PayPal™

Instant Payment Notification Guide

For Professional Use Only
Currently only available in English.

A usage Professional Uniquement
Disponible en Anglais uniquement pour l'instant.

Last updated: July, 2010

IPN Guide

Document Number: 10087.en_US-201007

Contents

Preface

This document describes the Instant Payment Notification (IPN) message service.

Intended Audience

This document is intended for merchants and developers implementing IPN.

Revision History

Revision history for *Instant Payment Notification Guide*.

***TABLE P.1** Revision History*

Date	Description
07/17/10	Updated information about IPN messages for Adaptive Payments and added information for Adaptive Accounts.
04/01/10	Updated IPN history and resend information and added information about IPN messages for Adaptive Payments.
06/26/09	Added information about IPN messages related to payment review. Updated the section about setting up IPN notifications on PayPal and added a note about the delivery of IPN messages to the notify URL.
06/10/09	Added IPN history and resend information and added information to the troubleshooting section.
03/16/09	Made additions to sample code.
02/16/09	First draft.

July, 2010

1 Introducing IPN

Instant Payment Notification (IPN) is a message service that notifies you of events related to PayPal transactions. You can use it to automate back-office and administrative functions, such as fulfilling orders, tracking customers, and providing status and other information related to a transaction.

IPN Overview

IPN notifies you when an event occurs that affects a transaction. Typically, these events represent various kinds of payments; however, the events may also represent authorizations, Fraud Management Filter actions and other actions, such as refunds, disputes, and chargebacks.

IPN is a message service that PayPal uses to notify you about events, such as

- Instant payments, including Express Checkout, Adaptive Payments, and direct credit card payments, and authorizations, which indicate a sale whose payment has not yet been collected
- eCheck payments and associated status, such as *pending*, *completed*, or *denied*, and payments pending for other reasons, such as those being reviewed for potential fraud
- Recurring payment and subscription actions
- Chargebacks, disputes, reversals, and refunds associated with a transaction

In many cases, the action that causes the event, such as a payment, occurs on your website; however, your website is not the only source of events. In many cases, events can be generated by Website Payment Standard buttons, the PayPal API, or by PayPal itself.

You detect and process IPN messages with a *listener*, sometimes called a *handler*, which is a script or program that you write. It waits for messages and passes them to various back-end or administrative processes that respond the messages. PayPal provides sample code that you can modify to implement a listener that detects IPN messages.

The actions to take when your listener is notified of an event are specific to your needs. Examples of the kinds of actions you might take when your listener receives an IPN message include the following:

- Trigger order fulfillment or enable media downloads when a check clears or a payment is made
- Update your list of customers
- Update accounting records
- Create specialized "to do" lists based on the kind of event

You are typically notified of events by email as well, but the IPN message service enables you to automate your response to events. The following diagram shows how events can occur and how PayPal responds with IPN messages that it sends to your listener:
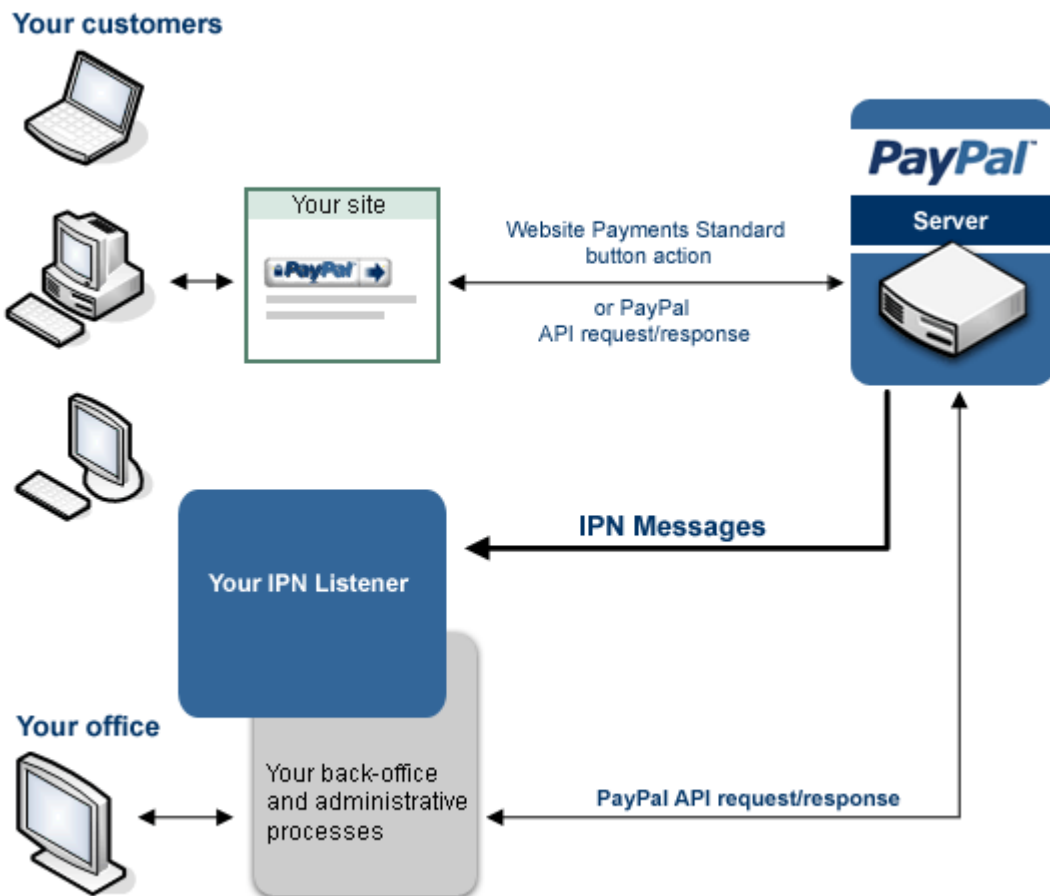
The diagram shows requests and responses, which are the result of processing button clicks or API operations on PayPal. PayPal sends an IPN message when it sends a response to a request. The IPN message is not actually part of the response sent to your website. Rather, the IPN message is sent to the your listener, which allows you to take actions that are not directly tied to the operation of your website.

NOTE: The diagram does not show the IPN authentication protocol messages that validate the IPN message.

IPN is an asynchronous message service, meaning that messages are not synchronized with actions on your website. Thus, listening for an IPN message does not increase the time it takes to complete a transaction on your website.

The IPN message service does not assume that all messages will be received by your listener in a timely manner. Because the internet is not 100% reliable, messages can become lost or delayed. To handle the possibility of transmission and receipt delays or failures, the IPN message service implements a retry mechanism that resends messages at various intervals until you acknowledge that the message has successfully been received. Messages may be resent for up to four days after the original message.

NOTE: Unless you are certain that a failure occurred on the the Internet, the most likely cause of lost, delayed, or duplicate IPN messages is faulty logic in the listener itself.

Because messages can be delivered at any time, your listener must always be available to receive and process messages; however, the retry mechanism also handles the possibility that your listener could become swamped or stop responding.

The IPN message service should not be considered a real-time service. Your checkout flow should not wait on an IPN message before it is allowed to complete. If your website waits for an IPN message, checkout processing may be delayed due to system load and become more complicated because of the possibility of retries.

IPN Protocol and Architecture

IPN is designed to be secure, reliable, and asynchronous. To meet these requirements, the protocol requires you to acknowledge receipt of IPN messages. The IPN service provides a retry mechanism to handle cases in which a message is not acknowledged; for example, when a transmission or receipt failure occurs.

When you enable IPN, PayPal sends messages to the IPN listener at the URL you specify in your account's profile. You can override the URL to associate other IPN listeners with specific transactions. In this case, you specify the listener's URL when you set up a Website Payment Standard button or a PayPal API operation.

The IPN protocol consists of three steps:

1. PayPal sends your IPN listener a message that notifies you of the event

2. Your listener sends the complete unaltered message back to PayPal; the message must contain the same fields in the same order and be encoded in the same way as the original message

3. PayPal sends a single word back, which is either `VERIFIED` if the message originated with PayPal or `INVALID` if there is any discrepancy with what was originally sent

Your listener must respond to each message, whether or not you intend to do anything with it. If you do not respond, PayPal assumes that the message was not received and resends the message. PayPal continues to resend the message periodically until your listener sends the

correct message back, although the interval between resent messages increases each time. The message can be resent for up to four days.

This resend algorithm can lead to situations in which PayPal resends the IPN message while you are sending back the original message. In this case, you should send your response again, to cover the possibility that PayPal did not actually receive your response the first time. You should also ensure that you do not process the transaction associated with the message twice.

