BluePay 2.0 Trans Notify Post

BluePay Trans Notify Post Webhook

Reference Guide

April 2025

http://www.fiserv.com

This document is classified as Fiserv Public.

Content

About this Document

This documentation provides technical guidance on BluePay's capability to send merchants a notifications after a transaction has been processed.

The system described in the document may be subject to minor changes.

Intended Audience

This document is written for merchants, partners, and developers who will be responsible for integrating payment processing functionality and the notification system with the BluePay Payment Gateway.

Assistance & Feedback

Use the following contact information to request Trans Notify Post setup, request help with BluePay Payment Gateway integration, or provide feedback on this document. When requesting setup, please supply the URL where transaction data should be posted and specify if all transactions should be sent or only a subset of transactions.

Support Level	Contact Details
BluePay Integration Support Team	bluepay-integration@fiserv.com

Support hours are Monday through Friday 8:00am to 5:00pm (CST UTC-6).

Overview

BluePay Trans Notify Post or Webhook notifies merchants about the transactions details through HTTP POST requests to their servers. These notifications include vital transaction information such as account ID, transaction ID, amount, and status.

This user-friendly system enhances merchant convenience and ensures efficient management of transaction processes. The BluePay Notify Post enables merchants to stay informed and effectively manage their transaction activities.

Response Format

BluePay expects the customer's server to respond with status code '200.' If the response differs, BluePay assumes the POST failed and may be retired.

Output Parameters

Within the body of that POST are uri-encoded name=value pairs containing the field names and values listed below.

TPS_HASH_TYPE	
Description:	The algorithm used to compute the BP_STAMP.
Valid Value:	• MD5 • SHA256 • SHA512 • HMAC_SHA256 • HMAC_SHA512
See the BP_STAMP section for more details.	

BP_STAMP_DEF	
Description:	List of the fields used to calculate the BP_STAMP.
See the BP_STAMP section for more details.	

BP_STAMP	
Description:	• It is Hash created from the merchant's Secret Key and the concatenated values of specific fields returned from BluePay. • BP_STAMP_DEF determines the exact fields and their order and TPS_HASH_TYPE determines the hash type. • This process is similar to handling the TAMPER_PROOF_SEAL in inbound APIs.
Length:	Length 32, 64, or 128 (depends on TPS_HASH_TYPE)
See the BP_STAMP section for more details.	

account_id	
Description:	The gateway account ID used to process the transaction.
Length:	12

trans_id	
Description:	The 12-digit transaction ID that BluePay assigns to this transaction.
Length:	12

master_id	
Description:	If this transaction references a prior transaction (such as for REFUND, or VOID) it will be here.
Length:	12

rebill_id	
Description:	If the transaction was part of a rebilling, this field contains the id of that rebilling.
Length:	12

card_account	
Description:	The system returns payment account details used for the transaction. • For credit card transactions, the system returns 12 x's followed by the last four digits. • For check transactions, the system returns the string "<account type>:<routing number>:<X's><last four digits of account number>."
Valid Value:	Account Type: • 'C' is checking • 'S' is savings
Length:	32
Example:	"C:123123123:xxxxxx4321"

card_expire	
Description:	Expiration date for a credit card. This is a blank for an ACH.
Length:	04

bank_name	
Description:	The name of the card issuing bank or bank ACH account.
Length:	64

amount	
Description:	The monetary amount for which the transaction was run.
Length:	09

trans_status	
Description:	The status of the transaction.
Valid Value:	• '1' for approved • '0' for declined • 'E' for error
Length:	01

fancy_status	
Description:	The text description of the transaction status.

trans_type	
Description:	The type of transaction run.
Valid Value:	• 'AUTH' • 'CAPTURE' • 'CREDIT' • 'REFUND' • 'SALE' • 'VOID'
Length:	08

card_type	
Description:	A four-character indicator of the credit card type used, if any.
Valid Value:	• AMEX = American Express • MC = Mastercard • DISC = Discover • VISA = VISA • JCB = JCB • DCCB = Diner's Club or Carte Blanche • ENRT = EnRoute • BNKC = BankCard • SWTC = Switch • SOLO = Solo
Length:	04

payment_type	
Description:	The type of payment used for the transaction.
Valid Value:	• 'ACH' for ACH transactions • 'CREDIT' for credit card transactions.
Length:	08

origin	
Description:	The name of the source that originated the transaction.
Valid Value:	• bp10emu: BluePay 1.0 Post • bp20post: BluePay 2.0 Post • asbyemu: Assurebuy Emulation mode • a.net-aim: Authorize.net Emulation mode • VTerm: Input on Virtual Terminal • AGG: Aggregation • BATCH • CAPQUEUE • FRAUDSCRUB • REBILL • REJECT: Automatic reject or chargeback from bank
Length:	16

order_id	
Description:	The merchant-supplied or system supplied order id.
Length:	128

