:

Field	Description	Example
MID	Your Merchant ID (MID) number, a 3 digit number issued by Netregistry	123
PASSWORD	Merchant password supplied by Netregistry	a1b2c3n4bsaf25kd01k2232kk
AMOUNT	Amount in dollars	AMOUNT=12.00
CCNUM	Credit Card number	CCNUM=5444471216129177
CCEXP	Credit card expiry date (format: mm/yy)	CCEXP=01/08
COMMENT	Optional comment for transaction log table	COMMENT=Refund%20error%20in%20payment.
CURRENCY	AU by default, but users with MultiCurrency gateways will be able to specify alternatives	AU
TXNREF	Transaction reference	1234567890123456

There are three possible results from making a transaction:

Programming for Netregistry's eCommerce Gateway

- **"approved"** indicates the transaction was successful.
- **"declined"** indicates the transaction was declined at the bank for some reason. (See the `ecom_response_text` command below).
- **"failed"** indicates there was a problem with connecting to the Gateway. This can be because there is a connection problem at Netregistry, an incorrect Merchant ID / password combination was entered or the Merchant account has not yet been setup and set to "live" mode by Netregistry. This is a rare occurrence. The majority of "failed" responses are caused by incorrect passwords or Merchant ID's. If you continually receive "failed" status, please double-check your password - if there is still a problem contact Netregistry Technical Support.

Example Classic ASP/VB script

```
<%  
  
dim payment  
set payment = CreateObject("Payment.NRGatewayClient")  
if err.number <> 0 then  
    response.write "Error " & Err.number & ":" & Err.description  
    else payment.setCredentials MID, "PASSWORD"  
    dim result  
    set result = payment.Purchase("CCNUM", "CCEXP", "AMOUNT", "CURRENCY", "COMMENT")  
    'set result = payment.Status("TXNREF")  
    response.write result("status")  
end if  
  
%>
```

Example ASP.NET script

The ASP.NET example is currently in development and will be available in a future update of the payment gateway documentation.

PROGRAMMING IN PHP

1. Transaction types

Both internally and externally hosted merchants have the ability to make purchases, provide refunds and obtain transaction statistics.

These functions are programmed using the commands `purchase`, `refund` and `status`. (Note that these can also be carried out through our admin site [<https://theconsole.Netregistry.com.au/>].)

NRGatewayClient.php

The NRGatewayClient.php is a class written in PHP and available on the Netregistry infrastructure, for Netregistry hosted customers only. It processes a transaction and returns a standard array of key values from our gateway plus a bank specific receipt where appropriate. Note: it is NOT available for download for security reasons.

The NRGatewayClient.php has the following functions:

Command	Expected Parameters
<code>purchase</code>	merchant, password, cardno, cardex, amount , comment
<code>preauth</code>	merchant, password, cardno, cardex, amount , comment
<code>threeparty_start</code>	amount, return_url , comment
<code>threeparty_complete</code>	dr* , comment
<code>refund</code>	txnref, amount , comment
<code>status</code>	txnref , comment

*dr=digital receipt. The digital receipt is passed by the 3rd Party Gateway to Netregistry.

Each function expects a set number and type of variable in the order listed above. The required variables are in bold. The card number must have no spaces, the expiry date in dd/mm/yy format and the amount in 00.00 format. Each function will return a standard result set in an array. If you are passing the result array between pages, it may be necessary to serialize prior to sending and un-serialize the value in the next page to be able to access the array values. For further information, refer to your PHP guide.

To utilize the NRGatewayClient functionality, include it in any PHP pages that will interact with the NRGatewayClient.php.

Example in PHP:

```
require ("NRGatewayClient.php");
```

PARTY TRANSACTION

In this example, we will interact with the NRGatewayClient to execute a 2 party transaction and return a result. You must always create an instance of the NRGatewayClient before interacting with it. Pass your Merchant ID and password to the class.

Example in PHP:

```
$NRGatewayClient = new NRGatewayClient ($merchant, $password);
```

To make a payment transaction, invoke the Purchase function and pass the following variables in the following order:

1. Credit Card number
2. Credit Card expiry date
3. Amount
4. Comment

The compulsory variables are **card number**, **expiry date** and **amount**. This function will return a result in an array of values and a bank specific receipt where appropriate. Refer to Appendix 1 for the result array.

Example in PHP:

```
$Result = $NRGatewayClient->Purchase($cardnumber, $cardexp, $amount, $comment);
```

To display a confirmation page with transaction results, refer to the receipt array in Appendix 1.1 and display the results as required in your confirmation page.