IMPORTANT: PayPal expects to receive a response to an IPN message within 30 seconds. Your listener should not perform time-consuming operations, such as creating a process, before responding to the IPN message.

After PayPal verifies the message, there are additional checks that your listener or back-end or administrative software must take:

- Verify that you are the intended recipient of the IPN message by checking the email address in the message; this handles a situation where another merchant could accidentally or intentionally attempt to use your listener.

- Avoid duplicate IPN messages. Check that you have not already processed the transaction identified by the transaction ID returned in the IPN message. You may need to store transaction IDs returned by IPN messages in a file or database so that you can check for duplicates. If the transaction ID sent by PayPal is a duplicate, you should not process it again.

- Because IPN messages can be sent at various stages in a transaction's progress, make sure that the transaction's payment status is "completed" before enabling shipment of merchandise or allowing the download of digital media.

- Verify that the payment amount actually matches what you intend to charge. Although not technically an IPN issue, if you do not encrypt buttons, it is possible for someone to capture the original transmission and change the price. Without this check, you could accept a lesser payment than what you expected.

IPN Messages Generated by Website Payments Standard

PayPal generates an IPN message when your customer clicks a Website Payments Standard payment button, such as a Buy Now button, and completes the transaction on PayPal. You can use this notification to kick-off order fulfillment, enable digital media downloads, store information in a customer relationship management (CRM) or accounting system, and so on.

The following diagram shows both the web flow and the IPN message authentication protocol:

Checkout flow
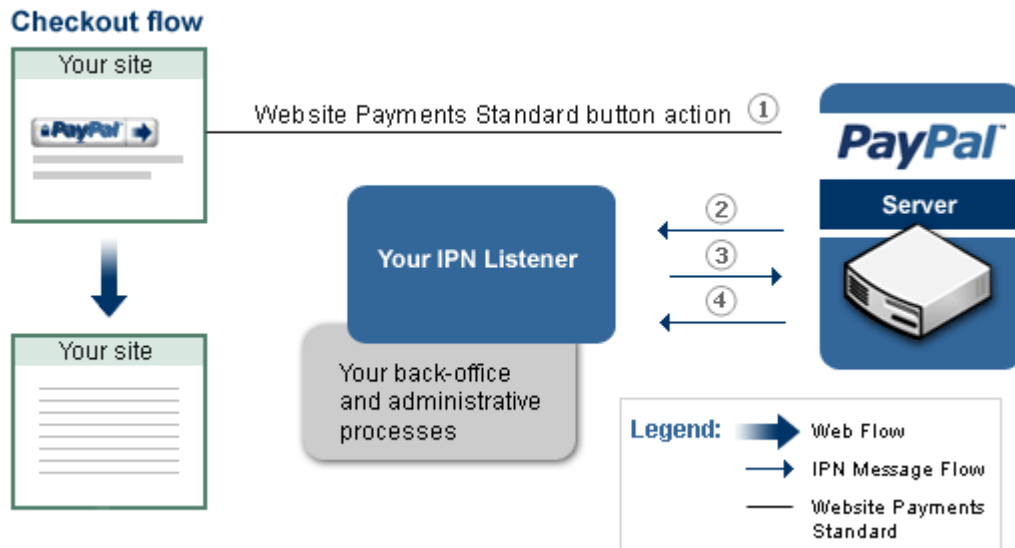
The numbers in diagram correspond to the following steps:
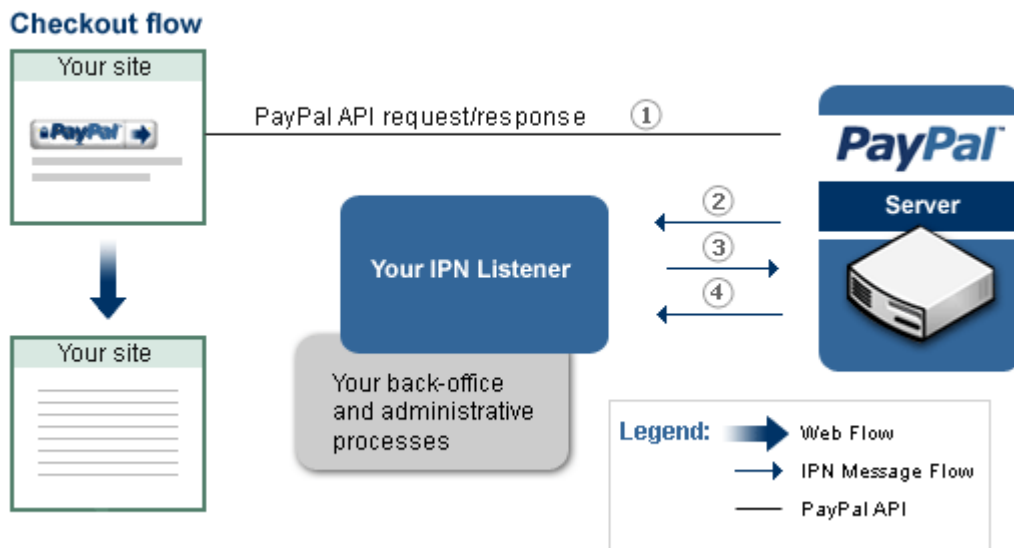
1. The button action initiates a payment that completes on PayPal

2. PayPal sends your IPN listener a message that notifies you of the event

3. Your listener sends the complete unaltered message back to PayPal; the message must contain the same fields in the same order and be encoded in the same way as the original message

4. PayPal sends a single word back, which is either `VERIFIED` if the message originated with PayPal or `INVALID` if there is any discrepancy with what was originally sent

Your IPN listener must implement the IPN authentication protocol (steps 2, 3, and 4 in this diagram). After successfully completing the protocol, your back-office or administrative process vets the contents of the message and responds appropriately. For example, if the payment status for the transaction is "Completed," your system can print a packing list or email a password to your customer for downloading digital media.

IPN Messages Generated by APIs

PayPal generates an IPN message when you invoke an API operation, such as `DoExpressCheckoutPayment` of `DoDirectPayment` during checkout or an Adaptive Payments `Pay`, `Preapproval`, or `ExecutePayment` API operation. You can use this notification to kick-off order fulfillment, enable digital media downloads, store information in a customer relationship management (CRM) or accounting system, and so on.

The following diagram shows both the web flow and the IPN message authentication protocol:

Checkout flow

The numbers in the diagram correspond to the following steps:

1. The API operation initiates a payment on PayPal.

2. PayPal sends your IPN listener a message that notifies you of the event

3. Your listener sends the complete unaltered message back to PayPal; the message must contain the same fields in the same order and be encoded in the same way as the original message

4. PayPal sends a single word back, which is either VERIFIED if the message originated with PayPal or INVALID if there is any discrepancy with what was originally sent

Your IPN listener must implement the IPN authentication protocol (steps 2, 3, and 4 in this diagram). After successfully completing the protocol, your back-office or administrative process vets the contents of the message and responds appropriately. For example, if the payment status for the transaction is "Completed," your system can print a packing list or email a password to your customer for downloading digital media.

IPN Messages Generated by a Back-Office Procedure

PayPal generates an IPN message when you perform actions that invoke the PayPal API, regardless of whether it is from your website or from a back-office or administrative procedure. You can use this notification to trigger an email to your customer, store information in a CRM or accounting system, and so on.

The following diagram shows both an administrative web flow and the IPN message authentication protocol:

Backend Process
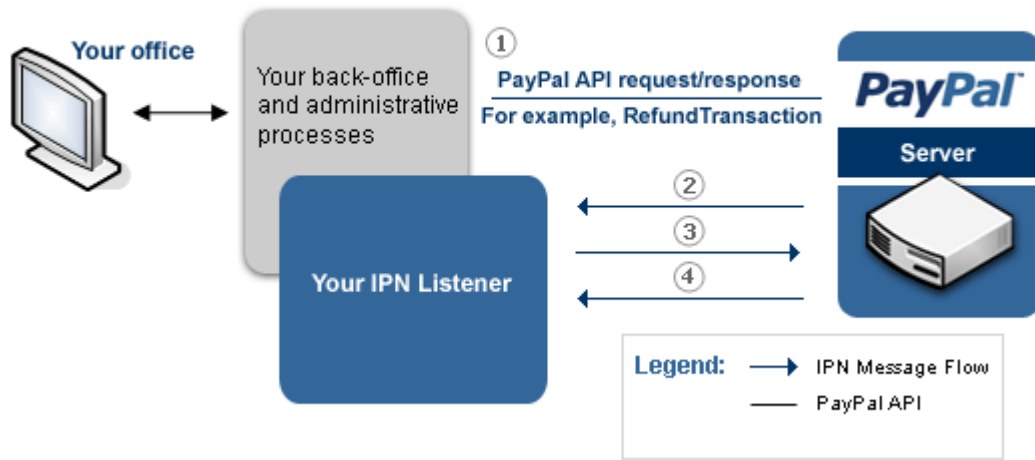
The numbers in diagram correspond to the following steps:

1. Your back-office or administrative process invokes a PayPal API operation; for example, it could invoke the `RefundTransaction` API operation when your employee issues a refund.

