Payment Gateway Merchant Integration

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

Version: 1.3   **Doc:** K-035   **Date:** 21 Nov, 2012

Table of Contents

1. Introduction

KNETPAY is Knet's 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 on the future.

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

2. Service Overview

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
Secured, Knet-hosted payment page
Online authorization of payment transactions
Merchant-gateway plug-in integration

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

3. Merchant Pre-requisites

Once a Merchant Nomination is received from one of our member Banks, Knet will require the Merchant Questionnaire form (See Appendix A) to be completed by the merchant and re-turned to Knet.

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

3.2 Plug-ins

Knet can support the following types of software plug-ins:
- Cold Fusion
- ASP
- .Net
- Java

Based on the Merchant application server Knet will provide the appropriate plug-in needed to be used on the merchant site.

3.3 Network and SSL

The merchant web server should have a valid SSL certificate V3. Knet server will receive transactions only from SSL certified web servers. Knet is currently running with Verisign 3.0 SSL certificate. It is the merchant's responsibility to contact an authorized certificate authority and purchase its own certificate.

Knet only accept the following SSL certificate issuers:

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

Knet is not going to provide an SSL certificate for the merchant.

Knet Payment Gateway server must import the Merchant's SSL certificate using the keytool utility from JDK.

If merchant's CA certificate is not included by default in the keystores packaged, the following procedure will be followed:

For JSP:

Knet CA certificate should be imported to a CA trust store using the keytool utility by Java or any other utility provided by the merchant's application server. (i.e. keyman from WebSphere)

The syntax of the keytool utility is as follows:

keytool –import –keystore <path_to_JDK_installation>/jre/lib/security/cacerts -alias

<some_nickname_for_the_site> -file <path_to_certificate_file>

All this is to be run on a single command line and will prompt for the key store password.

For ASP & .Net:

Import Knet CA certificate through Windows' Certificate Import Wizard. Make sure to place it in a trusted root CA store for the local computer account.

3.4 Protocol : https

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

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:

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

   - ASP
   - .NET
   - Java
   - ColdFusion

   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.

2. 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:
     1. Binaries (DLLs that will be called by your 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 you need to reference in your ASP.NET project in order to utilize e24PaymentPipe.dll

     2. Sample Application.
        a. Source Code - A demo ASP.NET/C# Web

Application written with Visual Studio 2008
and utilizing e34PaymentPipe.dll

b. Published Project (A compiled version of the
demo project ready to deploy in IIS)

Application Deployment Procedure:

1. Make sure that .NET Framework 2.0 or higher is
installed.

2. Make sure that the 'Bin' directory contains the
following files:
   a. e24PaymentPipe.dll
   b. Interop. e24PaymentPipe.dll
   c. The application's DLL

3. Register the 'e24PaymentPipe' DLL using the
command: C:\Windows\system32\regsrv32.exe
e24PaymentPipe.dll

4. Copy the files under the 'Published' folder to
'C:\Inetpub\wwwroot\<YourAppName>

5. Run 'IIS Management Console' and expand 'Default
Web Site' node under 'Local Computer > Web
Sites'

6. Right-click the < YourAppName> node and choose
'Properties'
   a. Click 'Create' button to create an
   application
   b. In the ASP.NET tab, make sure that
   'ASP.NET 2.0' is selected
   c. In the 'Directory Security' tab, Click Edit,
   and check 'Allow Anonymous Access'
7. 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
   user you've added

- **Java:** This plug-in is in the form of a class file that 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 2 SDK/JRE 1.4 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.

- **ColdFusion:** This plug-in is in the form of a DLL file that needs to be registered on the merchant server as in the case of the ASP plug-in.

3. For security purposes, Knet requires merchants to make use of the resource file to secure Payment Gateway communications with the merchant's plug-in. The resource file is specific to each merchant and contains encrypted data that identifies the merchant as a valid user of the Payment Gateway. Knet will generate the resource file and send it to the merchant including the alias to be used to access it. Upon receipt, the merchant will save the file in any directory.

4.2   Merchant Front End Integration:

The plug-in needs to be integrated the merchant front end to perform authorizations. 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 pay. 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 Transaction Capturing System requires the developer to communicate a handful of variables across the Internet via http post to Knet address.

All posted data must be in the form of an URL Encoded string of name value pairs.