Example in PHP:

```
echo $Result["status"]
```

Example Transaction in PHP:

```
<?php

// pass merchant id + password to Netregistry client class and process payment
to bank

require ("NRGatewayClient.php");
$merchant = 'XXX';
$password = 'YYYYYYYYYYYYYYYYYY';

$NRGatewayClient = new NRGatewayClient ($merchant, $password);
$cardnumber = '4111111111111111';
$cardexp = '08/08';
$amount = '1.00';
$comment = 'test';
$Result = $NRGatewayClient->Purchase($cardnumber, $cardexp, $amount, $comment);
echo $Result["status"];
?>
```

PROGRAMMING IN CGI/PERL

1. Transaction types

Both internally and externally hosted merchants have the ability to make purchases, provide refunds and obtain transaction statistics. These functions are programmed using the commands `purchase`, `refund` and `status`. (Note that these can also be carried out through our admin site [\[https://theconsole.Netregistry.com.au/\]](https://theconsole.Netregistry.com.au/).)

The Perl script method comprises three parts:

a) Your script

Your cgi script collects and checks the input data. It hands the data off to a subroutine that processes the transaction and returns a result. Your script then interprets the result, performs any necessary actions and prints the results to the screen.

b) `ecom.cfg`

This file can be added to your `/cgi-bin/ecom` directory of your Web site upon request from our Technical Support Team. It contains essential configuration information. Please note you cannot download or modify this file.

c) `ecom-lib.pl`

This is the system library that contains the subroutine you will use to process the credit card. To process the transaction you use the following function:

```
CC::Process2( $command, @args );
```

where `$command` is a string whose possible values are either `'purchase'`, `'refund'`, `'status'`, `'preauth'`, or `'completion'`.

Each command takes a variable number of parameters. These are, in order:

Command	Parameters
purchase	Card number, card expiry date, amount in dollars and comment (optional)
refund	Transaction reference number, amount in dollars and comment (optional) Note: Amount in dollars cannot exceed the value associated with the transaction reference #.
status	Transaction reference number and comment (optional)
preauth	Card number, card expiry date, amount in dollars and comment (optional) Note: PreAuthorisation command is only available for Merchants using St-George payment gateway. See section Section 5: FAQ: Preauth(orisation) section for more details.
completion	Preauthorisation number, card number, card expiry date, amount in dollars (must be the same as the preauthorisation amount) and comment (optional)

The `CC::Process2()` function also requires the variables: `$CCCONFIG::MID`, `$CCCONFIG::password`, and `$CCCONFIG::loaded`, which are provided in the `ecom.cfg` file.

Note: MID (Merchant ID) and password can both be found in TheConsole's "Merchant Facilities" section for your domain name. Click "Click here to view Merchant Details" to view this information.

The `CC::Process2()` function returns a reference to a hash with the following fields:

Reference	Fields
status	Can be failed , approved or declined
result	Can be -1 , 0 or 1
response_code	Response from the bank
response_text	Response text from the bank
txn_ref	Transaction reference number

In addition, the `status` command will return:

bank_ref	Bank reference number

Programming for Netregistry's eCommerce Gateway

settlement_date	Settlement date of the transaction (according to the bank)
card_type	Credit card type code
card_desc	Credit card type (text)

The only fields of interest are `status`, `bank_ref`, `txn_ref`, `response_code`, `response_text`, `card_type`, `card_desc` and `result`. All other fields returned by Quest and Ingenico will be removed in the future.

Using the `status` and `result` fields, your script can determine whether the transaction was successful. "approved" is returned for a transaction accepted by the bank, "declined" for a transaction that is not accepted, "failed" for a transaction where an error has occurred.

Transaction steps

1. HTML form calls user-written Perl script
2. Perl script validates the input
3. Perl script "require"s `ecom.cfg`
4. Perl script "require"s `ecom-lib.pl`
5. Perl script calls `&CC::Process` (in `ecom-lib.pl`)
6. `CC::Process` validates the values passed to it
7. `CC::Process` validates your user profile in `ecom.cfg`
8. `CC::Process` contacts eCommerce gateway and processes payment
9. `CC::Process` formats and returns results
10. Perl script acts on results
11. Perl script displays output to browser

Examples

```
$rv = &CC::Process2('purchase', '4111111111111111', '01/01', 100.00);  
  
$rv = &CC::Process2('status', '0123456789012345', 'just check this  
transaction!');
```

SECTION 4: Frequently Asked Questions

1. Preauth(orisation)

If a customer wants to use preauth for their gateway, please note: **only StGeorge allows preauth.**

To enable preauth transactions on a gateway, notify the customer to first contact the vendor of their merchant facility (STGeorge) to have it switched on at their end, and then contact us, so that we can re-log their merchant interface on (which will enable the preauth).

