KNET Payment Gateway
Merchant Integration Manual

Documentation and reference guide for integrating KNET Payment Gateway for e-commerce solutions

Doc : K-035 **Date : 15 May 2017** **Version : 2.0**

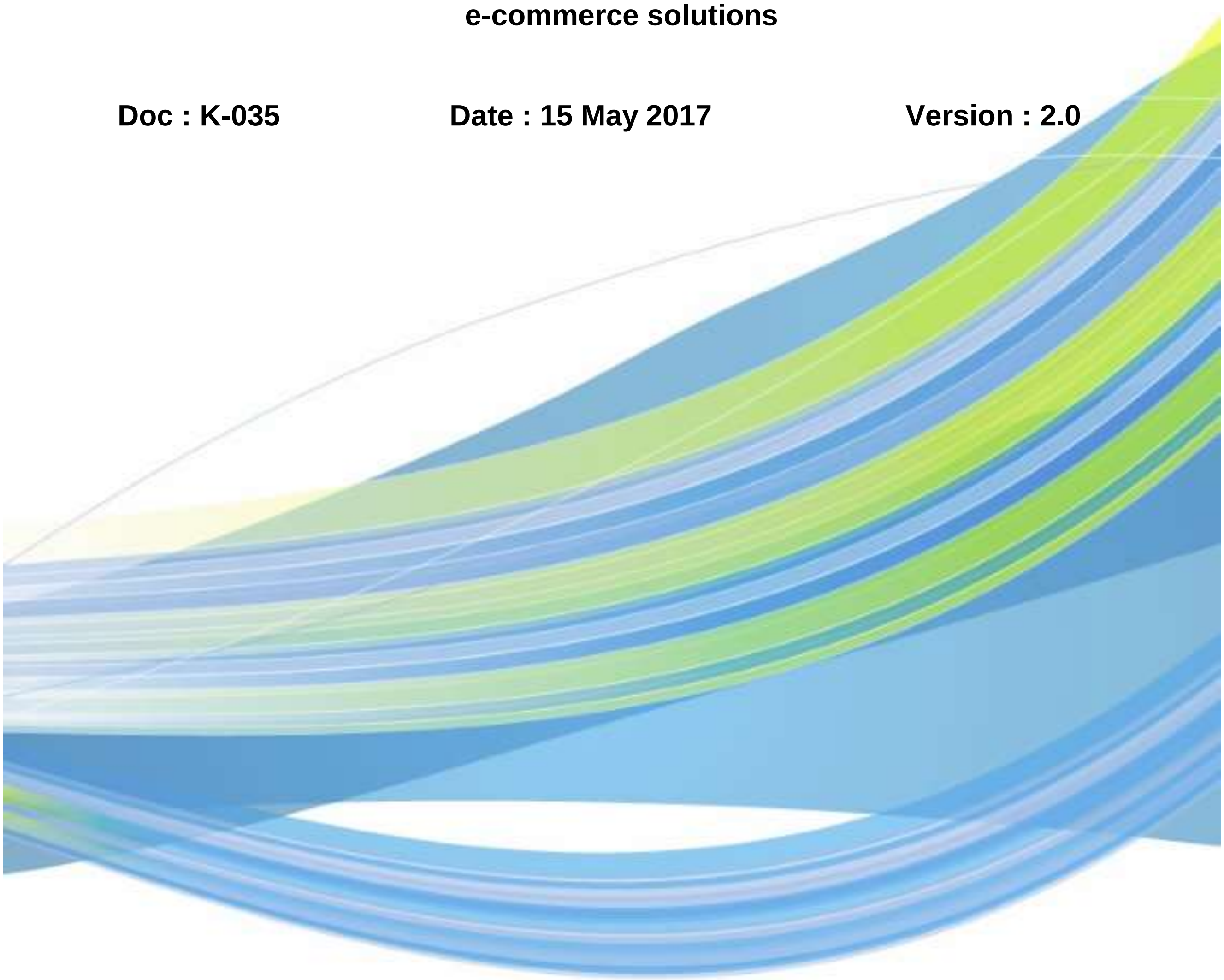


Table of Contents

1. Introduction

KNETPAY is the KNET Payment Gateway solution for Kuwaiti merchants and service companies with e-commerce capabilities. The service will support KNET co-branded ATM card transactions over the web. KNET will introduce credit cards in the future.

The objective of this document is to provide supplementary technical information on merchant website integration with KNET Payment Gateway.

2. Service Overview

The KNET Payment Gateway (PG) service will provide a payment facility for merchant services or goods. The main features of the KNET PG service are:

- Support KNET co-branded debit cards on a secured KNET-hosted payment page
- Online authorization of payment transactions through a merchant-gateway plug-in integration

KNET is using the Applied Communication Inc. (ACI) E24 Commerce Gateway solution.

3. Merchant Pre-requisites

On receiving a Merchant Nomination from one of the Member Banks, KNET requires that the Merchant Questionnaire form (See Appendix A) be completed by the merchant and returned to KNET.

This section describes the minimum technical requirements at the merchant's side.

3.1 Secured web/application server

The merchant must acquire a certificate from a certificate authority in order to do e-commerce transactions with KNET Payment Gateway.

3.2 Plug-ins

KNET supports the following types of software plug-ins:

- ASP
- .Net
- PHP
- Java
- Web API (SOAP)

Based on the Merchant's application, KNET will provide the appropriate plug-in needed to be used on the merchant site.

3.3 Network and SSL

The merchant's web server should have a valid SSL certificate. The KNET server will receive transactions only from SSL certified web servers. KNET is currently running with Verisign Extended Validation SSL. It is the merchant's responsibility to contact an authorized certificate authority and purchase their own certificate.

KNET only accepts the following SSL certificate issuers:

- www.Verisign.com
- www.Thawte.com
- www.geotrusteurope.com
- www.Equifax.com
- www.digicert.com
- www.comodo.com

***Note :** KNET will not provide an SSL certificate for the merchant.*

The Merchant's SSL certificate along with its intermediate and root certificates will be imported on to the KNET Payment Gateway Server.

3.4 Protocol: https

In order to send the authorization details to a merchant web server, the https port of the merchant web server should be made accessible to the KNET Payment Gateway Server. As a matter of fact, KNET will ONLY accept https communication between the KNET Payment Gateway (Port : 443) and a merchant website since the information being exchanged is sensitive (password, terminal ID, etc.)

4. Integration Guide

This section contains installation requirements and instructions for KNET PAY integration.

4.1 Installation Steps

This section outlines the step-by-step instructions for the installation and configuration of the KNET PAY solution at a merchant site:

A. Determine the type of plug-in required by the merchant. The following plug-in types are supported by KNET PAY:

- ASP
- .NET
- PHP
- Java
- Web API (SOAP)

The plug-in decision depends on the type of operating environment being used for the merchant website. KNET and the merchant will determine the appropriate plug-in prior to starting the implementation.

B. Install the appropriate plug-in as determined above. The installation procedure for each of the plug-in types is as follows:

- **ASP:** This plug-in is in the form of a DLL file and needs to be registered on the merchant server using the following on the command prompt:
regsvr32 e24PaymentPipe.dll, THE SERVER MUST BE RESTARTED

- **ASP.NET:**
Toolkit Contents:

I. Binaries (DLLs that will be called by the merchant's application to issue payment transactions).

a. e24PaymentPipe.dll - responsible for sending transaction data from merchant's application to KNET
b. Interop.e24PaymentPipe.dll - proxy DLL which needs to be referenced in the ASP.NET project in order to utilize e24PaymentPipe.dll

II. Sample Application

a. Source Code - A demo ASP.NET/C# Web Application written with Visual Studio 2008 and utilizing e24PaymentPipe.dll
b. Published Project (A compiled version of the demo project ready to deploy in IIS)

B.1 Application Deployment Procedure:

I. Ensure that .NET Framework 2.0 or higher is installed.

II. Ensure that the ‘Bin’ directory contains the following files:
   a. e24PaymentPipe.dll
   b. Interop.e24PaymentPipe.dll
   c. The application’s DLL

III. Register the ‘e24PaymentPipe’ DLL using the command:
C:\Windows\system32\regsrv32.exe e24PaymentPipe.dll

IV. If using 64 bit OS, place the DLL file in the (c:\windows\sysWOW64\) directory and then use the **regsvr32** command to complete the registration.