2. PayPal sends your IPN listener a message that notifies you of the event

3. Your listener sends the complete unaltered message back to PayPal; the message must contain the same fields in the same order and be encoded in the same way as the original message

4. PayPal sends a single word back, which is either `VERIFIED` if the message originated with PayPal or `INVALID` if there is any discrepancy with what was originally sent
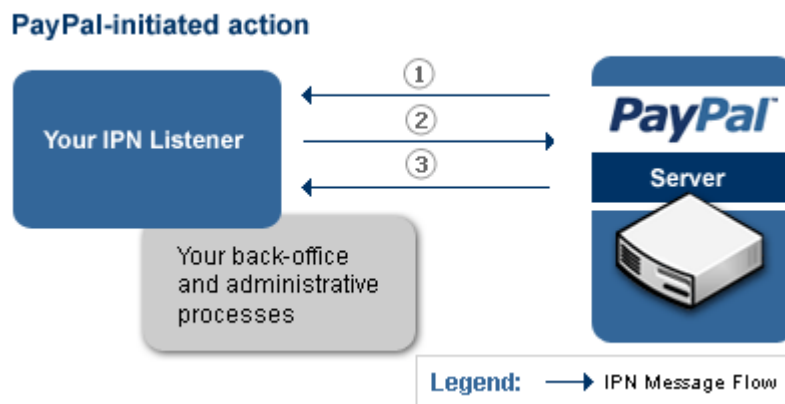
Your IPN listener implements the IPN authentication protocol (steps 2, 3, and 4 in this diagram). After successfully completing the protocol, your listener or back-office or administrative process vets the contents of the message and responds appropriately. For example, your system can notify the customer of the refund by email.

PayPal-Initiated IPN Messages

Some IPN messages generated by PayPal are not directly associated with a web flow. PayPal generates an IPN message when external events arise that might affect a transaction, such as disputes, chargebacks, echeck clearing, and various recurring payment and subscription events.

In this case, events that trigger IPN messages are not directly related to actions on your website. The following diagram shows the steps your listener must take:

PayPal-initiated action

The numbers in diagram correspond to the following steps, which implement the IPN message authentication protocol:

1. PayPal sends your IPN listener a message that notifies you of the event

2. Your listener sends the complete unaltered message back to PayPal; the message must contain the same fields in the same order and be encoded in the same way as the original message

3. PayPal sends a single word back, which is either VERIFIED if the message originated with PayPal or INVALID if there is any discrepancy with what was originally sent

After successfully completing the protocol, your back-office or administrative process vets the contents of the message and responds appropriately. For example, the IPN messages may trigger you to print shipping labels for items whose payments have cleared, investigate disputes and chargebacks, store information in an accounting system, and so on.

A Sample IPN Message and Response

An IPN message consists of variables that describe the transaction. These variables contain information about you, your customer, and the details of the transaction itself.

PayPal sends a message, similar to the following one, for a $19.95 purchase made by Express Checkout:

```
mc_gross=19.95&protection_eligibility=Eligible&address_status=confirmed&pay
er_id=LPLWNMTBWMFAY&tax=0.00&address_street=1+Main+St&payment_date=20%3A12%
3A59+Jan+13%2C+2009+PST&payment_status=Completed&charset=windows-
1252&address_zip=95131&first_name=Test&mc_fee=0.88&address_country_code=US&
address_name=Test+User&notify_version=2.6&custom=&payer_status=verified&add
ress_country=United+States&address_city=San+Jose&quantity=1&verify_sign=Atk
OfCXbDm2hu0ZELryHFjY-Vb7PAUvS6nMXgysbElEn9v-
1XcmSoGtf&payer_email=gpmac_1231902590_per%40paypal.com&txn_id=61E67681CH32
38416&payment_type=instant&last_name=User&address_state=CA&receiver_email=g
pmac_1231902686_biz%40paypal.com&payment_fee=0.88&receiver_id=S8XGHLYDW9T3S
&txn_type=express_checkout&item_name=&mc_currency=USD&item_number=&residenc
e_country=US&test_ipn=1&handling_amount=0.00&transaction_subject=&payment_g
ross=19.95&shipping=0.00
```

Variable	Notes
Information about you:	
receiver_email = gm_1231902686_biz@paypal.com	Check email address to make sure that this is not a spoof
receiver_id = S8XGHLYDW9T3S	
residence_country = US	
Information about the transaction:	
test_ipn = 1	Testing with the Sandbox
transaction_subject =	
txn_id = 61E67681CH3238416	Keep this ID to avoid processing the transaction twice
txn_type = express_checkout	Type of transaction
Information about your buyer:	
payer_email = gm_1231902590_per@paypal.com	
payer_id = LPLWNMTBWMFAY	
payer_status = verified	
first_name = Test	
last_name = User	
address_city = San Jose	
address_country = United States	
address_country_code = US	
address_name = Test User	
address_state = CA	
address_status = confirmed	

Variable	Notes
address_street = 1 Main St	
address_zip = 95131	
Information about the payment:	
custom =	Your custom field
handling_amount = 0.00	
item_name =	
item_number =	
mc_currency = USD	
mc_fee = 0.88	
mc_gross = 19.95	
payment_date = 20:12:59 Jan 13, 2009 PST	
payment_fee = 0.88	
payment_gross = 19.95	
payment_status = Completed	Status, which determines whether the transaction is complete
payment_type = instant	Kind of payment
protection_eligibility = Eligible	
quantity = 1	
shipping = 0.00	
tax = 0.00	
Other information about the transaction:	
notify_version = 2.6	IPN version; can be ignored
charset = windows-1252	
verify_sign = AtkOfCXbDm2hu0ZELryHFjY-Vb7PAUvS6nMXgysbElEn9v-1XcmSoGtf	

Before you can trust the contents of the message, you must first verify that the message came from PayPal. To verify the message, you must send back the contents in the exact order they were received and precede it with the command `_notify-validate`, as follows:

```
https://www.sandbox.paypal.com/cgi-bin/webscr?cmd=_notify-
validate&mc_gross=19.95&protection_eligibility=Eligible&address_status=conf
irmed&payer_id=LPLWNMTBWMFAY&tax=0.00&...&payment_gross=19.95&shipping=0.00
```

PayPal will then send one single-word message, `VERIFIED`, if the message is valid; otherwise, it will send another single-word message, `INVALID`.

IMPORTANT: After you receive the `VERIFIED` message, there are several important checks you must perform before you can assume that the message is legitimate and not already processed:

Confirm that the payment status is `Completed`.

Use the transaction ID to verify that the transaction has not already been processed, which prevents duplicate transactions from being processed.

Validate that the receiver's email address is registered to you.

Verify that the price, item description, and so on, match the transaction on your website.

Non-IPN Notification Mechanisms

You can use IPN with other notification mechanisms. For example, you can use PDT or the API to determine real-time information about a transaction and let IPN notify you of any changes after the transaction occurs.

If you are using Website Payments Standard, you can use PDT to obtain information about the transaction. If you are using Express Checkout or Direct Payment, the PayPal API notifies you of the status and details of the transaction immediately and automatically. In either case, you can immediately display to your customer the information being returned from PayPal. You should not use IPN for this purpose.

2 Implementing an IPN Listener

You write your IPN listener in the scripting or programming language of your choice and host it on your web server. You can use sample code provided by PayPal as a starting point.

The PayPal SDKs for Website Payments Standard contain sample code in various programming languages that you can modify to create your own listener. As a good programming practice, as well as to keep things simple, your IPN listener should listen for a post from PayPal and dispatch it immediately to another routine or process that handles the business logic associated with the message. If your listener is structured in this way, it will be a simple and tight loop that listens for a message and dispatches it for processing by your application logic.

Your listener software must

1. Wait for an HTTP post from PayPal.

2. Create a request that contains exactly the same IPN variables and values in the same order, preceded with `cmd=_notify-validate`.

3. Post the request to `www.paypal.com` or `www.sandbox.paypal.com`, depending on whether you are going live or testing your listener in the Sandbox.

4. Wait for a response from PayPal, which is either `VERIFIED` or `INVALID`.

5. Verify that the response status is 200.

6. If the response is `VERIFIED`, perform the following checks:

   – Confirm that the payment status is `Completed`.

