MPU NationalPayment Gateway Merchant Integration Guide

Version 1.4

Document Control			
Reference			
File Name	MPU NationalPayment Gateway Merchant Integration Guide V1.4.docx		
Author	2C2P (Myanmar) Co., Ltd.		
Revision History			
Version	Date	Revised By	Comments/Reasons
1.0	2Dec 2013	Lynn Htaik Aung	First Version
1.1	11 Feb 2014	Lynn Htaik Aung	Add Inquiry/Void
1.2	24 Apr 2014	Lynn Htaik Aung	Add Logo
1.2	29 Dec 2014	Lynn Htaik Aung	Change Urls
1.4	19 Jan 2015	PaingKyawZin	Change void/Inquiry request(InvoiceNo)

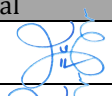
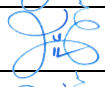
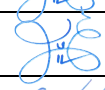
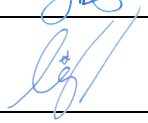
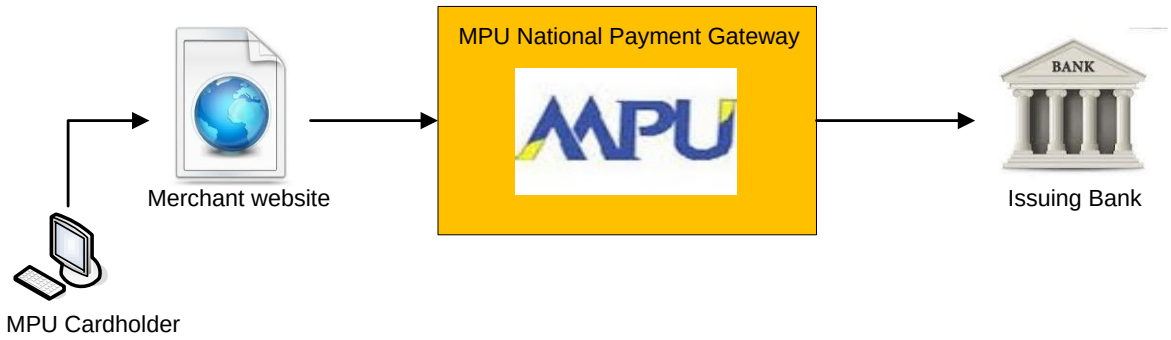
Document Approvals				
Reference				
File Name	MPU NationalPayment Gateway Merchant Integration Guide V1.4.docx			
Author	2C2P (Myanmar) Co., Ltd.			
Revision History				
Version	Approved By	Project Role	Approval Date	Signature/Electronic Approval
1.0	Thandar Aung	Project Manager	3 Dec 2013	
1.1	Thandar Aung	Project Manager	14 Feb 2014	
1.2	Thandar Aung	Project Manager	25 Apr 2014	
1.3	Thandar Aung	Project Manager	5 Jan 2015	
1.4	Lynn Htaik Aung	Senior Solution Architect	21 Jan 2015	

Table of Contents

1. Message Flow (Cardholder)



1. Cardholder visits merchant Web site and clicks check out button.
2. Merchant assembles transaction data and sends to MPU NationalPayment Gateway's payment screen.
3. MPUPayment Gateway checks authentication status of the cardholder.
4. If cardholder's authentication status is valid, OTP (one time password) is sent via SMS or email to the cardholder.
5. Upon successful authentication and OTP verification, MPU Payment Gateway sends authorization message to the issuing bank.
6. MPU Payment Gateway interprets response from the issuing bank and send back transaction results to Merchant Website.