invoice_id	
Description:	The merchant-supplied or system supplied invoice id.
Length:	64

name1	
Description:	The cardholder's name 1.
Length:	32

name2	
Description:	The cardholder's name 2.
Length:	32

company_name	
Description:	The cardholder's company name.
Length:	64

addr1	
Description:	The cardholder's address 1.
Length:	64

addr2	
Description:	The cardholder's address 2.
Length:	64

city	
Description:	The cardholder's city.
Length:	32

state	
Description:	The cardholder's state.
Length:	16

zip	
Description:	The cardholder's zip.
Length:	16

country	
Description:	The cardholder's country.
Length:	64

memo	
Description:	A memo that the system submits during the transaction.
Length:	128

phone	
Description:	The cardholder's phone number.
Length:	16

email	
Description:	The cardholder's email address.
Length:	64

auth_code	
Description:	The authorization code returned by the front-end processor. This field displays the reject code on ACH VOIDS.
Length:	08

message	
Description:	Message value returned by the card issuing bank.
Length:	64

issue_date	
Description:	The date when the transaction was entered into BluePay.
Data Type:	Timestamp without time zone
Example:	"yyyy-mm-dd hh:mm:ss"

avs_result	
Description:	The AVS result that the front-end processor returns after the transaction.
Length:	01

cvv2_result	
Description:	The CVV2 result that the front-end processor returns after the transaction.
Length:	01

custom_id1	
Description:	The merchant-supplied value for custom ID 1.
Length:	16

custom_id2	
Description:	The merchant-supplied value for custom ID 2.
Length:	64

f_void	
Description:	A flag to identify if the transaction was a void.
Length:	01

account_name	
Description:	The name of the merchant's account in BluePay.
Length:	32

mode	
Description:	The mode of the merchant's transaction.
Valid Value:	LIVE or TEST
Length:	08

dba_name	
Description:	A doing-business-as name of the merchant.
Length:	128

merchdata	
Description:	All MERCHDATA values combined.

merchdata_xxxxx	
Description:	Each individual MERCHDATA value is returned. The merchant provided portion of the field name will be lower case.

level_3_data	
Description:	All LV3 values combined. A linefeed character is inserted between records.

lodging_data	
Description:	All Lodging values combined.

vehicle_rental_data	
Description:	All Vehicle Rental values combined.

ach_same_day_funding	
Description:	Available only for payment_type = 'ACH'. Contains the flag value (0 or 1) to indicate the presence or absence of the same-day funding feature.
Valid Value:	• '1' (True) implies same-day funding • '0' (False) implies absence of same-day funding
Length:	01

validation_result	
Description:	Status value that includes bank account validation results.
Valid Value:	• '15': Known bad bank account. The transaction is immediately declined (validation fee charged for LIVE transactions) • '20': Unknown bank account but with a valid format (validation fee charged for LIVE transactions) • '25': Unknown bank account (validation fee charged for LIVE transactions) • '35': Bank account found but pending transaction settlement (validation fee charged for LIVE transactions) • '45': Known good bank account (validation fee charged for LIVE transactions) • 'B': Account validation bypassed (no validation fee charged) • 'R': Bank Account received known bad previously (no validation fee charged) • 'E': Error, Bank account validation failed (no validation fee charged) • null: Account validation not performed (no validation fee charged)
Length:	02

BP_STAMP

BluePay uses cryptographic hash (or "digest") functions as a means of both protecting transaction data from being altered and ensuring that the transaction is genuine. A cryptographic hash function is an algorithm that maps data of any size to a bit string of a fixed size that cannot be deciphered.

- All merchants have a default hash type assigned to their account.
- This can be viewed and updated on the merchant's Account Admin page of BluePay's Gateway (https://secure.bluepay.com) under "Hash Type in APIs."
- When using this API, the TPS_HASH_TYPE will always be the merchant's "Hash Type in APIs."

The TPS_HASH_TYPE may be any of the following algorithms (in hexadecimal form).

TPS_HASH_TYPE	Description
MD5	Use md5sum or a similar program to calculate a 128-bit hash, then convert it into hexadecimal form; the result is 32 hexadecimal characters.
SHA256	Use sha256sum or a similar program to calculate a 256-bit hash, then convert it into hexadecimal form; the result is 64 hexadecimal characters.
SHA512	Use sha512sum or a similar program to calculate a 512-bit hash, then convert it into hexadecimal form; the result is 128 hexadecimal characters.
HMAC_SHA256	A 128-bit hash, resulting in a sequence of 64 hexadecimal characters.
HMAC_SHA512	A 128-bit hash, resulting in a sequence of 128 hexadecimal characters

Steps to find the HMAC of either SHA256 (HMAC_SHA256) or SHA512 (HMAC_SHA512):

1. Compare the length of the key (the merchant's Secret Key) to the hash's input blocksize. SHA256 blocksize = 64, SHA512 blocksize = 128.
    - If the length of the key is > blocksize, set the key's value to the hash of the original key.
    - If the length of the key is < blocksize, pad the key to the right with zeros until its length equals the blocksize.
2. Create the inner key (inner_key):
   Create an inner padding value of 0x36 repeated the blocksize number of times. Perform a bitwise exclusive-OR (XOR) on the key and the inner padding to create the inner key.
3. Create the outer key (outer_key):
   Create an outer padding value of 0x5c repeated the blocksize number of times. Perform a bitwise exclusive-OR (XOR) on the key and the outer padding to create the outer key.