   PayPal sends IPN messages for pending and denied payments as well; do not ship until the payment has cleared.

   – Use the transaction ID to verify that the transaction has not already been processed, which prevents duplicate transactions from being processed.

   Typically, you store transaction IDs in a database so that you know you are only processing unique transactions.

   – Validate that the receiver's email address is registered to you.

   This check provides additional protection against fraud.

   – Verify that the price, item description, and so on, match the transaction on your website.

   This check provides additional protection against fraud.

7. If the verified response passes the checks, take action based on the value of the `txn_type` variable if it exists; otherwise, take action based on the value of the `reason_code` variable.

8. If the response is `INVALID` or the response code is not 200, save the message for further investigation.

A sample test listener using PHP

The following listener sends email to the address specified in the `ipn_email` variable, as in `https://`*`your_host`*`/live_ipn_mail.php?ipn_email=`*`email_address`*. You can use this listener as a starting point for your own listener; rather than send email, your listener could take action based on the type of transaction.

```php
<?php
error_reporting(E_ALL ^ E_NOTICE);
$email = $_GET['ipn_email'];
$header = "";
$emailtext = "";
// Read the post from PayPal and add 'cmd'
$req = 'cmd=_notify-validate';
if(function_exists('get_magic_quotes_gpc'))
{
    $get_magic_quotes_exits = true;
}
foreach ($_POST as $key => $value)
// Handle escape characters, which depends on setting of magic quotes
{
    if($get_magic_quotes_exists == true && get_magic_quotes_gpc() == 1){
        $value = urlencode(stripslashes($value));
    } else {
        $value = urlencode($value);
    }
    $req .= "&$key=$value";
}
// Post back to PayPal to validate
$header .= "POST /cgi-bin/webscr HTTP/1.0\r\n";
$header .= "Content-Type: application/x-www-form-urlencoded\r\n";
$header .= "Content-Length: " . strlen($req) . "\r\n\r\n";
$fp = fsockopen ('ssl://www.paypal.com', 443, $errno, $errstr, 30);


// Process validation from PayPal
// TODO: This sample does not test the HTTP response code. All
// HTTP response codes must be handles or you should use an HTTP
// library, such as cUrl

if (!$fp) { // HTTP ERROR
} else {
// NO HTTP ERROR
fputs ($fp, $header . $req);
while (!feof($fp)) {
    $res = fgets ($fp, 1024);
    if (strcmp ($res, "VERIFIED") == 0) {
        // TODO:
        // Check the payment_status is Completed
        // Check that txn_id has not been previously processed
        // Check that receiver_email is your Primary PayPal email
        // Check that payment_amount/payment_currency are correct
        // Process payment
        // If 'VERIFIED', send an email of IPN variables and values to the
        // specified email address
        foreach ($_POST as $key => $value){
        $emailtext .= $key . " = " .$value ."\n\n";
```

```
            }
            mail($email, "Live-VERIFIED IPN", $emailtext . "\n\n" . $req);
        } else if (strcmp ($res, "INVALID") == 0) {
            // If 'INVALID', send an email. TODO: Log for manual investigation.
            foreach ($_POST as $key => $value){
            $emailtext .= $key . " = " .$value ."\n\n";
            }
            mail($email, "Live-INVALID IPN", $emailtext . "\n\n" . $req);
        }
    }
    fclose ($fp);
?>
```

3 Identifying Your IPN Listener to PayPal

After you implement and test your IPN listener, you make your listener known to PayPal by specifying the listener's URL in your account's profile. Optionally, you can override the URL to specify another listener for specific payments.

- Setting Up IPN Notifications on PayPal
- Dynamically Setting the Notification URL

Setting Up IPN Notifications on PayPal

After you implement and test your IPN listener, you identify the listener to PayPal by selecting **Instant Payment Notification Preferences** from your account's profile. You then specify your listener's URL and click the box to activate it.

Your listener must be located at the URL that you specify in the profile. Before you activate your listener on PayPal, you should test the listener using the IPN simulator in the Sandbox.

IMPORTANT: IPN messages for Adaptive Payments do not use the IPN message handler that you can specify in the Profile. IPN messages are only sent to the notification URL specified in the Adaptive Payment API operation's request.

The steps to set up your listener for Sandbox testing and to set up your listener for live operation on PayPal are the same. The only difference is that you log into the Sandbox to set up your listener for Sandbox testing and you log into PayPal to set up your listener for live operation.

NOTE: Sandbox testing goes beyond simply using the IPN simulator. The IPN simulator only sends IPN messages to a listener; it does not perform the complete operation; for example, it does not complete a transaction. Sandbox testing involves performing the same actions in the Sandbox as you expect to be performed when the listener goes live.

After you log in, follow these instructions to set up your listener:

1. Click **Profile** on the **My Account** tab.

2. Click **Instant Payment Notification Preferences** in the Selling Preferences column.

3. Click **Choose IPN Settings** to specify your listener's URL and activate the listener.

   The following screen appears:

4. Specify the URL for your listener in the Notification URL field.

5. Click Receive IPN messages (Enabled) to enable your listener.

6. Click **Save**.

The following screen appears:

7. Click **Back to Profile Summary** to return to the Profile after activating your listener.

You also can click **Edit settings** to modify your notification URL or disable your listener. You can click **Turn Off IPN** to reset your IPN preferences.

Dynamically Setting the Notification URL

You can specify an IPN listener for a specific payment; this is the only way to receive IPNs associated with Adaptive Payments. In this case, PayPal sends the IPN message to the listener specified in the notification URL for a specific button or API operation instead of the listener specified in your Profile.

To specify a notification URL

For a/an ...	specify your IPN Listener's URL in the ...
Website Payments Standard button	`notify_url` HTML form variable
NVP API operation	`NOTIFYURL` field of the `DoDirectPayment`, `DoExpressCheckoutPayment`, or `DoReferenceTransaction` request
SOAP API operation	`NotifyURL` field of the `DoDirectPayment`, `DoExpressCheckoutPayment`, or `DoReferenceTransaction` request
Adaptive Payments API operation	`ipnNotificationUrl` field of the `Pay` or `Preapproval` request

NOTE: The IPN message will always be sent to your notification URL unless receiving IPNs have been disabled. Even though you have not enabled receiving IPN messages in your Profile or you have reset your preference by turning off IPN messages, PayPal still sends IPN messages to the notification URL you specify for a specific payment.

IPN messages not sent because you disabled the preference in your Profile will appear in the IPN history when you enable receiving IPNs. After they appear in the history, you can choose whether or not to resend them.

4 IPN Testing

After you implement your listener and start it running on your web server, you can use the IPN simulator in the Sandbox to send IPN messages to the URL at which your listener is running. This tool allows you to verify that you are receiving IPN messages correctly.

- Testing Your Listener
- IPN Troubleshooting Tips

Testing Your Listener

The first level of testing is to ensure that your IPN listener receives messages and handles them appropriately. This level of testing requires you to have your IPN listener running at your notification URL; however, it does not require you to set up the listener in the Profile.

You must be logged into the Sandbox to use the IPN simulator. Not all API operations are available using the IPN simulator.

To set up and send an IPN message using the simulator:

1. Select **Instant Payment Notification (IPN) simulator** from **Test Tools**.

2. Enter the URL to receive the notification and the kind of notification you want to test on the following screen:

When you select the kind of transaction that you want to test, a form containing test data appears:

3. Keep or modify the values of fields that you want to include in the IPN.

By default, only populated fields are displayed. You can check the **Show all fields** box to view all fields. The simulator does not check the validity of fields that you change.

4. Click **Send IPN**.

The IPN message is sent to the specified URL and the results of the operation are displayed at the top of the page.

After Completing This Task:

If your IPN listener receives a message, you know that it is properly installed on your web server. The default messages sent by the IPN simulator are valid, thus, if your listener responds correctly to the message, it should receive a `VERIFIED` message. If you do not receive any message or if you receive an `INVALID` message after responding to the original message from PayPal, you will need to troubleshoot your listener.

IPN Troubleshooting Tips

IPN failures fall into three categories: not receiving any IPN messages from PayPal, receiving some but not all IPN messages, and receiving `INVALID` messages from PayPal after responding to a message.

If you do not receive any IPN messages from PayPal

- Check the IPN History page on PayPal. It tells you whether PayPal sent the IPN message and whether your listener responded to it. It may also provide information about the status of the server on which your listener is running. If necessary, from this page you can also request that PayPal resend the IPN message.

- Check that the path to your IPN listener is correct and you are using that path correctly in your IPN notification URL; for example, the file path is often similar to, but not the same as, the URL.