V. Copy the files under the ‘Published’ folder to ‘C:\Inetpub\wwwroot\<YourAppName>

VI. Run ‘IIS Management Console’ and expand ‘Default Web Site’ node under ‘Local Computer > Web Sites’

VII. Right-click the < YourAppName> node and choose ‘Properties’
   a. Click ‘Create’ button to create an application
   b. In the ASP.NET tab, ensure that
   c. ‘ASP.NET 2.0’ is selected
   d. In the ‘Directory Security’ tab, Click Edit, and check ‘Allow Anonymous Access’

VIII. In ‘Windows Explorer’ right-click the directory containing the merchant resource file and choose properties

a. Go to 'Security' tab
b. Depending on IIS version in use, add the user
    i. NETWORK SERVICE for IIS 6/7
    ii. ASPNET for IIS 5
c. Grant full control over the directory for the added user

- **Java:** This plug-in contains the alias name, merchant's resource file and a class file. The path to the merchant's resource file and the alias should be specified for the plug-in to read and access the file. The class file needs to be copied to the local drive of the merchant server (e.g. the htdocs folder for Apache web server). The path to this file needs to be specified in the web server configuration files of the merchant site. The name of the class file is **e24PaymentPipe.class**.

  There is another class file called ***NotEnoughDataException.class*** that needs to be copied to the htdocs folder. This class provides exception-handling functionality such as checking that the correct number of parameters is being passed.

  The E24 Java plug-in requires the installation of the Java Secure Sockets Extension (JSSE) package or the Java 6 SDK/JRE 1.6 and above which already has JSSE integrated. They can be downloaded from the Sun website at http://java.sun.com. Once downloaded and unpacked, for plain JSSE, it is required to install the following jar files: jsse.jar, jnet.jar and jcert.jar.

  Registration of the SunJSSE provider in the java.security file is needed. Sample code and deployment instructions are provide along with this documentation for integration and reference.

- **PHP:** This plug-in contains the alias name, merchant's resource file and a class file. The path to the merchant's resource file and the alias should be specified for the payment initialization page. Name of the class file is ***e24PaymentPipe.inc.php***, which should be included in the payment initialization page. Sample code and deployment instructions are provide along with this documentation for integration and reference.

- **Web API (SOAP):** Any programing language can be used to build XML SOAP request to initialize the payment. Below are the request and response parameters:

HTTP Parameters		
URLs to Connect	**Production**	https://www.knetpay.com.kw/CGW/servlet/PaymentInitHTTPServlet
	Test	https://www.knetpaytest.com.kw/CGW302/servlet/PaymentInitHTTPServlet
HTTP Version		1.1
Verbs Supported		POST
Content type		application/x-www-form-urlencoded
Char set		utf-8
Compression		NO

Example HTTP Request:	
	POST /CGW302/servlet/PaymentInitHTTPServlet HTTP/1.1 **Host**: www.knetpaytest.com.kw **Content-Type**: application/x-www-form-urlencoded **Cache-Control**: no-cache **id**=116301 &**password**=2pgetcGBUN &**action**=1 &**amt**=15 &**currencycode**=414 &**langid**=ENG &**responseURL**=https://URL.com &**errorURL**=https://URL.com &**trackId**=ABC123
Example HTTP Response:	9787087261562730:https://www.knetpaytest.com.kw:443/CGW302/hppaction?formAction=com.aciworldwide.commerce.gateway.payment.action.HostedPaymentPageAction&
Redirect To Payment Page using above response values :	https://www.knetpaytest.com.kw:443/CGW302/hppaction?formAction=com.aciworldwide.commerce.gateway.payment.action.HostedPaymentPageAction&?PaymentID=9787087261562730

4.2 Merchant Front End Integration:

The plug-in must be integrated to the “Buy” button or equivalent on the merchant website. This usually takes place once the customer has completed shopping and is ready to pay.

5. Transaction Processing

5.1 Message flow

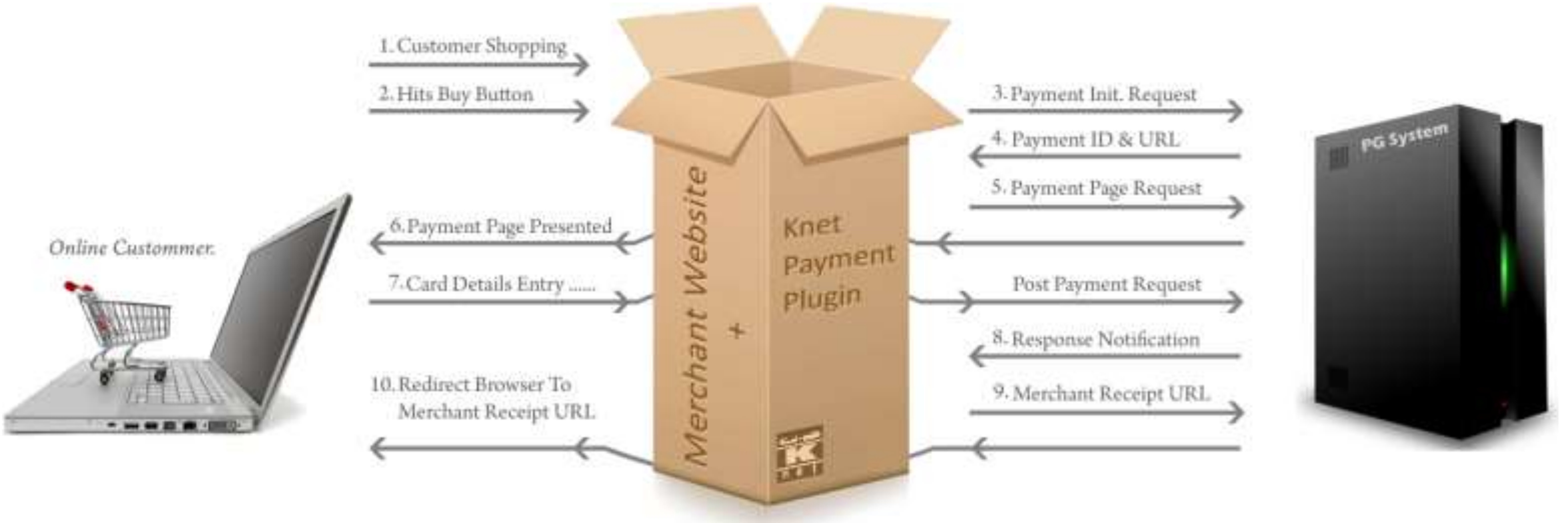


Figure 1. Processing Flow

1) Customer is shopping on the merchant site.
2) Customer finishes shopping; proceeds to check out. Merchant gathers required information from customer, including the total customer bill amount.
3) Merchant performs payment initialization with KNET PG on the required amount.
4) On successful payment initialization, KNET PG replies with the transaction unique payment ID and KNET payment URL.
5) Merchant saves payment initialization parameters with payment ID being the primary key, and redirects the customer to the KNET payment page.
6) KNET builds the payment page and sends it to the customer.
7) Customer enters card information and clicks submit.
8) KNET processes the transaction. If no error occurs, KNET sends a transaction response to the Merchant Response URL specified in payment initialization.

9) Merchant processes and saves the response and sends back their receipt URL to KNET.
10) KNET redirects customer to Merchant Receipt URL.

5.2 Payment Initialization

When the customer checks out from the merchant page, the merchant must send a payment initialization request to KNET as Post method.

Sample Payment Initialization Request Message:

"ResourcePath=D:\\Resource\\&Alias=YourAliasName&action=1&langid=USA&curren
cycode=414&amt=10.00&responseURL=www.merchant.com/response
&errorURL=www.merchant.com/error&trackid=uniquetrackingid&udf1=User Defined Field&udf2=User Defined Field &udf3=User Defined Field&udf4=User Defined Field&udf5=User Defined Field"

Payment Gateway will respond with a single line of text. The values of the response are colon delimited (:). In Payment Initialization, the returned values are the Payment ID and the Payment URL as shown below.

Sample Payment Initialization Response Message:

"PaymentId:PaymentURL"

For the Payment Initialization process, a readymade plug-in (e24PaymentPipe) is provided to facilitate the operation. Information about the plug-in can be found in Section 4.

On the return of a successful payment initialization, the merchant must store the payment ID for future reference (it will be needed to approve the transaction completion) and then redirect the customer to the KNET Payment Page URL defined in the PaymentURL variable. Also, the Payment ID must be sent with the URL as a get request, and the URL must be in an ***encoded*** form.

Successful Payment Initialization Redirect Example:

<Payment URL>?PaymentID=<Payment ID>

If the payment initialization fails, the merchant must redirect the customer to the merchant's error page in the form of an ***encoded*** URL.

Failed Payment Initialization Redirect Example:

<Merchant URL>/error.jsp?PaymentID=<Payment ID>

5.2.1 Payment Transaction Initialization Transmit Message Variables and Definitions

ResourcePath *(STRING type)* - *Resource Path* is the absolute path to the folder which contains the resource file. Resource file will be provided to the merchant with toolkit by KNET.