2. Payment Request/Response

i. Field description of Payment Request

No	Variable	Data Type	Length	Mandatory	Description	Remark
1	merchantID	Character	15	Y	Merchant ID	Merchant ID generated by MPU's acquiringBank.
2	invoiceNo	Character &Numeric	20	Y	Unique Invoice No	Provided by Merchant. The invoice number needs to be unique to trace the transaction. Please pad '0' to the left in the case of generated invoice number length is shorter than 20.
3	productDesc	Character	50	Y	Product Description	The product descriptionto be provided by the merchant.
4	amount	Numeric	12	Y	Transaction Amount	The amount needs to be padded with '0' from the left and include no decimal point. Example: 1.00 = 000000000100, 1.5 = 000000000150 Currency exponent follows

						standard ISO4217 currency codes.
5	currencyCode	Numeric	3	Y	Standard ISO4217 Currency Codes.	Refer to Appendix D
6	categoryCode	Character &Numeric	20	N	Category Code	Merchant can distinct the transaction by adding category code.
7	userDefined1	Character	150	N	Merchant Defined Information	(Optional) MPU system will response back to merchant whatever information include in request message of this field.
8	userDefined2	Character	150	N	Merchant Defined Information	(Optional) MPU system will response back to merchant whatever information include in request message of this field.
9	userDefined3	Character	150	N	Merchant Defined Information	(Optional) MPU system will response back to merchant whatever information include in request message of this field.
10	hashValue	Character	150	N	Hash value computed by HMACSHA1 with secret key provided by MPU System.	hashValue will be generated by the following methods: 1. All the input parameters, which are filled in by customer, are sorted by ASCII to be Signature String e.g. **000000050000150823344555557 64Invoice00345MerchantID01Product01UserDefined1UserDefined2UserDefined3**. Please see sample code for details. 2. Signature String will be encrypted by HMACSHA1 with secret key provided by MPU System.

ii. Field descriptions of Payment Response

No	Variable	Data Type	Length	Mandatory	Description	Remark
1	merchantID	Character	15	Y	Merchant ID	Merchant ID generated by MPU's acquiring bank. It can be optional. If it have the value in request, it will be responded back to merchant.
2	respCode	Character	2	Y	The code whether the transaction is successful or not.	00 = Approved; 05 = Do Not Honor; etc. Referto Appendix A for all result code.
3	pan	Character	16	Y	Masked Card Number	First 4 and last 4 digits of card number Example:

						954400xxxxxx1111
4	amount	Numeric	12	Y	Transaction Amount	The amount will be padded with '0' from the left and include no decimal point. Example: 1.00 = 000000000100, 1.5 = 000000000150 Currency exponent follows standard ISO4217 currency codes.
5	invoiceNo	Character &Numeric	20	Y	Unique Invoice No	Provided by Merchant. The invoice number needs to be unique to trace the transaction. Please pad '0' to the left in the case of generated invoice number length is shorter than 20.
6	tranRef	Character	28	Y	Transaction Reference for MPU.	Issued by MPU PGWSystem.
7	approvalCode	Character	6	Y	Transaction Approval Code from Card Issuer Host	
8	dateTime	Numeric	14	Y	Process Date Time value with yyyyMMddhhmmss format	
9	status	Character	2	Y	Last status of transaction	AP= Approved; Refer to Appendix B
10	failReason	Character	100	N	Reason of failure	Refer to Appendix B
11	userDefined1	Character	150	N	Merchant Defined Information	(Optional) MPU system will response back to merchant whatever information include in request message of this field.
12	userDefined2	Character	150	N	Merchant Defined Information	(Optional) MPU system will response back to merchant whatever information include in request message of this field.
13	userDefined3	Character	150	N	Merchant Defined Information	(Optional) MPU system will response back to merchant whatever information include in request message of this field.
14	hashValue	Character	150	Y	Hash value computed by HMACSHA1with secret key provided by MPU System.	hashValue will be generated by the following methods: 1. All the input parameters, which are filled in by customer, are sorted by ASCII to be Signature String e.g. **000000050000150823344555555764Invoice00345MerchantID01Product01UserDefined1UserDefine**

						d2UserDefined3. 2. Signature String will be encrypted by HMACSHA1 with secret key provided by MPU System.