- Verify that your firewall settings are not blocking HTTP POST messages from PayPal.

- If the logs for your web server are available, check the logs to confirm that messages are being sent to your web server and check for any errors that may have occurred.

If you receive some messages but not others

- Verify that your IPN listener is responding to all messages, even those you do not intend to process.

- Check that the account is valid and confirmed; for example, if you send money to an unconfimred account, PayPal does not send an IPN message.

If you receive an `INVALID` message

- Check that your are posting your response to the correct URL, which is `https://www.sandbox.paypal.com/cgi-bin/webscr` or `https://www.paypal.com/cgi-bin/webscr`, depending on whether you are testing in the Sandbox or you are live, respectively.

- Verify that your response contains exactly the same IPN variables and values in the same order, preceded with `cmd=_notify-validate`.

- Ensure that you are encoding your response string and are using the same character encoding as the original message.

NOTE: If you receive multiple IPN messages for the same transaction or if messages appear to be out of order, this is not necessarily an indication that your listener is malfunctioning. For example, if you do not respond in time, PayPal resends the message. You should investigate these situations; however, because they could be caused by a logic errors or performance problems as well.

5

IPN Operations on PayPal

The IPN History page on PayPal provides additional information to help you troubleshoot IPN messages. You can use the IPN History page to determine the status of IPN messages and to resend them, if necessary.

- Using the IPN History
- Resending IPN Messages

Using the IPN History

Use the IPN History page on PayPal to view IPN messages sent to you from PayPal and request that messages be resent. You can select the IPN messages to review by date range, by delivery status, and by PayPal transaction

The search results contain the following information:

- The date and time that PayPal created the IPN message

- Whether this IPN message was the original message or whether it was resent, which is indicated in the **Date/time created** column

- The IPN message ID assigned by PayPal

- The current status, which is one of the following values:

    - `Sent` indicates that PayPal sent the message to your IPN listener

    - `Failed` indicates that PayPal did not receive an acknowledgement to the message

- – `Queued` indicates that PayPal is ready to send the message
  - – `Retrying` indicates that message was resent between 1 and 15 times and PayPal continues to be resend the message
  - – `Disabled` indicates that the message will not be resent because the merchant's account has been disabled

  NOTE: If you have requested that PayPal resend the IPN message and the status has not been updated for the attempt, `resending` is appended to the status, e.g. `Failed - resending`.

- Your server's response to the HTTP POST that delivered the IPN message to your listener. For more information about these codes, see [RFC 2616: Hypertext Transfer Protocol -- HTTP/1.1](#).

- If the message is related to a PayPal transaction, the ID of the PayPal transaction associated with the message; you can

To see more information about an IPN message, click on the message ID. The IPN details page appears:

In addition to the information on the IPN History page, the details contain the following information:

- Whether this IPN message was the original message or whether it was resent
- The last time the message was resent
- The URL on which your listener was running when the message was sent
- The number of retries before the message was successfully acknowledged
- The type of IPN message

Resending IPN Messages

You can use the IPN History page to request that PayPal resend one or more IPN messages. You can search the IPN message history for the messages that may need to be resent and then select them.

To make a request that PayPal resend IPN messages, mark one or more messages for PayPal to resend and click the **Resend selected** button to make the request:

If you check the box for **To send an IPN again to a profile URL**..., the message will be resent to the current profile URL instead of the URL associated with the original message. If you do

not check this option, PayPal sends the message to the same URL to which the original message was sent, which is not necessarily the current profile URL.

You may receive a confirmation notice alerting you to the possibility that you will receive duplicate messages. Click **Confirm** to resend; otherwise click **Cancel**:
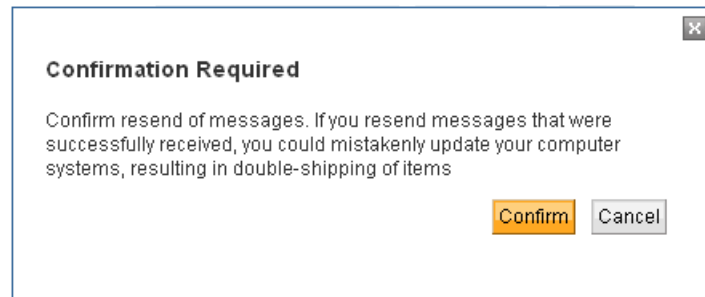
After you make the request, PayPal notifies you that the messages have been resent and updates the status. The status indicates that PayPal is resending the message until it actually has been resent:

When the message has been sent, your server's response to the HTTP POST is used to update the HTTP response code field. A value of 200 indicates that your server successfully received the IPN message. Other values typically indicate a server configuration error for the server that hosts your IPN listener. For more information about these codes, see RFC 2616: Hypertext Transfer Protocol -- HTTP/1.1. If you do not see a response code, you should check that your sever is running.

6

Using Fraud Management Filters With IPN

Fraud Management Filter actions are reported in IPN payment messages only when a filter causes the payment to be pended awaiting your review or a when you accept or deny a filter-pended payment. Filter actions are not reported when filters flag payments for review, allow payments to be accepted, or cause them to be denied.

When a payment occurs, an IPN message shows the transaction's payment status as `Completed`, regardless of whether a Fraud Management Filter was activated or not. There is no special notification for transactions that are flagged by a Fraud Management Filter. If a Fraud Management Filter is set to `Deny`, PayPal does not send an IPN message when the filter actually causes the payment to be denied.

When a transaction is pended, however, PayPal sends an IPN message containing one or more `fraud_management_pending_filters_`*n* variables, which identify the filters that caused the payment to be pended, where `n=1` specifies the first filter, and so on. In addition, the `payment_status` variable is set to `Pending`. The following example shows an IPN message in which two filters cause the transaction to be pended:

```
txn_type = virtual_terminal
payment_date = 17:11:42 Jul 15, 2008 PDT
last_name =
receipt_id = 3075-7371-4622-1677
residence_country = US
pending_reason = address
item_name =
payment_gross = 3.33
mc_currency = USD
business = acqrte_1215804264_biz@gmail.com
payment_type = instant
verify_sign = APYUGJhXGkUmvFnZf4I5co6CedKKAowZjfT4T7GXWJMDnZ0uFLkcq.oH
payer_status = unverified
test_ipn = 1
fraud_management_pending_filters_1 = Maximum Transaction Amount
tax = 0.00
txn_id = 5XN64179EB804362B
fraud_management_pending_filters_2 = Unconfirmed Address
quantity = 1
first_name =
receiver_email = acqrte_1215804264_biz@gmail.com
payer_id = PUWAJRBB8NM74
receiver_id = 2RXLTRMGT3M2G
item_number =
payment_status = Pending
shipping = 0.00
mc_gross = 3.33
custom =
charset = windows-1252
notify_version = 2.4
```

NOTE: If the transaction is for an authorization or an order, the `auth_status` variable may also be set to `Pending`.

If a transaction has been pended, PayPal sends an IPN message when the payment has been accepted or denied. The following example shows an IPN message indicating that a pended transaction has been accepted:

```
txn_type = virtual_terminal
payment_date = 17:11:42 Jul 15, 2008 PDT
last_name =
receipt_id = 3075-7371-4622-1677
residence_country = US
item_name =
payment_gross = 3.33
mc_currency = USD
business = acqrte_1215804264_biz@gmail.com
payment_type = instant
verify_sign = AFcWxV21C7fd0v3bYYYRCpSSRl31AjcbYkD.VCCBmpD4lZq.yYTxBKkr
payer_status = unverified
test_ipn = 1
fraud_management_pending_filters_1 = Maximum Transaction Amount
tax = 0.00
txn_id = 5XN64179EB804362B
fraud_management_pending_filters_2 = Unconfirmed Address
quantity = 1
receiver_email = acqrte_1215804264_biz@gmail.com
first_name =
payer_id = PUWAJRBB8NM74
receiver_id = 2RXLTRMGT3M2G
item_number =
payment_status = Completed
payment_fee = 0.45
mc_fee = 0.45
shipping = 0.00
mc_gross = 3.33
custom =
charset = windows-1252
notify_version = 2.4
```

The following example shows an IPN message indicating that a pended transaction has been denied:

```
txn_type = virtual_terminal
payment_date = 17:09:40 Jul 15, 2008 PDT
last_name =
receipt_id = 0739-3836-3393-2098
residence_country = US
item_name =
payment_gross = 2.11
mc_currency = USD
business = acqrte_1215804264_biz@gmail.com
payment_type = instant
verify_sign = AFcWxV21C7fd0v3bYYYRCpSSRl31ASrKFBPwac7aQm47p8CMLrdParSt
payer_status = unverified
test_ipn = 1
fraud_management_pending_filters_1 = Maximum Transaction Amount
tax = 0.00
txn_id = 53R82724RM1848354
fraud_management_pending_filters_2 = Unconfirmed Address
quantity = 1
first_name =
receiver_email = acqrte_1215804264_biz@gmail.com
payer_id = PUWAJRBB8NM74
receiver_id = 2RXLTRMGT3M2G
item_number =
payment_status = Denied
shipping = 0.00
mc_gross = 2.11
custom =
charset = windows-1252
notify_version = 2.4
```

7 IPN Variable Reference

PayPal returns related variables for each kind of IPN message. Not all variables are returned for each type of transaction.