Title:	Payment Gateway Merchant Integration	**Date:**	21 Nov. 12
Doc :	K-035	**Version:**	1.3

Page 7
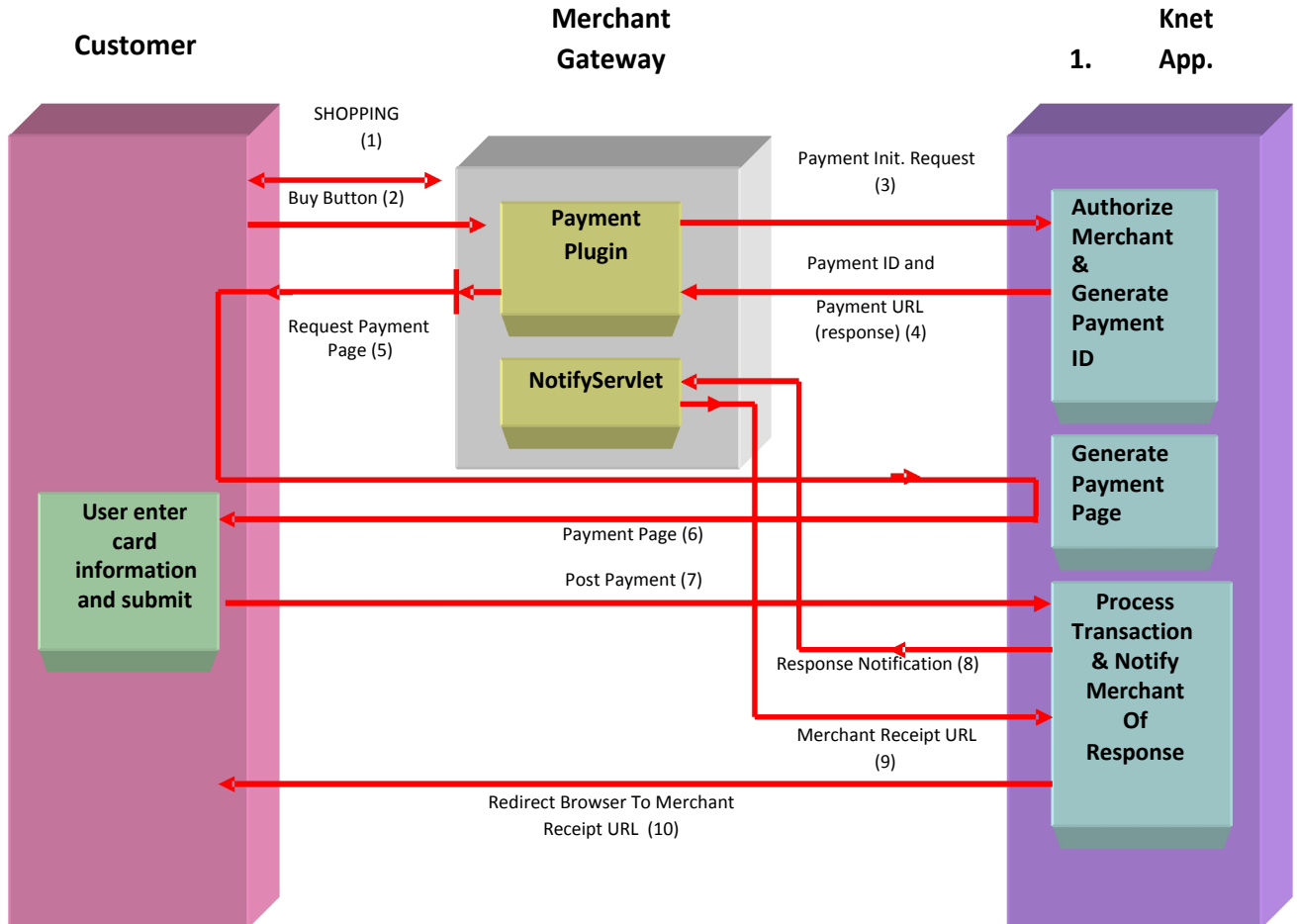
5. Transaction processing

5.1 Message flow

Figure 1. Processing Flow

1) Customer is shopping on the merchant side.
2) Customer finish shopping, proceed 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 save payment initialization parameters with payment ID being the primary key, and redirect the customer to the Knet payment page.
6) Knet builds the payment page and sends it to the customer.
7) User enters card information and clicks submit.
8) Knet process the transaction. If no error occurs, Knet send transaction response to Merchant Response URL specified in payment initialization.
9) Merchant process and save the response, send back Knet the receipt URL.
10) Knet redirects customer to Merchant Receipt URL.

5.2   Payment Initialization

When the customer checks out from 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 his future reference (he will need it 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 his 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 - *Resource Path* is the absolute path to the folder which contains your resource file. Resource file will be provided to you with toolkit by Knet.

Alias - Alias name is provided by Knet along with the toolkit. It is like userID to for the resource file.

action - Must always be in numeric format. ● Value – 1

(Purchase)

amt - The Transaction amount. The number includes the decimal point (always 3 for Kuwait Dinars), e.g 15.750, 10.000,…., etc.

currencycode - The currency code of the transaction. For KD, use currency code 414.

langid - 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.

transid - Transaction ID, used to identify the original transaction for a void and capture transaction.