4. Calculate the hash of the inner key concatenated with the text string, and then calculate the hash of the outer key concatenated with the previous hash result: hash(outer_key + hash (inner_key + string))
5. Convert the result into a hex string.

When using a program or function to calculate the BP_STAMP, make sure that it accepts a text string (or "message") argument and will return the hashed string (or "message digest") in hexadecimal form.

Calculating the BP_STAMP

Perform the following steps to calculate the value of BP_STAMP.

Step 1

Concatenate the values of the fields that make up the BP_STAMP_DEF in the same order that they are listed. Use "" (empty string - no space) as the value for any fields that are empty or unsent. The current BP_STAMP_DEF is ('+' represents string concatenation, and the field names represent the contents of the respective fields):

`message`	`= trans_id + trans_status + trans_type + amount + batch_id + batch_status + total_count + total_amount + bupload_id + rebill_id + reb_amount + status`

Step 2

Calculate the expected BP_STAMP and compare that value to the BP_STAMP in the response to verify that the response is genuine. The hash type used is the TPS_HASH_TYPE provided in the response from BluePay):

- If TPS_HASH_TYPE is 'MD5', 'SHA256', or 'SHA512', find the md5sum, sha256sum, or sha512sum of (the merchant's Secret Key + message) in hex format.
- If TPS_HASH_TYPE is 'HMAC_SHA256' or 'HMAC_SHA512', find the HMAC_SHA256 or HMAC_SHA512 of (the merchant's Secret Key, message) in hex format.

Finally, the merchant should take the result and compare it to the value of BP_STAMP. If they match, the response is genuine. If they do not, the response has been tampered with somehow.

Example

Merchant B's account information

Data Field	Values
Secret Key	"abcdabcdabcdabcd"
ACCOUNT_ID	"123412341234"

Merchant B's response post for a transaction might include the following output fields.

Output Field	Values
TPS_HASH_TYPE	? (We look at 3 examples)
BP_STAMP	? (This differs based on the TPS_HASH_TYPE used)
BP_STAMP_DEF	"trans_id trans_status trans_type amount batch_id batch_status total_count total_amount bupload_id rebill_id reb_amount status"
trans_id	"987654321001"
trans_status	1
trans_type	"SALE"
amount	"199.99"
rebill_id	"543215432154"

Step 1

Concatenate the values in the BP_STAMP_DEF to create a message string. Remember, if the field is not returned or if the value is undefined, use an empty string as that field's value.

message	`= trans_id + trans_status + trans_type + amount + batch_id + batch_status + total_count + total_amount + bupload_id + rebill_id + reb_amount + status`
	
	`= "987654321001" + "1" + "SALE" + "199.99" + "" + "" + "" + "" + "" + "" + "543215432154" + "" + ""`
	
	`= "9876543210011SALE199.99543215432154"`

Step 2

Calculate the expected BP_STAMP and compare that value to the BP_STAMP in the response to verify that the response is genuine. This step varies depending on which TPS_HASH_TYPE is included in the response.

- If the merchant's "Hash Type in APIs" = "MD5", the response would include:

Data Field	Value
TPS_HASH_TYPE	"MD5"
BP_STAMP	"5793c242a688f07a0e3e05dbc438bfbf"
Expected BP_STAMP	= md5sum(Secret Key + message) in hex format = md5sum("abcdabcdabcdabcd" + "9876543210011SALE199.99543215432154") in hex format = "5793c242a688f07a0e3e05dbc438bfbf"

Since the expected BP_STAMP matches the BP_STAMP from the response, the response is genuine.

- If the merchant's "Hash Type in APIs" = "HMAC_SHA256", the response would include:

Data Field	Value
TPS_HASH_TYPE	"HMAC_SHA256"
BP_STAMP	"58227eabad0c998141bbe62359176088a00ef037122370d10bba272429086900"
Expected BP_ STAMP	= HMAC_SHA256(Secret Key, message) in hex format = HMAC_SHA256("abcdabcdabcdabcd", "9876543210011SALE199.99543215432154") in hex format = "58227eabad0c998141bbe62359176088a00ef037122370d10bba272429086900"

Since the expected BP_STAMP matches the BP_STAMP from the response, the response is genuine.

Revision History

Version	Revision Date	Reason for Change
1.1	April 2025	• Updated the layout format of the document