- IPN Transaction Types
- Transaction and Notification-Related Variables
- Buyer Information Variables
- Payment Information Variables
- Auction Variables
- Mass Pay Variables
- Recurring Payments Variables
- Subscription Variables
- Pay Message Variables
- Preapproval Message Variables
- Adaptive Accounts IPN Messages
- Dispute Resolution Variables

IPN Transaction Types

Typically, your back-end or administrative processes will perform specific actions based on the kind of IPN message received. You can use the `txn_type` variable in the message to trigger the kind of processing you want to perform.

Transaction Type (txn_type)	Description
—	Credit card chargeback if the `case_type` variable contains `chargeback`
adjustment	A dispute has been resolved and closed
cart	Payment received for multiple items; source is Express Checkout or the PayPal Shopping Cart.
express_checkout	Payment received for a single item; source is Express Checkout
masspay	Payment sent using MassPay
merch_pmt	Monthly subscription paid for Website Payments Pro
new_case	A new dispute was filed

Transaction Type (txn_type)	Description
recurring_payment	Recurring payment received
recurring_payment _profile_created	Recurring payment profile created
send_money	Payment received; source is the **Send Money** tab on the PayPal website
subscr_cancel	Subscription canceled
subscr_eot	Subscription expired
subscr_failed	Subscription signup failed
subscr_modify	Subscription modified
subscr_payment	Subscription payment received
subscr_signup	Subscription started
virtual_terminal	Payment received; source is Virtual Terminal
web_accept	Payment received; source is a **Buy Now**, **Donation**, or **Auction Smart Logos** button

Transaction and Notification-Related Variables

Transaction and notification-related variables identify the merchant that is receiving a payment or other notification and transaction-specific information.

Variable Name	Description
business	Email address or account ID of the payment recipient (that is, the merchant). Equivalent to the values of receiver_email (if payment is sent to primary account) and business set in the Website Payment HTML. **NOTE:** The value of this variable is normalized to lowercase characters. Length: 127 characters
charset	Character set
custom	Custom value as passed by you, the merchant. These are pass-through variables that are never presented to your customer Length: 255 characters
notify_version	Message's version number
parent_txn_id	In the case of a refund, reversal, or canceled reversal, this variable contains the txn_id of the original transaction, while txn_id contains a new ID for the new transaction. Length: 19 characters

Variable Name	Description
receipt_id	Unique ID generated during guest checkout (payment by credit card without logging in).
receiver_email	Primary email address of the payment recipient (that is, the merchant). If the payment is sent to a non-primary email address on your PayPal account, the receiver_email is still your primary email. **NOTE:** The value of this variable is normalized to lowercase characters. Length: 127 characters
receiver_id	Unique account ID of the payment recipient (i.e., the merchant). This is the same as the recipient's referral ID. Length: 13 characters
resend	Whether this IPN message was resent (equals true); otherwise, this is the original message.
residence_country	ISO 3166 country code associated with the country of residence Length: 2 characters
test_ipn	Whether the message is a test message. It is one of the following values: • 1 – the message is directed to the Sandbox
txn_id	The merchant's original transaction identification number for the payment from the buyer, against which the case was registered.
txn_type	The kind of transaction for which the IPN message was sent.
verify_sign	Encrypted string used to validate the authenticity of the transaction

Buyer Information Variables

Buyer information identifies the buyer or initiator of a transaction by payer ID or email address. Additional contact or shipping information may be provided.

Variable Name	Description
address_country	Country of customer's address Length: 64 characters
address_city	City of customer's address Length: 40 characters
address_country_code	ISO 3166 country code associated with customer's address Length: 2 characters
address_name	Name used with address (included when the customer provides a Gift Address) Length: 128 characters

Variable Name	Description
address_state	State of customer's address Length: 40 characters
address_status	Whether the customer provided a confirmed address. It is one of the following values: ● confirmed – Customer provided a confirmed address. ● unconfirmed – Customer provided an unconfirmed address.
address_street	Customer's street address. Length: 200 characters
address_zip	Zip code of customer's address. Length: 20 characters
contact_phone	Customer's telephone number. Length: 20 characters
first_name	Customer's first name Length: 64 characters
last_name	Customer's last name Length: 64 characters
payer_business_name	Customer's company name, if customer is a business Length: 127 characters
payer_email	Customer's primary email address. Use this email to provide any credits. Length: 127 characters
payer_id	Unique customer ID. Length: 13 characters

Payment Information Variables

Payment information identifies the amount and status of a payment transaction, including fees.

Variable Name	Description
auth_amount	Authorization amount
auth_exp	Authorization expiration date and time, in the following format: HH:MM:SS DD Mmm YY, YYYY PST Length: 28 characters
auth_id	Authorization identification number Length: 19 characters
auth_status	Status of authorization

Variable Name	Description
exchange_rate	Exchange rate used if a currency conversion occurred.
fraud_managment_pending_filters_*x*	One or more filters that identify a triggering action associated with one of the following payment_status values: Pending, Completed, Denied, where *x* is a number starting with 1 that makes the IPN variable name unique; *x* is not the filter's ID number. The filters and their ID numbers are as follows:
invoice	Passthrough variable you can use to identify your Invoice Number for this purchase. If omitted, no variable is passed back. Length: 127 characters
item_name*x*	Item name as passed by you, the merchant. Or, if not passed by you, as entered by your customer. If this is a shopping cart transaction, PayPal will append the number of the item (e.g., item_name1, item_name2, and so forth). Length: 127 characters
item_number*x*	Pass-through variable for you to track purchases. It will get passed back to you at the completion of the payment. If omitted, no variable will be passed back to you. If this is a shopping cart transaction, PayPal will append the number of the item (e.g., item_number1, item_number2, and so forth) Length: 127 characters
mc_currency	• For payment IPN notifications, this is the currency of the payment. • For non-payment subscription IPN notifications (i.e., txn_type= signup, cancel, failed, eot, or modify), this is the currency of the subscription. • For payment subscription IPN notifications, it is the currency of the payment (i.e., txn_type = subscr_payment)
mc_fee	Transaction fee associated with the payment. mc_gross minus mc_fee equals the amount deposited into the receiver_email account. Equivalent to payment_fee for USD payments. If this amount is negative, it signifies a refund or reversal, and either of those payment statuses can be for the full or partial amount of the original transaction fee.
mc_gross	Full amount of the customer's payment, before transaction fee is subtracted. Equivalent to payment_gross for USD payments. If this amount is negative, it signifies a refund or reversal, and either of those payment statuses can be for the full or partial amount of the original transaction.
mc_gross_*x*	The amount is in the currency of mc_currency, where *x* is the shopping cart detail item number. The sum of mc_gross_*x* should total mc_gross.
mc_handling	Total handling amount associated with the transaction.
mc_shipping	Total shipping amount associated with the transaction.

Variable Name	Description
mc_shipping*x*	This is the combined total of shipping1 and shipping2 Website Payments Standard variables, where *x* is the shopping cart detail item number. The shipping*x* variable is only shown when the merchant applies a shipping amount for a specific item. Because profile shipping might apply, the sum of shipping*x* might not be equal to shipping.
memo	Memo as entered by your customer in PayPal Website Payments note field. Length: 255 characters
num_cart_items	If this is a PayPal Shopping Cart transaction, number of items in cart.
option_name1	Option 1 name as requested by you. PayPal appends the number of the item where *x* represents the number of the shopping cart detail item (e.g., option_name1, option_name2). Length: 64 characters
option_name2	Option 2 name as requested by you. PayPal appends the number of the item where *x* represents the number of the shopping cart detail item (e.g., option_name2, option_name2). Length: 64 characters
option_selection1	Option 1 choice as entered by your customer. PayPal appends the number of the item where *x* represents the number of the shopping cart detail item (e.g., option_selection1, option_selection2). Length: 200 characters
option_selection2	Option 2 choice as entered by your customer. PayPal appends the number of the item where *x* represents the number of the shopping cart detail item (e.g., option_selection1, option_selection2). Length: 200 characters
payer_status	Whether the customer has a verified PayPal account. ● verified – Customer has a verified PayPal account. ● unverified – Customer has an unverified PayPal account.
payment_date	Time/Date stamp generated by PayPal, in the following format: HH:MM:SS DD Mmm YY, YYYY PST Length: 28 characters
payment_fee	USD transaction fee associated with the payment. payment_gross minus payment_fee equals the amount deposited into the receiver email account. Is empty for non-USD payments. If this amount is negative, it signifies a refund or reversal, and either of those payment statuses can be for the full or partial amount of the original transaction fee. **NOTE:** This is a deprecated field. Use mc_fee instead.