3. Preparing and sending payment request message using HTTP form post

E-Commerce merchant needs to prepare the following Payment Request Form Post message. To prepare this one, merchant must identifyURL on **method="post" action=** to theenvironmentthat merchant would like to locate (Test or Production);

Test:http://122.248.120.252:60145/UAT/Payment/Payment/pay
Production:https://www.mpu-ecommerce.com/Payment/Payment/pay

```
<Form method="post" action="http://122.248.120.252:60145/UAT/Payment/Payment/pay">
<input type="text" id="merchantID" name="merchantID" value="400123456712345"/>
<input type="text" id="invoiceNo" name=" invoiceNo" value="1234567890333"/>
<input type="text" id="productDesc" name="productDesc" value="Test Product"/>
<input type="text" id="amount" name="amount" value="000000010000"/>
<input type="text" id="currencyCode" name="currencyCode" value="840"/>
<input type="text" id="userDefined1" name="userDefined1" value="userDefined1"/>
<input type="text" id="userDefined2" name="userDefined2" value="userDefined2"/>
<input type="text" id="userDefined3" name="userDefined3" value="userDefined3"/>
<input type="text" id="hashValue" name="hashValue"
value="94E8E91C29E73B9648011FADBAE19849B520B24B"/>
</Form>
```

After sending the above HTTP Form Post request , the merchant will have the following information regarding to the transaction, at merchant's "POSTURL" page.**POSTURL** will have both : **FrontEndUrl**- for browser level and **BackEndUrl**- for sever to server. These urls will be configured in the merchant setting by MPU Admin portal.

```
merchantID=400123456712345&
respCode=00&pan=954433XXXXXX1111&amount=000000001000&invoiceNo=1234567890333
&tranRef=123&approvalCode=234567& dateTime= 20130430111211& status=AP&
userDefined1=userDefined1& userDefined2=userDefined2& userDefined3=userDefined3
&hashValue=34E8E91C29E73B9648011FADBAE19849B520B24A
```

Merchant can process the result information by using the following form request methods in various programming languages.

In ASP.Net in C#

```
string merchantID = "",respCode = "",pan = "",amount = "",invoiceNo = "",tranRef =
"",approvalCode = "", dateTime = "",status = "";

merchantID = Request.Form["merchantID"];
respCode = Request.Form["respCode"];
pan = Request.Form["pan"];
```

In Java J2EE

```
String merchantID = "",respCode = "",pan = "",amount = "",invoiceNo = "",tranRef =
"",approvalCode = "", eci = "",dateTime = "",status = "";

merchantID = request.getParameter("merchantID");
respCode = request.getParameter("respCode");
pan = request.getParameter("pan");
```

In PHP

```
$merchantID = "", $respCode = "", $pan = "", $amount = "", $invoiceNo = "",
$tranRef = "", $ approvalCode = "", $eci = "", $dateTime = "", $status = "";
$merchantID = _REQUEST["merchantID"];
$respCode = _REQUEST["respCode"];
$pan = _REQUEST["pan""];
```

The following is a sample code snippet of HMACSHA1in **.Net** programming language.

```
private string getHMAC(string signatureString, string secretKey)
    {
        System.Text.UTF8Encoding encoding = new System.Text.UTF8Encoding();
byte[] keyByte = encoding.GetBytes(secretKey);

        HMACSHA1 hmac = new HMACSHA1(keyByte);
byte[] messageBytes = encoding.GetBytes(signatureString);
byte[] hashmessage = hmac.ComputeHash(messageBytes);
returnByteArrayToHexString(hashmessage); //Convert Byte to String.
    }
private string ByteArrayToHexString(byte[] Bytes)
    {
      stringHexAlphabet = "0123456789ABCDEF";
foreach (byte B in Bytes)
        {
Result.Append(HexAlphabet[(int)(B >> 4)]);
Result.Append(HexAlphabet[(int)(B & 0xF)]);
        }
returnResult.ToString();
    }
```