Alias *(STRING type)* - Alias name is provided by KNET along with the toolkit. It is similar to a userID for the resource file.

Action *(STRING type)* - Must always be in numeric format. ●
Value = 1

amt *(STRING type)*- The Transaction amount should be passed as STRING. The number includes the decimal point (always 3 for Kuwaiti Dinars), e.g 15.750, 10.000,…., etc.

currencycode *(STRING type)*- The currency code of the transaction. For KD, use currency code 414.

langid *(STRING type)*- The language to use when presenting consumer pages. For English, use the string USA. For Arabic, use the string ARA.

paymentid - Payment ID, used to identify the payment for subsequent transactions. ***(Length 14-16 numeric)***

transid - Transaction ID, used to identify the original transaction for a void and capture transaction. ***(Length 14 numeric)***

responseURL ***(STRING type)***- The URL to send the merchant notification request containing the authorization response. The domain name of this URL should match that of the merchant's server certificate.

errorURL ***(STRING type)***- The URL to redirect the consumer browser to, if an error occurs. A track id is passed with the error page to associate it with the particular transaction.

trackid - A unique tracking id issued by the merchant's commerce system which is stored with the transaction. (Avoid spaces and extended characters, only alpha-numeric is to be used). The merchant should initiate such ID to be unique for every transaction irrespective of the customer. The merchant should store trackid on their local database for their records. ***(Length 255 alpha-numeric)***

udf1- udf4 - User Defined Fields 1,2,3,4 can be used to pass and store any additional transaction data required to be archived with the transaction and available as a searching criteria. ***(Length 255 alpha-numeric)***

udf5- User Defined Field 5, can be used to pass and store any additional transaction data required to be archived with the transaction and available as a searching criteria. ***(Length 255 alpha-numeric)***

NOTE: This field has been dedicated for merchants who would like to pass additional customer data and wish to include it in the acquiring bank's report. Refer to section 5.2.2 for more details

When a customer hits the "Buy" button, the customer will be redirected to KNET Hosted Payment Page, where the customer will enter his/her debit card details.

Once the "Pay" button is hit on the KNET Hosted Payment Page, the transaction is routed to KNET system for authorization. The response is returned by KNET system to the website URL specified in the Response URL defined at the time of payment initialization.

The integration process requires customization of the merchant website; hence it needs to be performed by the merchant. To assist the merchant, KNET will provide the merchant with sample ASP/.Net/PHP/JSP pages.

- KNET will provide an **ASP/.NET** sample toolkit with a resource file so the merchant will be able to connect to KNET test environment

- Merchant must do the following once the sample toolkit is received:
  - *Change all URLs in the attached files to reflect merchant's website address*
  - *Change the alias in buy.asp to "____" (MyObj.Alias = "____") KNET will send each merchant a unique alias*
  - *Copy the attached resource.cgn into a specific directory and point to it inside buy.asp (MyObj.ResourcePath = "D:\\resource\\") KNET will send each merchant a unique resource*
  - *Register the e24paymentpipe.dll on the system as follows: CMD> regsvr32 e24paymentpipe.dll*
  - *Provide a copy of the website's SSL certificate for KNET to add it to the KNET system*

- KNET will provide an **PHP** sample toolkit with a resource file in order for the merchant to be able to connect to the KNET test environment

- Merchant must do the following once the sample toolkit is received:
  - *Change all URLs in the attached files to reflect*
  - *merchant's website address*
  - *Change the alias in buy.php to " ---- " (alias = " ----- ")*
  - *Copy the attached resource.cgn into a specific directory and point to it inside $Pipe->setResourcePath*
  - *(resourcePath=D:\\Resource\\)*
  - *Change the URLs in Response.php to reflect the merchant site.*
  - *Provide a copy of the website's SSL certificate for KNET to add it to the KNET system*

- KNET will provide an JSP sample toolkit with a resource file in order for the merchant to be able to connect to KNET test environment

- Merchant must do the following once the sample toolkit is received:
  - *Change all URLs in the attached files to reflect the merchant's website address*
  - *Change the alias in config.txt to "______" (alias = "____") KNET will send each merchant a unique alias.*
  - *Copy the attached resource.cgn into a specific directory and point to it inside config.txt (resourcePath=D:\\Resource\\) KNET will send each merchant a unique resource*
  - *Recompile NotifyServlet.java after modifying its content*
  - *Provide a copy of the website's SSL certificate for KNET to add it to the KNET system.*

Clicking on the "Pay" button will initiate a connection to the payment gateway Server, which will generate a response that is handled by the sample Response page.

5.2.2 Passing additional customer data to acquiring bank report

Some merchants are obliged by the design of their online payment suite to pass the additional customer data for the acquiring bank's purpose. Merchants can use *UDF5* to pass the additional data preceded by the text "ptlf" as follows:

UDF5 = "ptlf amx1234567890"

For example, if the additional customer data is amx1234567890 and merchant would like to see this value in the acquiring bank report, then UDF5 should be in the following format:

UDF5 = "ptlf amx1234567890"

Note: Merchants must not exceed 50 characters including ptlf when using ptlf in UDF5

5. 3 Transaction Response Notification

If payment is initialized and the customer is redirected to KNET payment page, the customer enters his/her card details on the KNET payment page. After the customer submits card information, KNET will process the request and send back the response as a URL ***Encoded*** POST request to the Notification URL specified by the merchant in the payment initialization, namely *responseURL*. The Merchant **MUST** save the returned values by KNET in their database.

Transaction Response Sample:

The response message is formatted differently according to the result value:

- **CAPTURED:** Transaction is approved by the bank
  paymentid=8962225231041180&result=CAPTURED&auth=999554&ref=411857974849&tranid=9929892231041180&postdate=0427&trackid=632186

- **NOT CAPTURED:** Transaction is not approved by the bank
  paymentid=8962225231041180&result=NOT+CAPTURED&auth=999554&ref=411857974849&tranid=9929892231041180&postdate=0427&trackid=632186

- **HOST TIMEOUT:** Bank does not respond within the timeout period
  paymentid=8962225231041180&result=HOST+TIMEOUT&ref=411857974849&tranid=9929892231041180&trackid=632186

- **DENIED BY RISK**: Risk profile to prevent fraud, exp(5trnx/day)
  paymentid=8065439461292910&result=DENIED+BY+RISK&tranid=1221558461292910&trackid=7055476&udf1=1&udf2=1&udf3=1&udf4=1&udf5=ptlf+testing+the+udf5&responsecode=J

- **CANCELED:** PaymentID=1986997010911430&Result=CANCELED

The merchant must then process the response and store the transaction response detail; after which the Merchant Notification URL must write a single line response to the Payment Gateway in the following form:

Merchant Notification URL Response:

REDIRECT=<Merchant Receipt URL>?paymentID=<Payment ID>

Payment Gateway will then read the response from the Merchant Notification URL page and redirect the customer to the Merchant Receipt Page where the merchant will present the customer with the transaction response details.

If for any reason KNET Payment Gateway cannot process the customer transaction, KNET will redirect the customer to the merchant error URL specified in *errorURL* parameter sent in payment initialization.

5.3.1 Transaction Response Transmit Message Variables and Definitions

The response consists of a text string that contains the following data:
PaymentId - Unique order ID generated by Commerce Gateway.

Result - Returned as the transaction response evaluator. The Result should be evaluated to determine if the transaction has been performed successfully after checking for errors.

- CAPTURED - Transaction was approved.
- VOIDED - Transaction was voided.
- NOT CAPTURED - Transaction was not approved.
- CANCELED – Canceled Transaction.
- DENIED BY RISK - Risk denied the transaction.
- HOST TIMEOUT - The authorization system did not respond within the timeout limit.

Auth - The resulting authorization number of the transaction generated by the bank.

Ref - The unique resulting reference number of the transaction generated by KNET. This number or series of letters is used for referential purposes by some acquiring institutions and should be stored properly.

Postdate - Transaction Post Date in the format of “MMDD”. Postdate is used for reconciliation purpose and it can be different than the actual date in some cases.

TransId - Unique transaction ID issued by Commerce Gateway.

TrackId - The Track ID sent by the merchant in the transaction request.

udf1- udf5 User Defined Fields 1,2,3,4,5 can be used to pass and store any additional transaction data (except mentioned below) required to be archived with the transaction and available as a searching criteria.