Variable Name	Description
payment_fee *x*	If the payment is USD, then the value is the same as that for mc_fee *x*, where *x* is the record number; if the currency is not USD, then this is an empty string. **NOTE:** This is a deprecated field. Use mc_fee *x* instead.
payment_gross	Full USD amount of the customer's payment, before transaction fee is subtracted. Will be empty for non-USD payments. This is a legacy field replaced by mc_gross. If this amount is negative, it signifies a refund or reversal, and either of those payment statuses can be for the full or partial amount of the original transaction.
payment_gross *x*	If the payment is USD, then the value for this is the same as that for the mc_gross *x*, where *x* is the record number the mass pay item. If the currency is not USD, this is an empty string. **NOTE:** This is a deprecated field. Use mc_gross *x* instead.
payment_status	The status of the payment: Canceled_Reversal: A reversal has been canceled. For example, you won a dispute with the customer, and the funds for the transaction that was reversed have been returned to you. Completed: The payment has been completed, and the funds have been added successfully to your account balance. Created: A German ELV payment is made using Express Checkout. Denied: You denied the payment. This happens only if the payment was previously pending because of possible reasons described for the pending_reason variable or the Fraud_Management_Filters_*x* variable. Expired: This authorization has expired and cannot be captured. Failed: The payment has failed. This happens only if the payment was made from your customer's bank account. Pending: The payment is pending. See pending_reason for more information. Refunded: You refunded the payment. Reversed: A payment was reversed due to a chargeback or other type of reversal. The funds have been removed from your account balance and returned to the buyer. The reason for the reversal is specified in the ReasonCode element. Processed: A payment has been accepted. Voided: This authorization has been voided.
payment_type	echeck: This payment was funded with an eCheck. instant: This payment was funded with PayPal balance, credit card, or Instant Transfer.

Variable Name	Description
pending_reason	This variable is set only if payment_status = Pending.
	address: The payment is pending because your customer did not include a confirmed shipping address and your Payment Receiving Preferences is set yo allow you to manually accept or deny each of these payments. To change your preference, go to the **Preferences** section of your **Profile**.
	authorization: You set the payment action to Authorization and have not yet captured funds.
	echeck: The payment is pending because it was made by an eCheck that has not yet cleared.
	intl: The payment is pending because you hold a non-U.S. account and do not have a withdrawal mechanism. You must manually accept or deny this payment from your **Account Overview**.
	multi-currency: You do not have a balance in the currency sent, and you do not have your **Payment Receiving Preferences** set to automatically convert and accept this payment. You must manually accept or deny this payment.
	order: You set the payment action to Order and have not yet captured funds.
	paymentreview: The payment is pending while it is being reviewed by PayPal for risk.
	unilateral: The payment is pending because it was made to an email address that is not yet registered or confirmed.
	upgrade: The payment is pending because it was made via credit card and you must upgrade your account to Business or Premier status in order to receive the funds. upgrade can also mean that you have reached the monthly limit for transactions on your account.
	verify: The payment is pending because you are not yet verified. You must verify your account before you can accept this payment.
	other: The payment is pending for a reason other than those listed above. For more information, contact PayPal Customer Service.
protection_eligibility	ExpandedSellerProtection: Seller is protected by Expanded seller protection
	SellerProtection: Seller is protected by PayPal's Seller Protection Policy
	None: Seller is not protected under Expanded seller protection nor the Seller Protection Policy
quantity	Quantity as entered by your customer or as passed by you, the merchant. If this is a shopping cart transaction, PayPal appends the number of the item (e.g. quantity1, quantity2).

Variable Name	Description
reason_code	This variable is set if payment_status =Reversed, Refunded, or Canceled_Reversal. adjustment_reversal: Reversal of an adjustment buyer-complaint: A reversal has occurred on this transaction due to a complaint about the transaction from your customer. chargeback: A reversal has occurred on this transaction due to a chargeback by your customer. chargeback_reimbursement: Reimbursement for a chargeback chargeback_settlement: Settlement of a chargeback guarantee: A reversal has occurred on this transaction due to your customer triggering a money-back guarantee. other: Non-specified reason. refund: A reversal has occurred on this transaction because you have given the customer a refund. **NOTE:** Additional codes may be returned.
remaining_settle	Remaining amount that can be captured with Authorization and Capture
settle_amount	Amount that is deposited into the account's primary balance after a currency conversion from automatic conversion (through your Payment Receiving Preferences) or manual conversion (through manually accepting a payment).
settle_currency	Currency of settle_amount.
shipping	Shipping charges associated with this transaction. Format: unsigned, no currency symbol, two decimal places.
shipping_method	The name of a shipping method from the Shipping Calculations section of the merchant's account profile. The buyer selected the named shipping method for this transaction.
tax	Amount of tax charged on payment. PayPal appends the number of the item (e.g., item_name1, item_name2). The tax$x$ variable is included only if there was a specific tax amount applied to a particular shopping cart item. Because total tax may apply to other items in the cart, the sum of tax$x$ might not total to tax.
transaction_entity	Authorization and Capture transaction entity

Auction Variables

Auction information identifies the auction for which a payment is made and additional information about the auction.

Variable Name	Description
`auction_buyer_id`	The customer's auction ID. Length: 64 characters
`auction_closing_date`	The auction's close date, in the following format: HH:MM:SS DD Mmm YY, YYYY PST Length: 28 characters
`auction_multi_item`	The number of items purchased in multi-item auction payments. It allows you to count the `mc_gross` or `payment_gross` for the first IPN you receive from a multi-item auction (`auction_multi_item`), since each item from the auction will generate an Instant Payment Notification showing the amount for the entire auction.
`for_auction`	This is an auction payment—payments made using Pay for eBay Items or Smart Logos—as well as Send Money/Money Request payments with the type **eBay items** or **Auction Goods (non-eBay)**.

Mass Pay Variables

Mass pay information identifies the amounts and status of transactions related to mass payments, including fees.

Variable Name	Description
`masspay_txn_id_x`	For Mass Payments, a unique transaction ID generated by the PayPal system, where *x* is the record number of the mass pay item Length: 19 characters
`mc_currency_x`	For Mass Payments, the currency of the amount and fee, where *x* is the record number the mass pay item
`mc_fee_x`	For Mass Payments, the transaction fee associated with the payment, where *x* is the record number the mass pay item
`mc_gross_x`	The gross amount for the amount, where *x* is the record number the mass pay item
`mc_handlingx`	The *x* is the shopping cart detail item number. The `handling_cart` cart-wide Website Payments variable is also included in the `mc_handling` variable; for this reason, the sum of `mc_handlingx` might not be equal to `mc_handling`
`payment_date`	For Mass Payments, the first IPN is the date/time when the record set is processed and the second IPN is the date/time when all payments are completed/returned. Format: HH:MM:SS DD Mmm YY, YYYY PST Length: 28 characters

Variable Name	Description
payment_status	`Completed`: For Mass Payments, this means that all of your payments have been claimed, or after a period of 30 days, unclaimed payments have been returned to you. `Denied`: For Mass Payments, this means that your funds were not sent and the Mass Payment was not initiated. This may have been caused by lack of funds. `Processed`: Your Mass Payment has been processed and all payments have been sent.
reason_code	This variable is only set if `status = Failed`. `1001`: Invalid UserID. `1003`: Country of Residence check failure `1004`: Country of Funding Source check failure
receiver_email *x*	For Mass Payments, the primary email address of the payment recipient, where *x* is the record number of the mass pay item. Length: 127 characters
status *x*	For Mass Payments, the status of the payment, where *x* is the record number `Completed`: The payment has been processed, regardless of whether this was originally a unilateral payment `Failed`: The payment failed because of insufficient PayPal balance. `Returned`: Payment has been returned after 30 days. `Reversed`: This is for unilateral payments that were not claimed after 30 days and have been returned to the sender. Or the funds have been returned because the Receiver's account was locked. `Unclaimed`: This is for unilateral payments that are unclaimed.
unique_id *x*	For Mass Payments, the unique ID from input, where *x* is the record number. This allows the merchant to cross-reference the payment Length: 13 characters

Recurring Payments Variables

Recurring payments information identifies the amounts and status associated with recurring payments transactions.