The following is a sample code snippet of HMACSHA1in **J2EE** programming language.

```
public static String getHMAC (String data, String key) {
String HMAC_SHA1_ALGORITHM = "HmacSHA1";
String result = null;

try {
Key signingKey = new SecretKeySpec(key.getBytes(), HMAC_SHA1_ALGORITHM);
Mac mac = Mac.getInstance(HMAC_SHA1_ALGORITHM);
mac.init(signingKey);
byte[] rawHmac = mac.doFinal(data.getBytes());
result = Base64.encodeBase64String(rawHmac);
}
catch (Exception e) {
e.printStackTrace();
}

return result;
}
```

The following is a sample code snippet of HMACSHA1in **PHP** programming language.

```
<?php

$signData = hash_hmac('sha1',
"00000005000015082334455555764Invoice00345MerchantID01Product01UserDefined
1UserDefined2UserDefined3",'746D7SCHAIQ0QUZ0MRJWU0PQ3AD7PJ8B', false);
$signData=  strtoupper($signData);
echourlencode($signData);
 ?>
```

4. Inquiry Request/Response

i. Field description of Inquiry Request

No	Variable	Data Type	Length	Mandatory	Description	Remark
1	merchantID	Character	15	Y	Merchant ID	Merchant ID generated by MPU's acquiring Bank.
2	invoiceNo	Character &Numeric	20	Y	Unique Invoice No	Provided by Merchant. The invoice number needs to be unique to trace the transaction. Please pad '0' to the left in the case of generated invoice number length is shorter than 20.
3	actionType	Character	1	Y	Transaction action type	I = Inquiry
4	hashValue	Character	150	N	Hash value computed by HMACSHA1 with secret key provided by MPU System.	hashValue will be generated by the following methods: 1. All the input parameters, which are filled in by customer, are sorted by ASCII to be Signature String e.g. **150823344555557 64I** Please see sample code for details. 2. Signature String will be encrypted by HMACSHA1 with secret key provided by MPU System.

ii. Field descriptions of Inquiry Response

No	Variable	Data Type	Length	Mandatory	Description	Remark
1	merchantID	Character	15	Y	Merchant ID	Merchant ID generated by MPU's acquiring bank. It can be optional. If it have the value in request, it will be responded back to merchant.
2	respCode	Character	2	Y	The code whether the transaction is successful or not.	00 = Approved; 04= Not Found Refer to Appendix A for all result code.

3	pan	Character	16	Y	Masked Card Number	First 4 and last 4 digits of card number Example: 954400xxxxxx1111
4	amount	Numeric	12	Y	Transaction Amount	The amount will be padded with '0' from the left and include no decimal point. Example: 1.00 = 000000000100, 1.5 = 000000000150 Currency exponent follows standard ISO4217 currency codes.
5	invoiceNo	Character &Numeric	20	Y	Unique Invoice No	Provided by Merchant. The invoice number needs to be unique to trace the transaction. Please pad '0' to the left in the case of generated invoice number length is shorter than 20.
6	tranRef	Character	28	Y	Transaction Reference for MPU.	Issued by MPU PGW System.
7	approvalCode	Character	6	Y	Transaction Approval Code from Card Issuer Host	
8	dateTime	Numeric	14	Y	Process Date Time value with yyyyMMddhhmmss format	
9	status	Character	2	Y	Last status of transaction	AP= Approved; Refer to Appendix B
10	failReason	Character	100	N	Reason of failure	Refer to Appendix B
11	categoryCode	Character &Numeric	20	N	Category Code	Merchant can distinct the transaction by adding category code.
12	userDefined1	Character	150	N	Merchant Defined Information	(Optional) MPU system will response back to merchant whatever information include in request message of this field.
13	userDefined2	Character	150	N	Merchant Defined Information	(Optional) MPU system will response back to merchant whatever information include in request message of this field.
14	userDefined3	Character	150	N	Merchant Defined Information	(Optional) MPU system will response back to merchant whatever information include in request message of this field.
15	hashValue	Character	150	Y	Hash value computed by HMACSHA1 with secret key provided by MPU System.	hashValue will be generated by the following methods: 1. All the input parameters, which are filled in by customer, are sorted by ASCII to be Signature String

						e.g. **000000050000150823344555555764Invoice00345MerchantID01Product01UserDefined1UserDefined2UserDefined3**. 2. Signature String will be encrypted by HMACSHA1 with secret key provided by MPU System.