Merchant must not use the following sym in the TrackId and UDFs

Hack Characters

~	x7E	TILDE
`	x60	LEFT SINGLE QUOTATION MARK, GRAVE ACCENT
!	x21	EXCLAMATION POINT (bang)
#	x23	NUMBER SIGN (pound sign)
$	x24	DOLLAR SIGN
%	x25	PERCENT SIGN
^	x5E	CIRCUMFLEX ACCENT
|	x7C	VERTICAL LINE (pipe)
\	x5C	REVERSE SLANT (REVERSE SOLIDUS)(backslash, backslash)
:	x3A	COLON
'	x27	APOSTROPHE, RIGHT SINGLE QUOTATION MARK, ACUTE ACCENT (single quote)
"	x22	QUOTATION MARK, DIAERESIS
/	x2F	SLANT (SOLIDUS), (slash)

Black Listed Characters

<	x3C	LESS-THAN SIGN (left angle bracket)
>	x3E	GREATER-THAN SIGN (right angle bracket)
(	x28	LEFT PARENTHESIS (open parenthesis)
)	x29	RIGHT PARENTHESIS (close parenthesis)
{	x7B	LEFT BRACE (left curly bracket)
}	x7D	RIGHT BRACE (right curly bracket)
[	x5B	LEFT (SQUARE) BRACKET (open bracket)
]	x5D	RIGHT (SQUARE) BRACKET (close bracket)
?	x3F	QUESTION MARK
;	x3B	SEMICOLON
&	x26	AMPERSAND
*	x2A	ASTERISK

6. Testing and Certification

KNET will go through a testing process to certify a merchants' integration to the KNET Payment Gateway. There are 2 stages of testing to complete the certification process:

- Initial testing
- Final testing

Once the merchant receives the Plug-in & toolkit from KNET after completing the merchant questionnaire, they can start testing with KNET test environment. In case of any problem, the KNET technical staff assigned should be contacted for assistance.

Requirements

Initial testing

- SSL communication between the merchant and KNET server using merchant's test certificate
    - Plug-in (DLL or class file) recognition of merchant server
    - Resource file (resource.cgn) integrity
    - Deployment of demo application included in the toolkit
    - Full transaction cycle on the demo application

Final testing

- SSL communication between the merchant and KNET server using merchant's production certificate
    - Deployment of production application
    - Full transaction cycle on the production application with merchant terminal connected to bank simulator
    - Stress testing that includes the following results:
        - At least 5 CAPTURED
        - At least 5 NOT CAPTURED
        - At least 5 NOT CAPTURED and VOIDED in a "NOT NOTIFIED" case simulation

7. Go-live

A merchant will go live once they have been officially certified. Once certified, KNET creates the terminal on the production, processes the necessary paper work, and sends them to the sponsoring bank via e-mail. The activation of the terminal on the production is subject to the bank’s approval whenever the merchant is done with the final preparations.

A. Appendices

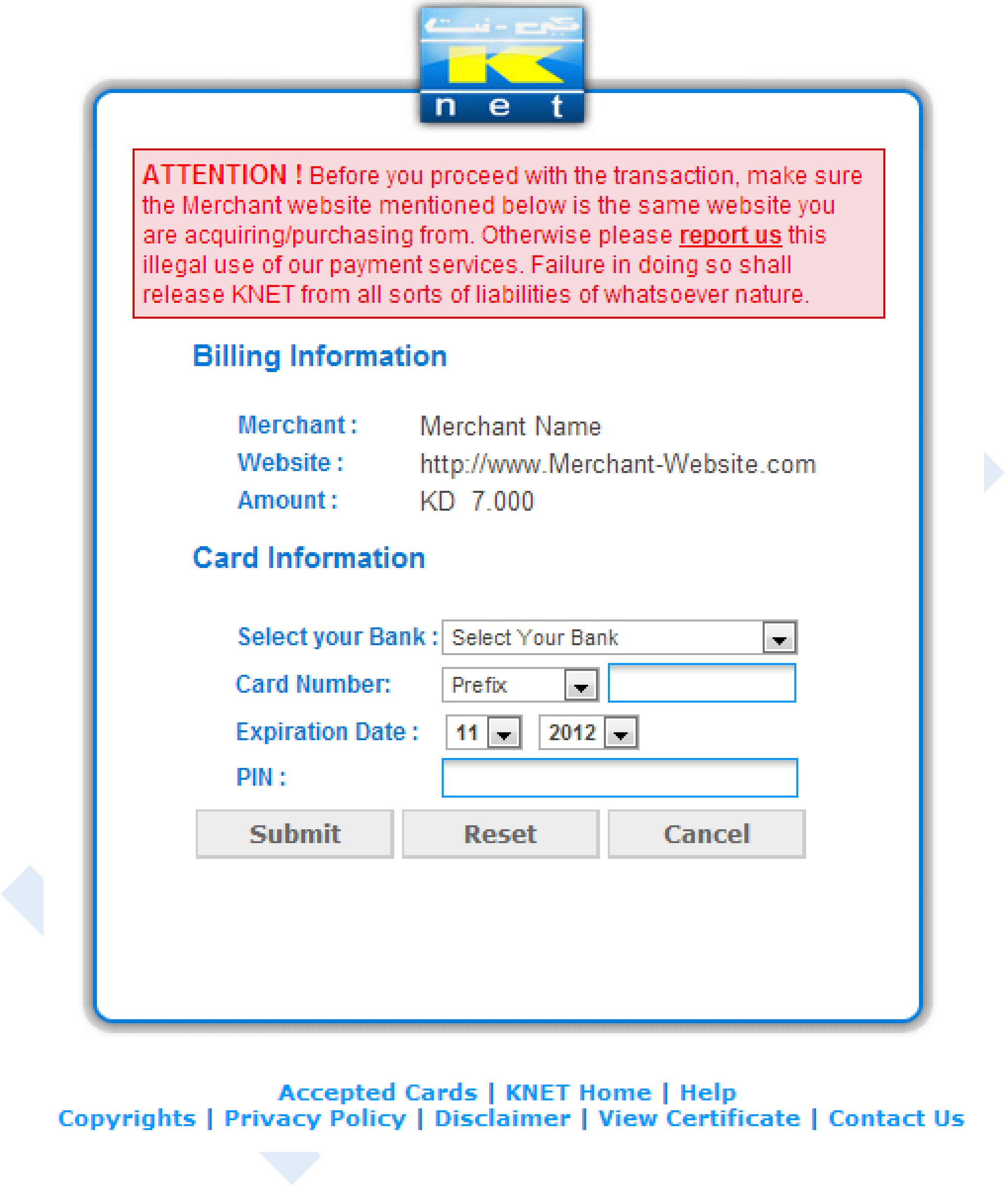
Appendix A: PG Merchant Questionnaire

The merchant must fill this form and return it to KNET prior to testing.

Merchant's Name	
Sponsor Bank	Select Bank from the List
Date	

Do you have an existing site and/or payment provider?	Yes □	No □
What environments do you use/want to use?	Java □ ASP □ asp.Net □ Cold Fusion □ Php □ Other :	
If you are a JAVA shop, then which Application Server do you use? ***Details of Hardware platform and O/S as well.***		
Which Web Server do you use? What Hardware does it run on?		
➢ Do you run a database?	Yes □	No □
➢ Which Database you are using?		
➢ Version of You Database?		
➢ Platform and O/S?		
Are you using SSL certificate for your website?	Yes □	No □
➢ Name of the Provider?	www.Verisign.com	□
	www.Thawte.com	□
	www.geotrusteurope.com	□
	www.Equifax.com	□
	None of above	□
Is the "shop" home built or using a package solution?	Home built □	Package solution □
➢ If it's "Package", which?		
How KNET integration will be performed?	In-house □	Outsource □
➢ If "**Outsource**" please provide the required information.	Company Name	
	Contact Name	
	Telephone number	
	E-mail Address	

Appendix B: KNET PG screens shots



ATTENTION ! Before you proceed with the transaction, make sure the Merchant website mentioned below is the same website you are acquiring/purchasing from. Otherwise please report us this illegal use of our payment services. Failure in doing so shall release KNET from all sorts of liabilities of whatsoever nature.

