



## **ACCOSA-PG Integration Document**

**Version 0.3**

**Last Updated:  
Jun 30, 2009**

For more information, please contact:

**enStage Inc.**  
1174, First Floor, 12<sup>th</sup> B Main  
HAL 2<sup>nd</sup> Stage, Indiranagar  
Bangalore  
[www.enstage.com](http://www.enstage.com)  
Ph: +91-80-3272 2080  
Email: [ipgsupport@enstage.com](mailto:ipgsupport@enstage.com)

## Revision History

Version Number	Modification Made	Modified By	Date
0.1	Base Document	Akshath	Jun 21, 2009
0.2	Updated Doc	Harsha H P	Jun 26, 2009
0.3	Updated integration section	Akshath	Jun 30, 2009
0.4	Updated appendix	Deepak	July 07, 2009

# Table of Contents

1 Introduction..... 4

2 Transaction Flow..... 4

3 Working with Integration Kit ..... 5

4 Request Format..... 6

5 Response Format..... 7

6 Appendix A (Transaction Status Code)..... 8

7 Appendix B (PG Error Codes) ..... 8

## 1 Introduction

enStage has created an internet payment – ACCOSA-PG that offers payment services to Billers, Merchants and Consumers.

This document describes how a merchant will have to integrate with enStage's ACCOSA-PG. enStage's ACCOSA-PG can be deployed as a hosted setup or in-house at the bank's premises. In both cases, the integration specs are the same.

There may be customizations based on how the Payment Gateway is setup and the details of that will be arrived at during implementation.

## 2 Transaction Flow

- Customer Selects "PAY" or "CHECKOUT" option on the merchant website
- Merchant computes request (as outlined below) using provided integration kit and forwards (http post) the customer with transaction details to ACCOSA-PG servers over an https (SSLv3) connection.
- Customer Selects "PAY" or "CHECKOUT" option on the merchant website
- Merchant computes request (as outlined below) using provided integration kit and forwards the customer with transaction details to ACCOSAPG servers over an https (SSLv3) connection.
- ACCOSA-PG disassembles the transaction information to confirm transaction origination and stores details of the transaction into its databases
- ACCOSA-PG asks customers to enter details of the credit card/debit card to be used for making the payment. The details asked are :
  - Type of Card (Visa/MasterCard)
  - Card Number
  - Name of Card Holder
  - Date of Expiry
  - CVV2 Value
  - Email address (optional)
- ACCOSA-PG also stores the IP address from where the client is performing this transaction.
- ACCOSA-PG packages the transaction and payment details and forwards it for 3DS authorization via ACCOSA MPI.
- On successful completion of the authentication and authorization, the details are stored into the ACCOSA-PG databases securely.
- ACCOSA-PG sends a POST request to merchant designated URL with the details of transaction status.
- Transaction Status Code of 50020 and pg\_error\_code of 0 means transaction was successful and any other status code or error code means it failed. Refer to appendix for more info.
- Merchant can see order report anytime online through the merchant admin module.
- For orders which get successfully authorized, consumer's card limit is booked to the value of order amount. But the card is actually debited only after the

merchant ships the transaction through the merchant admin module and PG subsequently settles it.

### 3 Working with Integration Kit

Use this procedure to install the Integration kit

1. Please refer to your 'Welcome' email from PG Admin Team and get the following details
  - ✓ Merchant Admin Console URL
  - ✓ Your Merchant Admin Console Username
  - ✓ Your Merchant Admin Console Password
  - ✓ Your Merchant ID
  - ✓ Your PG Name
  - ✓ Your Admin Console Password
2. Download the Integration kit from the 'Merchant Admin Console' site for one of the following appropriate languages
  - ✓ asp/asp.net
  - ✓ jsp/java
  - ✓ php
3. Modify the config file with required values. Values for these would be displayed in your Merchant Admin Console's 'Download Integration Kit' section. Do refer: "change me" edit fields

String pgInstanceId = "<change me>"; --> PG instance ID being added here

String merchantId = "<change me>"; --> Merchant ID being added here

String hashKey = "<change me>"; -----> hash key value should be added here

4. You may now host these files on your web/app server and provide the full URL to 'PG Admin Team' that points to the pgresponse page e.g.;  
<http://www.merchant.com/pg/pgresponse.jsp> on your web server.
5. 'PG Admin team' will now enable your merchant as 'TEST' merchant. This will enable you to perform test transaction without incurring any financial impact.
6. Now you may start a test transaction by accessing index page in the kit e.g.;  
<http://www.merchant.com/pg/index.html>
7. Now integrate these files on your merchant shopping site and update 'PG Admin Team' if the pgresponse page needs to be modified.
8. Fire a test transaction and confirm if every thing is working fine.
9. Now request 'PG Admin team' to mark you as 'ACTIVE' merchant, so that you can do production transactions.