5. Void Request/Response

i. Field description of Void Request

No	Variable	Data Type	Length	Mandatory	Description	Remark
1	merchantID	Character	15	Y	Merchant ID	Merchant ID generated by MPU's acquiring Bank.
2	invoiceNo	Character &Numeric	20	Y	Unique Invoice No	Provided by Merchant. The invoice number needs to be unique to trace the transaction. Please pad '0' to the left in the case of generated invoice number length is shorter than 20.
3	actionType	Character	1	Y	Transaction action type	V = Void
4	amount	Numeric	12	Y	Transaction Amount	The amount needs to be padded with '0' from the left and include no decimal point. Example: 1.00 = 000000000100, 1.5 = 000000000150 Currency exponent follows standard ISO4217 currency codes.
5	currencyCode	Numeric	3	Y	Standard ISO4217 Currency Codes.	Refer to Appendix D
6	hashValue	Character	150	N	Hash value computed by HMACSHA1 with secret key provided by MPU System.	hashValue will be generated by the following methods: 1. All the input parameters, which are filled in by customer, are sorted by ASCII to be Signature String e.g. **000000050000150823344555555764104V** . Please see sample code for details. 2. Signature String will be encrypted by HMACSHA1 with secret key provided by MPU System.

ii. Field descriptions of Void Response

No	Variable	Data Type	Length	Mandatory	Description	Remark
1	merchantID	Character	15	Y	Merchant ID	Merchant ID generated by MPU's acquiring bank. It can be optional. If it have the value in request, it will be responded back to merchant.
2	respCode	Character	2	Y	The code whether the transaction is successful or not.	00 = Voided 01 = Not Approved 02 = Voided Already 03 = Amount Mismatched 04 = Not Found
3	pan	Character	16	Y	Masked Card Number	First 4 and last 4 digits of card number Example: 954400xxxxxx1111
4	amount	Numeric	12	Y	Transaction Amount	The amount will be padded with '0' from the left and include no decimal point. Example: 1.00 = 000000000100, 1.5 = 000000000150 Currency exponent follows standard ISO4217 currency codes.
5	invoiceNo	Character &Numeric	20	Y	Unique Invoice No	Provided by Merchant. The invoice number needs to be unique to trace the transaction. Please pad '0' to the left in the case of generated invoice number length is shorter than 20.
6	tranRef	Character	28	Y	Transaction Reference for MPU.	Issued by MPU PGW System.
7	approvalCode	Character	6	Y	Transaction Approval Code from Card Issuer Host	
8	dateTime	Numeric	14	Y	Process Date Time value with yyyyMMddhhmmss format	
9	status	Character	2	Y	Last status of transaction	AP= Approved; Refer to Appendix B
10	hashValue	Character	150	Y	Hash value computed by HMACSHA1 with secret key provided by MPU System.	hashValue will be generated by the following methods: 1. All the input parameters, which are filled in by customer, are sorted by ASCII to be Signature String e.g. **000000050000150823344555555764Invoice00345**

						MerchantID01Product01. 2. Signature String will be encrypted by HMACSHA1 with secret key provided by MPU System.