Billing Information

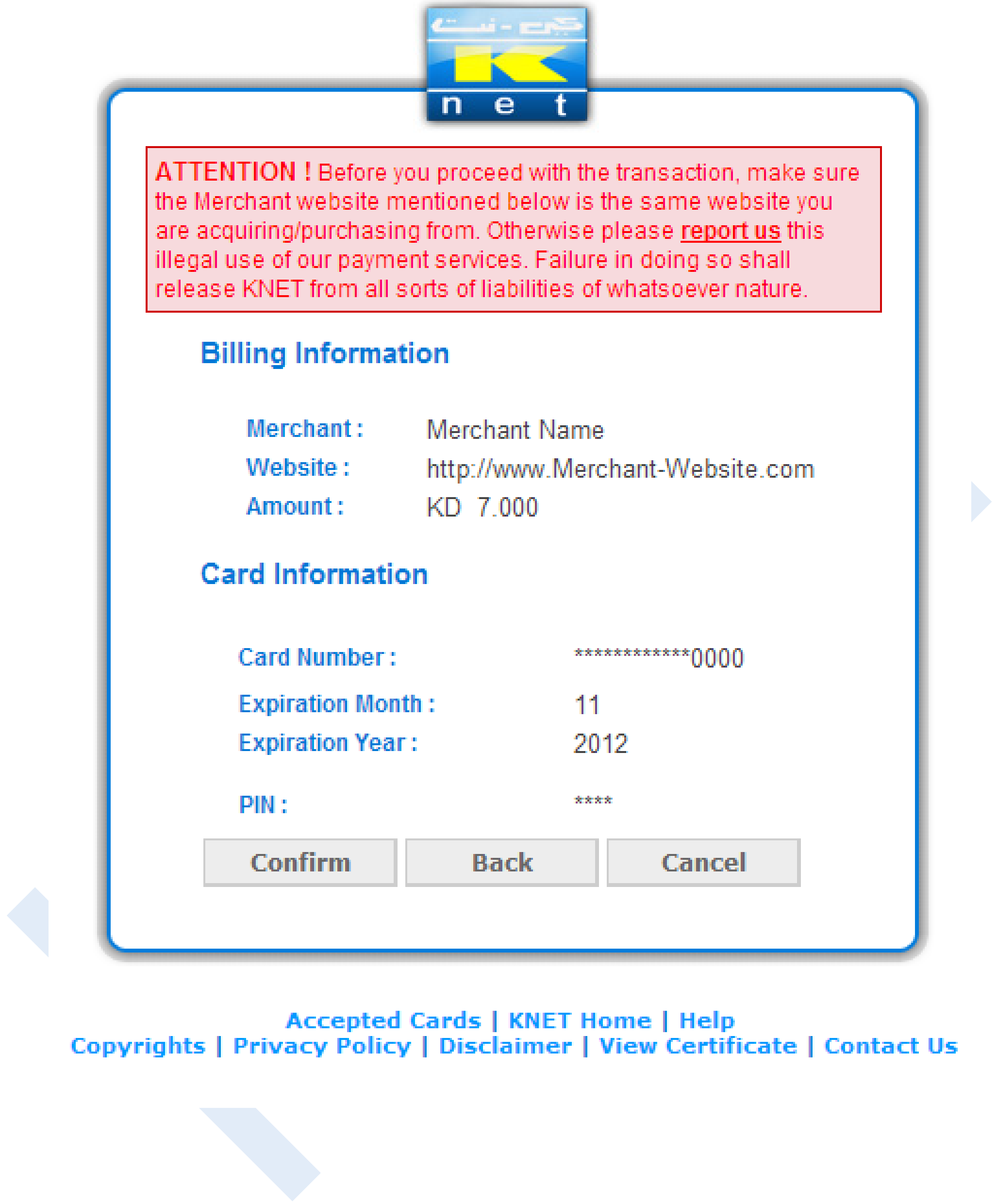
Merchant :	Merchant Name
Website :	http://www.Merchant-Website.com
Amount :	KD 7.000

Card Information

Select your Bank :	Select Your Bank
Card Number:	Prefix
Expiration Date :	11 2012
PIN :	

Submit
Reset
Cancel
Accepted Cards | KNET Home | Help
Copyrights | Privacy Policy | Disclaimer | View Certificate | Contact Us

Figure 1. Payment Page



ATTENTION ! Before you proceed with the transaction, make sure the Merchant website mentioned below is the same website you are acquiring/purchasing from. Otherwise please report us this illegal use of our payment services. Failure in doing so shall release KNET from all sorts of liabilities of whatsoever nature.

Billing Information

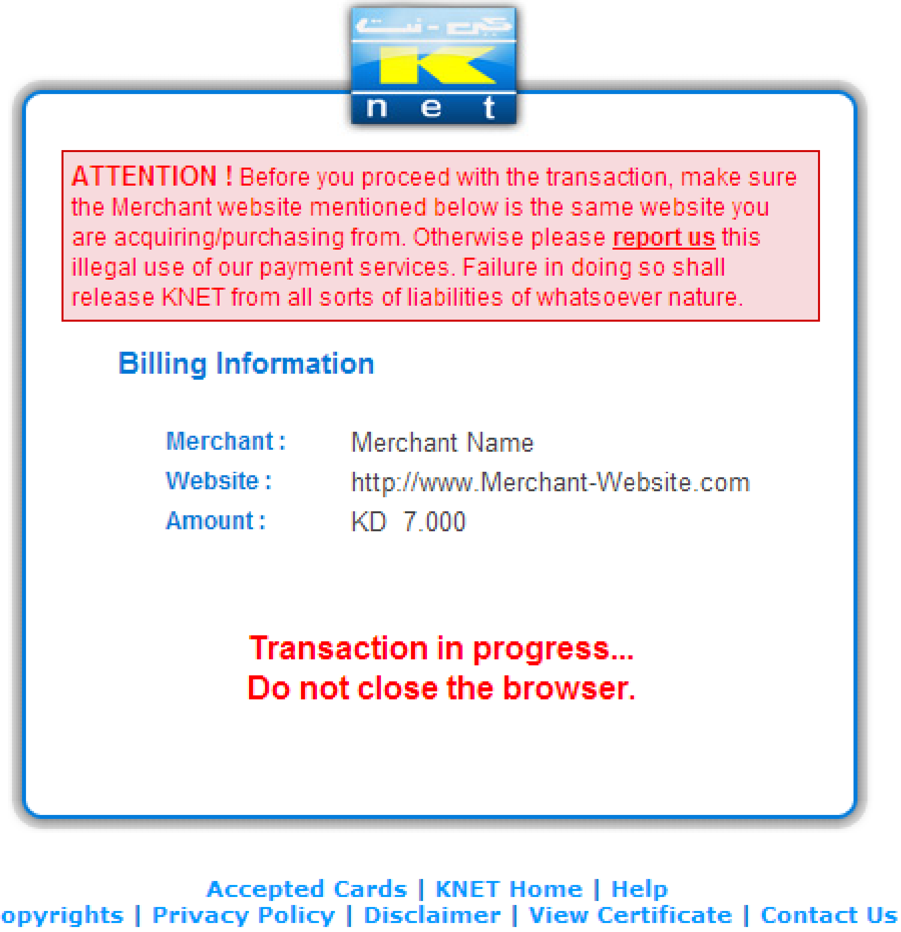
Merchant :	Merchant Name
Website :	http://www.Merchant-Website.com
Amount :	KD 7.000

Card Information

Card Number :	************0000
Expiration Month :	11
Expiration Year :	2012
PIN :	****

Confirm
Back
Cancel
Accepted Cards | KNET Home | Help
Copyrights | Privacy Policy | Disclaimer | View Certificate | Contact Us

Figure 2. Confirmation Page



ATTENTION ! Before you proceed with the transaction, make sure the Merchant website mentioned below is the same website you are acquiring/purchasing from. Otherwise please report us this illegal use of our payment services. Failure in doing so shall release KNET from all sorts of liabilities of whatsoever nature.

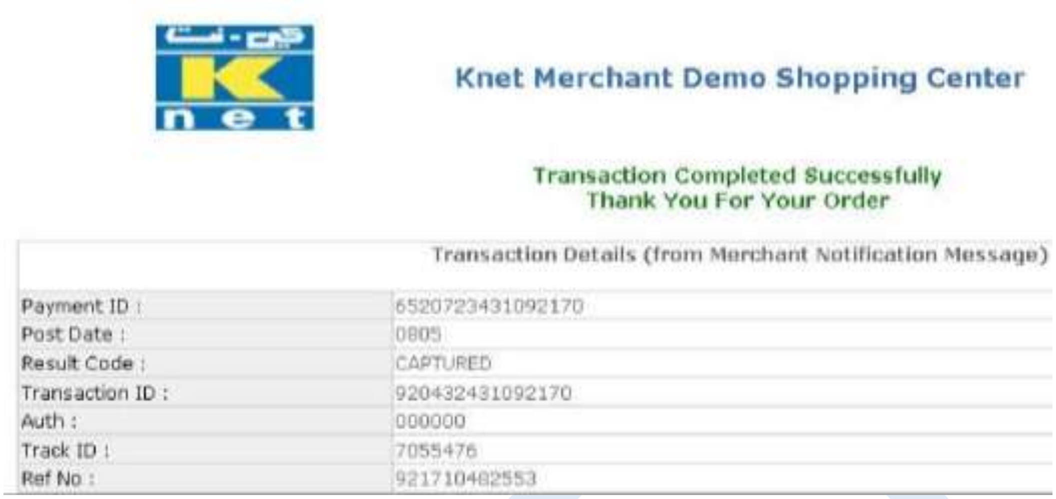
Billing Information

Merchant :	Merchant Name
Website :	http://www.Merchant-Website.com
Amount :	KD 7.000

Transaction in progress...
Do not close the browser.