## 4 Request Format

\*M/O = Mandatory/Optional/Conditional

Html Field Name	M/O/C	Description	Example
pg_instance_id	M	PG instance ID	79070160
merchant_id	M	PG Merchant ID	44065131
perform	M	Action To Be Performed	initiatePaymentCapture#sale initiatePaymentCapture#preauth
device_category	O	0 for Web,	0
currency_code	O	ISO Code for the currency of amount field. For INR this is 356. If not passed default currency would be used. Len: 3 char.	356
amount	M	Purchase amount including the minor units of currency with all punctuation removed i.e., implied decimals and no formatting	100
merchant_reference_no	M	Merchant reference number	AX143565
order_desc	M	Brief description of items purchased. Len: 0-125 chars	Apple iPod 20GB
recur_frequency	O	Minimum number of days between authorizations. This is applicable only for recurring transactions on the merchant site.	30
recur_expiry	O	The date after which no further authorizations should be performed as part of the recurring set. Format: YYYYMMDD	20061225
installments	O	Indicates the maximum number of permitted authorizations for installment/recurring payments. Len: 0-3 char and must be > 1(if not empty)	4
message_hash	M	<p>This is a hash of the fields sent to ensure that no modification is done in transit between Merchant/PG and web servers.</p> <p><i>Composition of message_hash</i></p> <p>STANDARD:6:Base64(SHA1(pg_instance_id merchant_id perform amount merchant_reference_no key)) OR CURRENCY:7:Base64(SHA1(pg_instance_id merchant_id perform currency_code amount merchant_reference_no key)) OR INSTALLMENT:9:Base64(SHA1(pg_instance_id merchant_id perform amount merchant_reference_no recur_frequency recur_expiry installments key))</p> <p>When the PG receives the request, it follows the same process as above to re-generate the hash and check it with the message_hash field. An exact match ensures that the message has not been modified in transit.</p> <p>Note: PG will provide a software kit for hash generation and verification.</p>	See description. Note: PG will provide a software kit for request generation (including hash) and verification. No need to code for it.

The page will have to be posted to following URL:

**<https://<Payment Gateway Domain>/AccosaPG/verify.jsp>**

## 5 Response Format

\*M/O/C = Mandatory/Optional/Conditional

Html Field Name	M/O/C	Description	Example
pg_instance_id	M	PG instance ID	79070160
merchant_id	M	PG Merchant ID	44065131
transaction_type_code	M	transaction type code	9001 = Sale 9009 = PreAuth
installments	M	maximum num of authorizations	0
transaction_id	M	Unique value generated for each transaction internally by the PG	123456
amount	M	Purchase amount including the minor units of currency with all punctuation removed i.e., implied decimals and no formatting	100
exponent	M	exponent of amount The minor units of currency specified in ISO 4217. This value can be setup differently for each Acquirer Bank. For example, US Dollars has a value of 2; Japanese Yen has a value of 0. Default value is 2, which is applicable for INR.	2
currency_code	M	ISO Code for the currency of amount field. For INR this is 356. Len: 3 char.	356
merchant_reference_no	M	Merchant reference number	AX143565
status	M	Transaction Status	50020 = SUCCESS (any other value is a failure, full status codes shall be provided in appendix A)
batch_id	C	Transaction Batch ID	1002
approval_code	C	Approval code of Transaction	560252
3ds_eci	C	<i>ECI suggested by MPI</i>  <i>ECI for Visa</i> 05 – successful authentication 06 – attempted authentication 07 – failed authentication  <i>ECI for MasterCard</i> 01 – authentication failed/not available 02 – successful authentication 00 – failed authentication	05
3ds_cavv_aav	C	Present only if ECI is 05,06 or 02. It is base64 encoded. Len = 28 characters	AAABBGkgUhI0VngQACBSA AAAAAA=
3ds_status	O	Y – success A – attempts N – failure U – unable to authenticate	
pg_error_code	M	PG Error Code	0 = No Error (any other value means some kind of error, full error codes list shall be provided in appendix B)
pg_error_detail	O	PG Error detail	orderDesc was not set
pg_error_msg	O	PG Error Msg	Bad Input Data in Request

message_hash	M	<p>This is a hash of the fields sent to ensure that no modification is done in transit between PG and Merchant web servers.</p> <p><i>Composition of message_hash</i></p> <p>13:Base64(SHA1(pg_instance_id merchant_id transaction_type_code installments transaction_id amount exponent currency_code merchant_reference_no status 3ds_e ci pg_error_code key))</p> <p><i>Note: PG will provide a software kit for hash generation and verification.</i></p>	See description. Note: PG will provide a software kit for response verification (including hash). No need to code for it.
--------------	---	---	---

## 6 Appendix A (Transaction Status Code)

STATUS CODES	DESCRIPTION
50010	Init
50011	Capture Aborted
50012	3DS Start
50013	3DS Completed
50014	3DS Failed
50015	3DS Aborted
50016	Switch Start
50017	Switch Timeout
50018	Switch Aborted
<b>50020</b>	<b>Success</b>
50021	Failed
50097	Test Transaction

## 7 Appendix B (PG Error Codes)

PG ERROR CODES	Description
<b>0</b>	<b>No Error</b>
1	Call Issuer
2	Contact Switch Admin
3	Retry After Some Time.
10001	Disabled Instance
10002	Test Instance
10003	Instance under Maintenance
10004	Internal Server Error
10005	Invalid Data Sent to Switch
10006	Internal Error caused contact Switch Admin



## ACCOSA-PG (Payment Gateway) Integration Document

---

10011	Disabled Acquirer
10012	Test Acquirer
10013	Acquirer under Maintenance
10021	Disabled Merchant
10022	Test Merchant
10023	Merchant under Maintenance
10024	Bad Input Data in Request
10025	PGInterface not allowed
10026	Merchant velocity check failed
10030	Capture Aborted
10031	Auth Aborted
10032	Card Association not enabled
10033	Card Range not enabled
10040	Transaction not allowed - flow error
12001	Acquirer Server Error
12002	Acquirer Timeout
12003	Acquirer Down
12004	Acquirer Declined
12005	Batch Closed
12006	Totals Mismatched
12007	Unable to settle
13001	Issuer Server Error
13002	Issuer Timeout
13003	Issuer Down
13004	Issuer Declined
13005	Invalid Amount
13006	Issuer Insufficient Funds
14001	3DS Failed
14002	3DS Aborted
14003	MPI Error