6. Preparing and sending inquiry/void request message using HTTP form post

Test:https://122.248.120.252:60145/UAT/Payment/Action/api
Production:https://www.mpu-ecommerce.com/Payment/Action/api

Request

http://22.248.120.252:60145/UAT/Payment/Action/api?merchantID=400123456712345&tranRef=9549&actionType=I&hashValue=34E8E91C29E73B9648011FADBAE19849B520B24A

Response

```
merchantID=400123456712345&
respCode=00&pan=954433XXXXXX1111&amount=000000001000&invoiceNo=1234567890333
&tranRef=123&approvalCode=234567& dateTime= 20130430111211& status=AP&
userDefined1=userDefined1& userDefined2=userDefined2& userDefined3=userDefined3
&hashValue=34E8E91C29E73B9648011FADBAE19849B520B24A
```

Appendix A - Result Code Table

Result Code	Result Description
00	Approved
01	Refer to Card Issuer
02	Refer to Issuer's Special Conditions
03	Invalid Merchant ID
04	Pick Up Card
05	Do Not Honor
06	Error
07	Pick Up Card, Special Conditions
08	Honor with ID
09	Request in Progress
10	Partial Amount Approved
11	Approved VIP
12	Invalid Transaction
13	Invalid Amount
14	Invalid Card Number
15	No Sun Issuer

16	Approved, Update Track 3
17	Customer Cancellation
18	Customer Dispute
19	Re-enter Transaction
20	Invalid Response
21	No Action Taken
22	Suspected Malfunction
23	Unacceptable Transaction Fee
24	File Update not Supported by Receiver
25	Unable to Locate Record on File
26	Duplicate File Update Record
27	File Update Field Edit Error
28	File Update File Locked Out
29	File Update not Successful
30	Format Error
31	Bank not Supported by Switch
32	Completed Partially
33	Expired Card - Pick Up
34	Suspected Fraud - Pick Up
35	Contact Acquirer - Pick Up
36	Restricted Card - Pick Up
37	Call Acquirer Security - Pick Up
38	Allowable PIN Tries Exceeded
39	No Credit Account
40	Requested Function not Supported
41	Lost Card - Pick Up
42	No Universal Amount
43	Stolen Card - Pick Up
44	No Investment Account
45	Settlement Success
46	Settlement Fail
47	Reserved
48	Cancel Fail
49	No Transaction Reference Number
50	Host Down
51	Insufficient Funds
52	No Cheque Account
53	No Savings Account
54	Expired Card
55	Incorrect PIN
56	No Card Record
57	Trans. not Permitted to Cardholder
58	Transaction not Permitted to Terminal
59	Suspected Fraud
60	Card Acceptor Contact Acquirer

61	Exceeds Withdrawal Amount Limits
62	Restricted Card
63	Security Violation
64	Original Amount Incorrect
65	Exceeds Withdrawal Frequency Limit
66	Card Acceptor Call Acquirer Security
67	Hard Capture - Pick Up Card at ATM
68	Response Received Too Late
69	Reserved
70	Settle amount cannot more than authorized amount
71	Inquiry Record Not Exist
72	Reserved
73	Reserved
74	Reserved
75	Allowable PIN Tries Exceeded
76	Invalid Credit Card Format
77	Invalid Expiry Date Format
78	Invalid Three Digits Format
79	Reserved
80	User Cancellation by closing Internet Browser
81	Reserved
82	Reserved
83	Reserved
84	Reserved
85	Reserved
86	ATM Malfunction
87	No Envelope Inserted
88	Unable to Dispense
89	Administration Error
90	Cut-off in Progress
91	Issuer or Switch is Inoperative
92	Financial Institution not Found
93	Trans Cannot be Completed
94	Duplicate Transmission
95	Reconcile Error
96	System Malfunction
97	Reconciliation Totals Reset
98	MAC Error
99	System Unavailable

Appendix B – Transaction Status Table

Status	**Result Description**

AP	Approved
SE	Settled
VO	Voided
DE	Declined
FA	Failed
RE	Refunded

Appendix C – Currency Codes

CurrencyCode	**Currency Name**
104	Myanmar Kyat (MMK)

=====End of Document =====