Accepted Cards | KNET Home | Help
Copyrights | Privacy Policy | Disclaimer | View Certificate | Contact Us

Figure 3. Transaction in Progress



Knet Merchant Demo Shopping Center

Transaction Completed Successfully
Thank You For Your Order

	Transaction Details (from Merchant Notification Message)
Payment ID :	6520723431092170
Post Date :	0805
Result Code :	CAPTURED
Transaction ID :	920432431092170
Auth :	000000
Track ID :	7055476
Ref No :	921710482553

Figure 4. Sample Receipt Page

Appendix C: Merchant Certification Checklist

Before going live with any new merchant, the merchant must go through a checklist that summarizes all the previous steps to confirm that all requirements for that merchant are completed before the final certification test.

Please note that this is a generic certification checklist, which needs to be customized for each merchant.

1. Merchant import KNET CA certificates successfully from the following url:
   a. *For testing environment: https://www.knetpaytest.com.kw*
   b. *For production environment: https://www.knetpay.com.kw*

2. Merchant installs resource file and communicates with KNET properly using SSL.

3. Merchant must provide KNET an SSL V3 certificate.
   a. *KNET only accepts the following SSL certificate issuers:*
      - www.Verisign.com
      - www.Thawte.com
      - www.geotrusteurope.com
      - www.Equifax.com
      - www.digicert.com
      - www.comodo.com

4. Proper initialization of the E24PaymentPipe with the web page equivalent to “buy” demo page.

5. Customer redirected to KNET payment page.

6. Track ID generation by merchant at the page before redirecting to KNET payment page. Please ensure that each transaction has a unique Track ID.

7. Ensure that the “PAY” button on the page right before redirecting to KNET payment page can be clicked ONLY ONCE. This is to ensure that no duplicate transactions are created at the click of the button.

8. Merchant must present a receipt page after a successful or unsuccessful transaction and display at least the following:
   a. *Date/time (Server Date & Time)*
   b. *Indication of successful or unsuccessful transaction.*
   c. *Amount*
   d. *Reference ID*
   e. *Payment ID*
   f. *Merchant track ID*

9. Merchant must send a notification e-mail to the customer after a successful transaction displaying the receipt page with the values mentioned above.

10. Merchant must present an error page after an unsuccessful transaction and display all the values mentioned in point 8 along with a clear notification of unsuccessful transaction.

11. Merchant MUST save all the records for all the transaction such as

- Result Code
- Auth Code
- Track ID
- All the UDFs
- Payment ID
- Transaction ID
- Reference ID
- Amount
- Date/time

Appendix D: Merchant Certification Checklist by Knet

Before going live with any new Merchant, KNET will go through a checklist that summarizes all the previous steps at a high level to confirm that all the requirement for that merchant are completed before moving it to Knet PG production environment. This will serve as a final health check before going live.

Please note that this is a generic go-live checklist, which needs to be customized for each merchant**.**

Certification Checklist by Knet

Merchant:…………….

Date:…………………….

- ➢ Merchant uploaded KNET CA certificates successfully
- ➢ Knet uploaded merchant certificates (CA and server) successfully
- ➢ Merchant installs resource file and communicates with KNET properly using SSL Proper initialization of the E24PaymentPipe with the web page equivalent to "buy" demo page
- ➢ Track ID generation by merchant
- ➢ Payment ID generation by KNET
- ➢ Customer redirected to KNET payment page.
    - o If approved:
        - ▪ Notification details are passed to merchant after a successful transaction and presentation of merchant's receipt for the customer with reference number displayed.
        - ▪ Email notification with transaction details (receipt page) sent by merchant to customer. Receipt page should include:
            1. Transaction reference number
            2. Track ID (issued by Merchant)
            3. Items purchased and price
            4. Total price
            5. Customer support phone number in case the client has questions about the transaction
            6. Other details as required by Merchant

NOTE: Ensure that the email notification matches the receipt page by sending the receipt page to the customer using HTML format.

- If declined:
  - Declined code returned to merchant and customer is redirected to merchant’s error page
  - Verify executed transactions in KNET server and logs
  - Verify “Notify” flag is NOT set when merchant goes offline
  - Transactions amount & total should match between KNET & merchant from the merchant database extract for a given period of time.
  - Track ID generation by merchant at the page before redirecting to KNET payment page. Ensure that each transaction has a unique Track ID.
  - Ensure the “PAY” button can be clicked ONLY ONCE. This is to ensure that no duplicate transactions are created.
  - Not Notified – A “CAPTURED” transaction would be “VOIDED” whenever there is a failure in the notification of details to the merchant

Remarks:……………………………………………………………………………………….

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

Conducted by: …………………………..

Appendix E: Void Transaction Guide

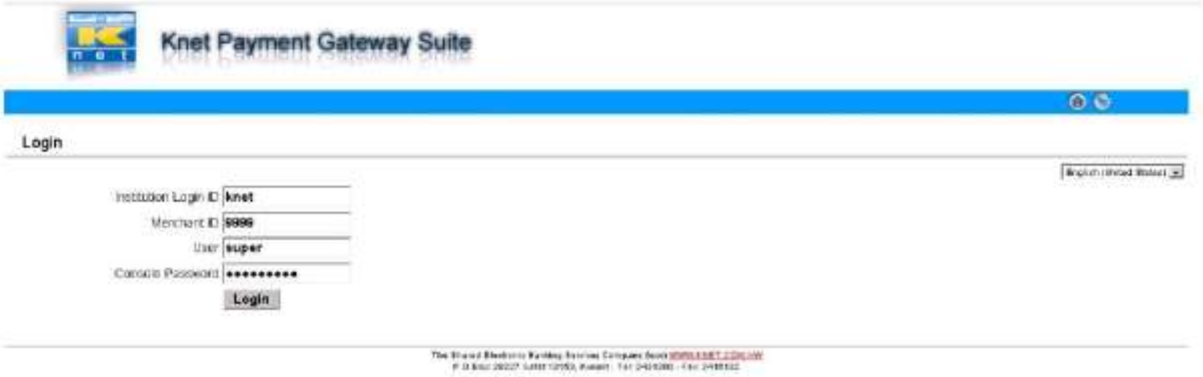
Introduction

The objective of this section is to provide supplementary information to the subject who is authorized to generate void transactions. This will serve as a guide to the end user who wants to make use of the Payment Gateway's Reversal feature in cases whereby a merchant considers a transaction to be a dispute or some such other case.

Procedure

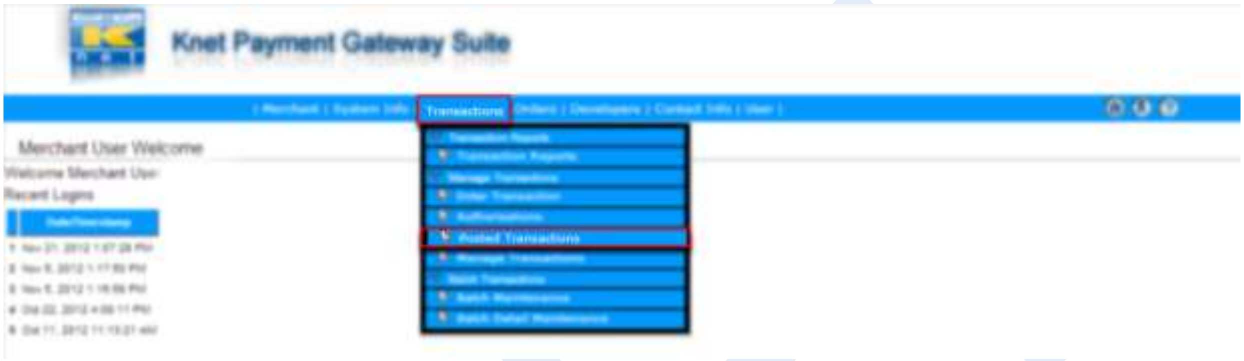
The following screen shots demonstrate the steps to be carried out in voiding a transaction.

- Login to the merchant site via the URL https://www.knetpay.com.kw/CGW/Merchant using the information provided by KNET to the merchant such as the institution id, merchant id, username and password.