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

errorURL - 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, use only alpha-numeric). Such ID should be unique for every transaction the merchant initiate irrespective of the customer. Merchant should store trackid on his local database for his records.

udf1- udf4 User Defined Fields 1,2,3,4 can be used to pass and store any additional transaction data you would like archived with the transaction and available as a searching criteria.

udf5- User Defined Field 5, can be used to pass and store any additional transaction data you would like archived with the transaction and available as a searching criteria.
NOTE: This field has been dedicated for merchants who would like to pass additional customer data and wish to include it in acquiring bank's report. Refer to section 5.2.2 for more details

When the customer hits the "Buy" button, control will be transferred 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 the Knet system to the website URL specified in the Response URL hidden field.

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/JSP pages. ASP/JSP pages must also be provided on the Merchant's website to receive, interpret and display the response from the Knet system.

- Knet will provide an **ASP** 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:

  o *Change all URLs in the attached files to reflect your website address*

  o *Change your alias in buy.asp to "____"* *(MyObj.Alias = "____") Knet will send each merchant a unique alias*

  o *Copy the attached resource.cgn into a specific directory and point at in inside your buy.asp (MyObj.ResourcePath = "D:\\resource\\") Knet will send each merchant a unique resource*

  o *Register the e24paymentpipe.dll on your system as follows: CMD> regsvr32 e24paymentpipe.dll*

  o *Provide a copy of your website's SSL certificate so we can add it to our system*

- Knet will provide an **JSP** 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:

  o Change all URLs in the attached files to reflect your website address

  o Change your alias in config.txt to "_____" (alias = "____") *Knet will send each merchant a unique alias*

  o Copy the attached resource.cgn into a specific directory and point at in inside your config.txt (resourcePath=D:\\Resource\\) *Knet will send each merchant a unique resource*

  o Recompile NotifyServlet.java after modifying its content

  o *Provide a copy of your website's SSL certificate so we can add it to our 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 initialization and customer redirection to Knet payment page has succeeded, the customer is asked in the Knet payment page to enter card details. After the customer submits his 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*. Merchant **MUST** save the returned values by Knet into his 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&t
  rackid=632186*

- **NOT CAPTURED:** Transaction is not approved by the bank

paymentid=8962225231041180&result=NOT+CAPTURED&auth=9
99554&ref=411857974849&tranid=9929892231041180&postdate=
0427&trackid=632186

- **HOST TIMEOUT:** Bank doesn't respond within the timeout period
  paymentid=8962225231041180&result=HOST+TIMEOUT&ref=411
  857974849&tranid=9929892231041180&trackid=632186

- **DENIDED 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&u
  df4=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 you have checked for an error.

- 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) you would like 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 Merchants' integration to 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 questioner, they can start testing with Knet test environment. In case of any problem, their assigned Knet technical staff 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

  o At least 5 CAPTURED
  o At least 5 NOT CAPTURED
  o At least 5 NOT CAPTURED and VOIDED in a "NOT NOTIFIED" case simulation

7. Go-live

A merchant will go live once it has 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.

Title:	Payment Gateway Merchant Integration	Date:	21 Nov. 12
Doc :	K-035	Version:	1.3

Page 18

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

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	04/02/2010

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

Figure 1. Payment Page

Figure 2. Confirmation Page
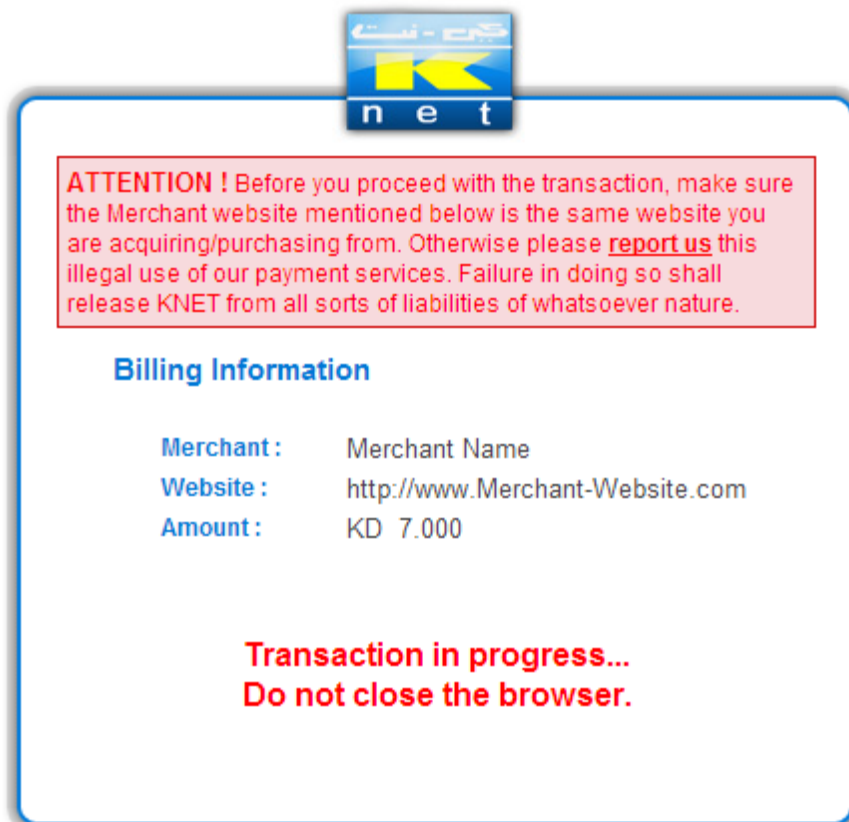
Figure 3. Transaction in Progress