Frequently asked questions by customers regarding preauth:

Q: *The command 'completion' requires the preauthnum field. Where do I get the value for this field from? Is this the transaction number received from a preauth command?*

A: Yes – the format and content of the pre-auth and completion commands is shown on Page 5 of this document.

Q: *I've been sending preauth commands and the response I get is 'NOT SUPPORTED'. Is this due to the invalid card number 4111111111111111 or is it due to something else?*

A: Could be either, the client (Merchant) needs to get the bank to setup the ability to do pre-authorization, once they have done this then the bank will tell us and we will make a slight change to the setup. Also you need to submit the card number and the transaction number to complete the pre-auth (as detailed in question 1 of this FAQ)

Q: *Can you please give me the complete list of response codes (ie. W6, 31, etc) and their definitions?*

A: There are quite a lot of these and we parse the majority of them before they are passed to you as they are system level errors that can only be action-ed by our systems administrators. The two responses you need to be concerned with are “approved” and “declined”. All responses codes that you will see do have an accompanying response text.

Q: *I'm an ASP client and want to implement preauthorisation on my gateway service. Your documentation only mentions preauth with CGI/Perl. How can I implement it?*

A: Use the CGI/Perl method. There is no reason why your forms should not be successful using CGI/Perl.

2. MultiCurrency

MultiCurrency is a service offered by Quest (National Australia Bank) **only**. It allows Australian merchants to process transactions in currencies outside of AU. If you wish to enable MultiCurrency on your Merchant account with NAB, please first contact the bank and they will issue you with an alternative set of terminal access numbers. Forward those details to Netregistry and advise which currencies (with currency codes e.g. USD, GBP) you will be using. We will make the necessary modifications to your Gateway setup from this end.

Q: How do I parse the currency to the gateway server?

A: Standard programming for Netregistry Payment Gateway:

Internally and externally hosted customers using the external2.pl, should simply specify the currency after the AMOUNT variable, for example "AMOUNT=10AUD" will deduct \$10AUD, "AMOUNT=10USD" will deduct \$10USD or "AMOUNT=10.00USD" will deduct \$10USD. Note that the amount is in dollar format.

Alternatively, internally (Netregistry) hosted customers can use one of the methods outlined below.

1) PROGRAMMING IN CLASSIC ASP & ASP.NET:

Specify the currency in the **CURRENCY** field as described in **PROGRAMMING IN CLASSIC ASP & ASP.NET** on page 9.

2) PROGRAMMING IN PHP:

Please note that the amount is in dollar format. E.g. 1.00USD will deduct \$1USD.

3) PROGRAMMING IN CGI/PERL:

Please note that for the **purchase** command the amount is in dollar format. E.g. 55NZD will deduct \$55NZD.

3. AMEX, DINERS and JCB

If you would like to take payments from AMEX, DINERS or JCB cards, please note – this is **not** automatically enabled on your Merchant account with your bank. Please follow these instructions on enabling it on your Merchant bank account and your Netregistry Gateway service:

1. Contact AMEX, DINERS and JCB card vendors;
2. Request authorisation to make transactions on those cards. The vendor (AMEX, Diners or JCB) will supply you with appropriate authorisation agreement and letter which you will need to pass onto your Merchant vendor (bank);
3. Contact the appropriate merchant vendor (St-George/NAB/Commonwealth)
4. Supply the authorisation letter from the card vendors to the merchant vendor, who will then modify your Merchant account to accept payments from those card types.
5. Contact Netregistry Customer Service advising what changes have been made to your Merchant account, provide your AMEX Merchant ID (Only for AMEX) and request that we update the Gateway service appropriately.

We will make the necessary changes and advise you by return e-mail.

4. Test credit cards and prefixes

Q: *Is there a way I can send a test transaction?*

A: Yes. Please use the following Credit Card Numbers to confirm that your transactions are successfully being processed:

- Visa: 4111-1111-1111-1111
- MasterCard: 5431-1111-1111-1111
- Amex: 341-1111-1111-1111

In the name field, anything can be used. Please ensure you enter a valid expiry date, ie. six months in the future. A successful transaction should say 'declined', as it is not a live card.

Q: *What are the Credit Card prefix numbers for Visa, MasterCard, Discover and AMEX?*

A: Please refer to the following list of prefixes and numbers.

- Visa: 13 or 16 numbers starting with 4
- MasterCard: 16 numbers starting with 5
- AMEX: 15 numbers starting with 34 or 37