Institution Login ID: knet
Merchant ID: 9999
User: super
Console Password: ●●●●●●●●●●

- Navigate to **Transactions** and click **Posted Transactions** to get the Transaction Management Filter screen as shown below. Fill out the criteria fields or leave it as is. By default, the current date is set as both start and end dates. When done, click **Search** button to display the report.

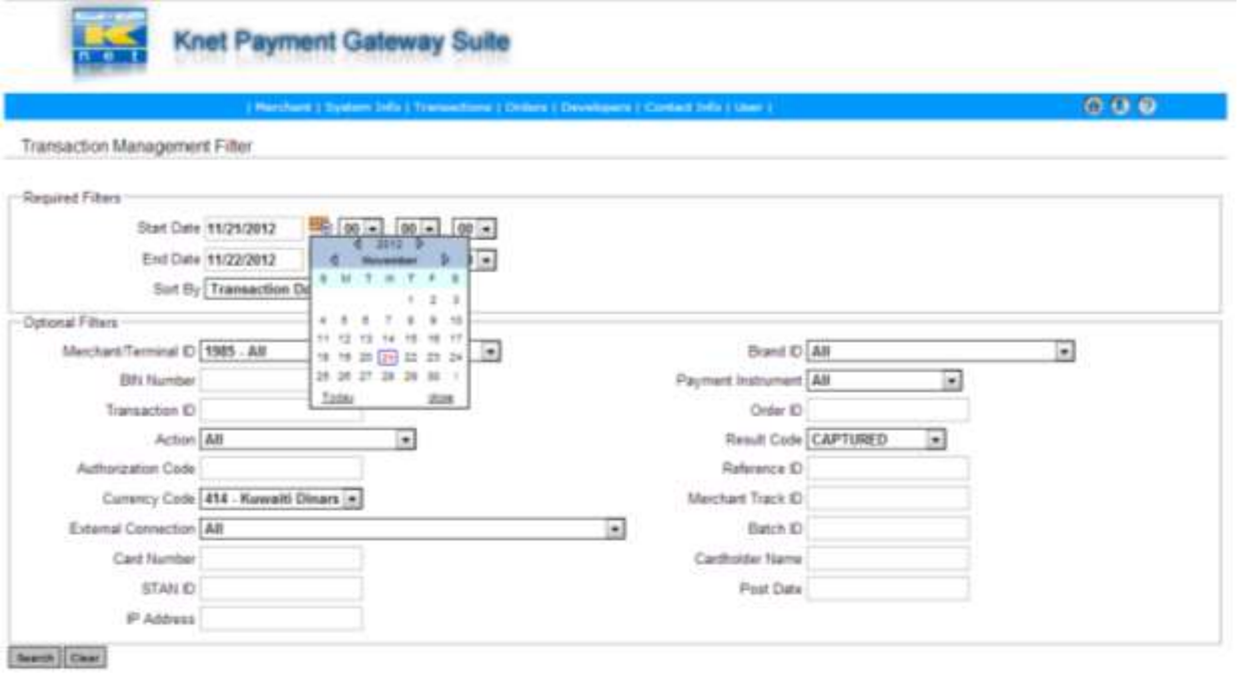


Merchant User Welcome

Welcome Merchant User

Recent Logins

1 Nov 21, 2012 1:07:28 PM
2 Nov 8, 2012 1:17:52 PM
3 Nov 8, 2012 1:16:56 PM
4 Oct 22, 2012 4:08:11 PM
5 Oct 11, 2012 11:13:21 AM


Knet Payment Gateway Suite
| Merchant | System Info | Transactions | Orders | Developers | Contact Info | User |
2012
November
S M T W T F S
Today
close
Search
Clear

Transaction Management Filter

Required Filters

Start Date 11/21/2012 00 00 00

End Date 11/22/2012

Sort By Transaction Da

Optional Filters

Merchant/Terminal ID 1985 - All

BIN Number

Transaction ID

Action All

Authorization Code

Currency Code 414 - Kuwaiti Dinars

External Connection All

Card Number

STAN ID

IP Address

Brand ID All

Payment Instrument All

Order ID

Result Code CAPTURED

Reference ID

Merchant Track ID

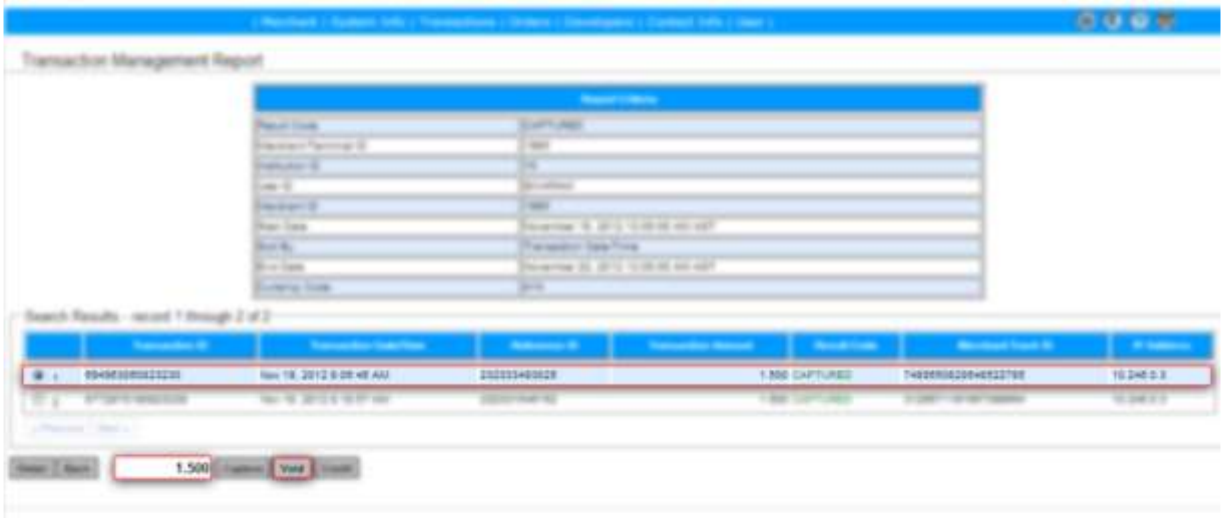
Batch ID

Cardholder Name

Post Data

- The screen shot below is the Transaction Management Report generated by the previous step.

  From here, all transactions that meet the criteria from the Transaction Management Filter are displayed. Select the transaction to void. By selecting the transaction, the amount of this transaction will appear in the text box (as shown below). Clicking on the **Void** button will void this selected transaction.



| Merchant | System Info | Transactions | Orders | Developers | Contact Info | User |

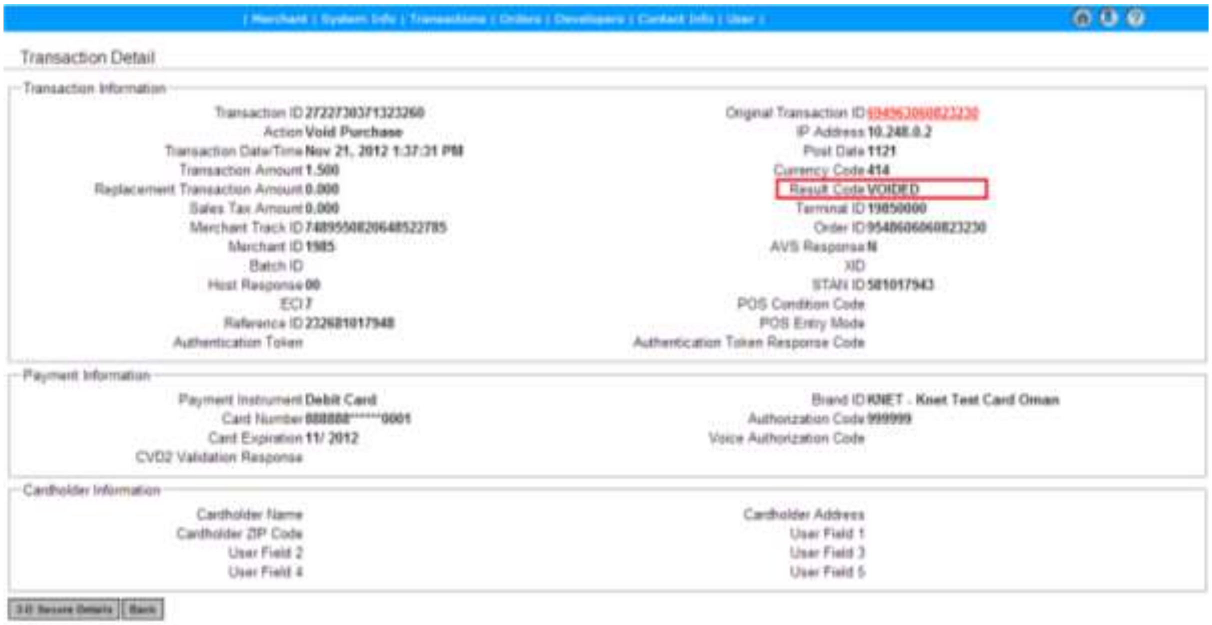
Transaction Management Report

Report Criteria	
Result Code	CAPTURED
Merchant Terminal ID	[illegible]
Institution ID	[illegible]
User ID	[illegible]
Merchant ID	[illegible]
Start Date	November 18, 2012 12:00:00 AM AST
Sort By	Transaction Date/Time
End Date	November 22, 2012 12:00:00 AM AST
Currency Code	[illegible]