Figure 4. Sample Receipt Page

Appendix C: Merchant Certification Checklist

Before going live with any new Merchant, 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:
    1. *For testing environment: https://www.knetpaytest.com.kw*
    2. *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.
    1. Knet only accept the following SSL certificate issuers:
        - www.Verisign.com
        - www.Thawte.com
        - www.geotrusteurope.com
        - www.Equifax.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's payment page. Please ensure that each transaction has a unique Track ID.

7. Make sure the "PAY" button on the page right before redirecting to Knet's 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:
    1. *Date/time (Server Date & Time)*
    2. *Indication of successful or unsuccessful transaction.*
    3. *Amount*
    4. *Reference ID*
    5. *Payment ID*
    6. *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 the 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 <u>resource file</u> 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. Knet

logs indicate PRESENTED for customer.

    o   If approved:

Notification details are passed to merchant after a successful transaction and presentation of merchant's receipt for the customer with <u>reference</u> number displayed.

Email notification with transaction details (receipt page) sent by merchant to customer. Receipt page should include:

1. Transaction <u>reference</u> number
2. <u>Track ID </u>(issued by Merchant)
3. Items purchased and price
4. Total price
5. Customer support's phone number in case the client has questions about the transaction.
6. Other details as required by Merchant

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

    o   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 match between Knet & merchant from a merchant database extract for a given period of time.

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

Make sure the "PAY" button on the page right before redirecting to Knet's payment

page can be clicked ONLY ONCE. This is to ensure that no duplicate transactions are created at the click of the button.

Not Notified – A "CAPTURED" transaction would be "VOIDED" whenever there's a failure in the notification of details to the merchant.

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

………

………

………

Conducted by: …………………………..

Appendix E: Void Transaction Guide

Introduction

The objective of this document is to provide supplementary information to the subject who's 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 where in a merchant considers a transaction to be a dispute or some other cases.

Procedure

The following screen shots exemplify the steps need to be done in voiding a transaction.

- Login to the merchant site via this URL
  https://www.knetpay.com.kw/CGW/Merchant using the information passed by Knet to the merchant such as the institution id, merchant id, username and 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 out as is. By default, the current date is set as both start and end dates. When done, click Search button to display the report.

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

From here, all transactions that met the criteria from the Transaction Management Filter are displayed. Select a transaction you want to void. By selecting the transaction the amount of this transaction will appear in the below text box. Clicking on void button will void this selected transaction.

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

Appendix F: Payment Gateway Troubleshooting Reference

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:

Make sure 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:\Windiws\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. Your server is able to reach www.knetpaytest.com.kw through port **80** and port **443 (telnet to these ports)**
9. UDFs & Track ID do not contain hack characters. Here's 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)

2. Auto-void problem

Description:
After entering the card No. and the PIN in KNET's payment page and confirming the payment, you get the KNET's error message: *"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 your response page

- The response page has some execution errors

- The response page output is not in the right format

- SSL Certificate problem

Resolution:
Make sure 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)

7. Verify this by navigating with your browser to the URL of the response page (or notification servlet). View the page source and again you should see the same text line: **REDIRECT=https://…?...**

3. Payment Gateway suddenly stopped working

Description:
The payment gateway was working fine and suddenly started to give an error (Auto-void or take me to my error page)

Resolution:
Make sure that:

1. The SSL certificate has not expired

2. If SSL certificate or the website domain name was changed, KNET PG support team should be 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 send through the bank.

3. If the website's source code has been modified, Re-check sections 1 & 2

Title:	Payment Gateway Merchant Integration	**Date:**	21 Nov. 12
Doc :	K-035	**Version:**	1.3

Page 35