Search Results - record 1 through 2 of 2

	Transaction ID	Transaction Date/Time	Reference ID	Transaction Amount	Result Code	Merchant Track ID	IP Address
● 1	[illegible]	Nov 18, 2012 [illegible] AM	[illegible]	1.500	CAPTURED	[illegible]	[illegible]
○ 2	[illegible]	Nov 18, 2012 [illegible] AM	[illegible]	1.500	CAPTURED	[illegible]	[illegible]

- After voiding the transaction, details are displayed with reference to the transaction that was voided.


| Merchant | System Info | Transactions | Orders | Developers | Contact Info | User |

Transaction Detail

Transaction Information

Transaction ID	2722730371323260	Original Transaction ID	694963060823230
Action	Void Purchase	IP Address	10.248.0.2
Transaction Date/Time	Nov 21, 2012 1:37:31 PM	Post Date	1121
Transaction Amount	1.500	Currency Code	414
Replacement Transaction Amount	0.000	Result Code	VOIDED
Sales Tax Amount	0.000	Terminal ID	19850000
Merchant Track ID	7489550820648522785	Order ID	954860606060823230
Merchant ID	1985	AVS Response	N
Batch ID		XID	
Host Response	00	STAN ID	581017943
ECI	7	POS Condition Code	
Reference ID	232681017948	POS Entry Mode	
Authentication Token		Authentication Token Response Code	

Payment Information

Payment Instrument	Debit Card	Brand ID	KNET . Knet Test Card Oman
Card Number	888888******0001	Authorization Code	999999
Card Expiration	11/ 2012	Voice Authorization Code	
CVD2 Validation Response			

Cardholder Information

Cardholder Name		Cardholder Address	
Cardholder ZIP Code		User Field 1	
User Field 2		User Field 3	
User Field 4		User Field 5	

Appendix F: Payment Gateway Troubleshooting Reference

F-1 Initialization problem

Description:
An error occurs when the PerformInitTransaction() (or performPaymentInitialization()*in jsp*) is called, or it returns a non-zero value indicating a problem.

Resolution:

Ensure that:

1. The Alias is set correctly
2. The resource file exists in the referenced path
3. The resource file path ends with a backslash '\' e.g. (D:\Resource\)
4. The *e24PaymentPipeLib* DLL is registered using the command *C:\Windows\system32\regsvr32.exe.* If a 64-bit Windows Server 2008 server is used, the DLL should be placed in the *c:\windows\sysWOW64\* directory before registration.
5. The resource path/directory grants full control (Read, Write, Execute) for the web server process account (NETWORK SERVICE or ASPNET, …)
6. The *e24PaymentPipeLib.dll* path/directory grants full control (Read, Write, Execute) for the web server process account (NETWORK SERVICE or ASPNET, …)
7. In case a 64-bit Windows Server is used, the IIS application pool which runs the merchants website should be configured to allow 32-bit applications.
8. Ensure the reachability from client server to www.knetpaytest.com.kw through port **80** and port **443 (telnet to these ports)**
9. UDFs & Track ID do not contain hack characters. Following is a list of the hack characters:

~	x7E	TILDE
`	x60	LEFT SINGLE QUOTATION MARK, GRAVE ACCENT
!	x21	EXCLAMATION POINT (bang)
#	x23	NUMBER SIGN (pound sign)
$	x24	DOLLAR SIGN
%	x25	PERCENT SIGN
^	x5E	CIRCUMFLEX ACCENT
|	x7C	VERTICAL LINE (pipe)
\	x5C	REVERSE SLANT (REVERSE SOLIDUS)(backslash, backslash)
:	x3A	COLON
'	x27	APOSTROPHE, RIGHT SINGLE QUOTATION MARK, ACUTE ACCENT (single quote)
"	x22	QUOTATION MARK, DIAERESIS
/	x2F	SLANT (SOLIDUS), (slash)

F-2 - Auto-void problem

Description:

After entering the card number and the PIN in KNET payment page and confirming the payment, this error message is received from KNET.: *"An error has occurred. Your order has been cancelled and amount has been refunded back into your account. Please try again later"*

Causes:

- KNET Payment Gateway is unable to reach the merchant's response page
- The response page has some execution errors
- The response page output is not in the right format
- SSL Certificate problem

Resolution:

Ensure that:

1. SSL certificate is installed and valid and KNET is notified after installing the SSL certificate
2. The response page (or notification servlet) URL passed in the buy page is valid and accessible from outside the web server
3. The response page (or notification servlet) does not cause any server error and does not throw any exception. All exceptions should be handled.
4. The response page does not use Response.Redirect function. Instead, Response.Write (or echo or out.println) should be used.
5. The response page (or notification servlet) output is only the text line REDIRECT=https://…?... KNET PG will read the output line, extract the URL part and redirect the client's browser to that URL.

6. The response page (or notification servlet) does not output any markup (HTML tags). This implies that it should have no MasterPage/Template and is excluded out of the security system (e.g. ASP.NET membership)

Verify this by navigating on the browser to the URL of the response page (or notification servlet). View the page source and the same text line re-appears: **REDIRECT=https://…?...**

F-3 - Payment Gateway suddenly stopped working

Description:

The Payment Gateway was working well and suddenly started to give an error (Auto-void/customer redirected to error page)

Resolution:

Ensure that:

1. The SSL certificate has not expired

2. If SSL certificate or the website domain name was changed, KNET PG support team should have been informed few days prior to this activity. In case of the domain name changes, KNET PG Support team will not be activated until this information is sent through the bank.

3. If the website’s source code has been modified, Re-check point 4 & 5.

8. Document Update Notice

Revision No : 1

Reason for Change: Migration from CGW 3.0.2 to CGW 3.1.2

Section	**Existing Page(s) / Date**	**New Page(s) / Date**	**Action**
i	i/20 Jul. ‘06	i / 21 Oct. ‘09	Replace
3	3-1 / 20 Jul. ‘06	3-1 / 21 Oct. ‘09	Replace
4	4-1 – 4-2 / 20 Jul. ‘06	4-1 – 4-2 / 21 Oct. ‘09	Replace
5	5-4 – 5-9 / 20 Jul. ‘06	5-4 – 5-9 / 21 Oct. ‘09	Replace
7	-	7-1 / 21 Oct. ‘09	Insert
8	-	8-1 / 21 Oct. ‘09	Insert
A	A1 / 20 Jul. ‘06	A1 / 21 Oct. ‘09	Replace
B	B-1 – B-4/ 20 Jul. ‘06	B-1 – B-4/21 Oct. ‘06	Replace
C	C1 / 20 Jul. ‘06	C1 / 21 Oct. ‘09	Replace
E	E-2 – E-5/ 20 Jul. ‘06	E-2 – E-5/21 Oct. ‘06	Replace

Revision No : 2

Reason for Change: Addition of missing points

Section	Existing Page(s) / Date	New Page(s) / Date	Action
I	i / 21 Oct. '09	Table of Contents / 24 May. '11	Updated
3	3-1 / 21 Oct. '09	2-3 /24 May. '11	Updated
4	4-1 – 4-2 / 21 Oct. '09	4-6 /24 May. '11	Updated
5	5-4 – 5-9 / 21 Oct. '09	6-12 / 24 May. '11	Updated
6	6-1 / 21 Oct. '09	17 / 24 May. '11	Updated
7	7-1 / 21 Oct. '09	18 / 24 May. '11	-
8	8-1 / 21 Oct. '09	19 / 24 May. '11	Updated
A	A1 / 21 Oct. '09	20 / 24 May. '11	Updated
B	B-1 – B-4/21 Oct. '06	21 / 24 May. '11	-
C	C1 / 21 Oct. '09	25 /24 May. '11	-
E	E-2 – E-5/21 Oct. '06	29-32 /24 May. '11	Updated
F	-	33-35 /24 May. '11	Addition

Revision No : 3

Reason for Change: Re-issue (Version 2.0 dated 15 May 2017)