Variable Name	Description
amount	Amount of recurring payment
amount_per_cycle	Amount of recurring payment per cycle
initial_payment_amount	Initial payment amount for recurring payments
next_payment_date	Next payment date for a recurring payment

Variable Name	Description
outstanding_balance	Outstanding balance for recurring payments
payment_cycle	Payment cycle for recurring payments
period_type	Kind of period for a recurring payment
product_name	Product name associated with a recurring payment
product_type	Product name associated with a recurring payment
profile_status	Profile status for a recurring payment
recurring_payment_id	Recurring payment ID
rp_invoice_id	The merchant's own unique reference or invoice number, which can be used to uniquely identify a profile. Length: 127 single-byte alphanumeric characters
time_created	When a recurrng payment was created

Summary of recurring payment variables

Variables	Profile created message	Recurring payment message
Basic Information		
business		X
receiver_email	X	X
receiver_id		X
Transaction Information		
payment_status		X
payment_type		X
payment_date		X
txn_id		X
initial_payment_status	X	
initail_payment_txn_id		X
txn_type	recurring_payment_profile_ created	recurring_payment
Currency and Exchange		
mc_gross		X
mc_fee		X
mc_currency		X
payment_gross		X
currency_code	X	X
payment_fee		X
Buyer Information		
first_name	X	X
last_name	X	X
address_name		X
address_street		X
address_city		X
address_state		X
address_zip		X
address_country		X
payer_email	X	X

Variables	Profile created message	Recurring payment message
payer_id	X	X
payer_status	X	X
residence_country	X	X
address_country_code		X
address_status		X
Recurring Payment		
recurring_payment_id	X	X
rp_invoice_id	X	X
product_name	X	X
product_type	X	X
period_type	X	X
payment_cycle	X	X
outstanding_balance	X	X
amount_per_cycle	X	X
initial_payment_amount	X	X
profile_status	X	X
amount	X	X
time_created	X	X
next_payment_date	X	X
Other Information		
notify_version	X	X
charset	X	X

Subscription Variables

Subscription information identifies the amounts and parameters associated with subscription transactions.

Variable Name	Description
amount1	Amount of payment for trial period 1 for USD payments; otherwise blank (optional).

Variable Name	Description
amount2	Amount of payment for trial period 2 for USD payments; otherwise blank (optional).
amount3	Amount of payment for regular subscription period for USD payments; otherwise blank.
mc_amount1	Amount of payment for trial period 1, regardless of currency (optional).
mc_amount2	Amount of payment for trial period 2, regardless of currency (optional).
mc_amount3	Amount of payment for regular subscription period, regardless of currency.
password	(optional) Password generated by PayPal and given to subscriber to access the subscription (password will be encrypted). Length: 24 characters
period1	(optional) Trial subscription interval in days, weeks, months, years (example: a 4 day interval is "period1: 4 D").
period2	(optional) Trial subscription interval in days, weeks, months, or years.
period3	Regular subscription interval in days, weeks, months, or years.
reattempt	Indicates whether reattempts should occur upon payment failures (1 is yes, blank is no).
recur_times	The number of payment installments that will occur at the regular rate.
recurring	Indicates whether regular rate recurs (1 is yes, blank is no).
retry_at	Date PayPal will retry a failed subscription payment.
subscr_date	Start date or cancellation date depending on whether transaction is subscr_signup or subscr_cancel. Time/Date stamp generated by PayPal, in the following format: HH:MM:SS DD Mmm YY, YYYY PST
subscr_effective	Date when the subscription modification will be effective (only for txn_type = subscr_modify). Time/Date stamp generated by PayPal, in the following format: HH:MM:SS DD Mmm YY, YYYY PST
subscr_id	ID generated by PayPal for the subscriber. Length: 19 characters
username	(optional) Username generated by PayPal and given to subscriber to access the subscription. Length: 64 characters

Summary of subscription variables

Variable	Signup	Cancel	Modify	USD Payment	Multi-Currency Payment	Re-fund	Failed	EOT
Basic Information								
business	X	X	X	X	X		X	X
receiver_email	X	X	X	X	X		X	X
receiver_id				X	X			
item_name	X	X	X	X	X		X	X
item_number	X	X	X	X	X		X	X
Advanced and Custom Information								
invoice	X	X	X	X	X		X	X
custom	X	X	X	X	X		X	X
option_name1	X	X	X	X	X		X	X
option_selection1	X	X	X	X	X		X	X
option_name2	X	X	X	X	X		X	X
option_selection2	X	X	X	X	X		X	X
Transaction Information								
payment_status				X	X	X		
pending_reason				X	X			
reason_code				X	X			
payment_date				X	X			
txn_id				X	X			
parent_txn_id				X	X			
txn_type	subscr_ signup	subscr_ cancel	subscr_ modify	subscr_payment			subscr_ failed	subscr_eot
Currency and Exchange information								
mc_gross				X	X			
mc_fee				X	X			
mc_currency	X	X	X	X	X		X	X
settle_amount				X	X			
exchange_rate				X	X			

Variable	Signup	Cancel	Modify	USD Payment	Multi-Currency Payment	Re-fund	Failed	EOT
payment_gross				X			X	X
payment_fee				X				
Buyer Information								
first_name	X	X	X	X	X		X	X
last_name	X	X	X	X	X		X	X
payer_business _name	X	X	X	X	X		X	
address_name	X	X	X	X	X		X	
address_street	X	X	X	X	X		X	
address_city	X	X	X	X	X		X	
address_state	X	X	X	X	X		X	
address_zip	X	X	X	X	X		X	
address_ country	X	X	X	X	X		X	
payer_email	X	X	X	X	X		X	X
payer_id	X	X	X	X	X		X	X
payer_status	X	X	X	X	X		X	X
payment_type				X	X			
Subscription Information								
subscr_date	X	X	X					
subscr_ effective			X					
period1	X	X	X					
period2	X	X	X					
period3	X	X	X					
amount1	X	X	X					
amount2	X	X	X					
amount3	X	X	X					
mc_amount1	X	X	X					
mc_amount2	X	X	X					
recurring	X	X	X					

Variable	Signup	Cancel	Modify	USD Payment	Multi-Currency Payment	Re-fund	Failed	EOT
reattempt	X	X	X					
retry_at							X	
recur_times	X	X	X					
username	X	X	X	X	X		X	X
password	X	X	X	X	X		X	X
subscr_id	X	X	X	X	X		X	X

Dispute Resolution Variables

Dispute resolution information identifies the case ID and status associated with a dispute.

Variable Name	Description
case_creation_date	Date and time case was registered, in the following format: HH:MM:SS DD Mmm YY, YYYY PST
case_id	Case identification number. Format: PP-nnn-nnn-nnn where n is any numeric character.
case_type	• chargeback: A buyer has filed a chargeback with his credit card company, which has notified PayPal of the reason for the chargeback. • complaint: A buyer has logged a complaint through the PayPal Resolution Center. • dispute: A buyer and seller post communications to one another through the Resolution Center to try to work out issues without intervention by PayPal.

Variable Name	Description
reason_code	Reason for the case. Values for case_type set to complaint: • non_receipt: Buyer claims that he did not receive goods or service. • not_as_described: Buyer claims that the goods or service received differ from merchant's description of the goods or service. Values for case_type set to chargeback: • unauthorized • adjustment_reimburse: A case that has been resolved and close requires a reimbursement. • non_receipt: Buyer claims that he did not receive goods or service. • duplicate: Buyer claims that a possible duplicate payment was made to the merchant. • merchandise: Buyer claims that the received merchandise is unsatisfactory, defective, or damaged. • special: Some other reason. Usually, special indicates a credit card processing error for which the merchant is not responsible and for which no debit to the merchant will result. PayPal must review the documentation from the credit card company to determine the nature of the dispute and possibly contact the merchant to resolve it.

Pay Message Variables

PayPal generates an IPN message that contains information about the pay request or payment in response to the Adaptive Payments Pay and ExecutePayment API operations.

Variable	Description
transaction_type	The type of transaction. Possible values are: • Adaptive Payment PAY This notification occurs when is a payment is made due to a PayRequest. The variables for the Adaptive Payment Pay notification are similar to the PaymentDetailsResponse fields. • Adjustment This can be for a chargeback, reversal, or refund; check the reason_code to see which it is.

Variable	Description
status	The status of the payment. Possible values are: • CREATED – The payment request was received; funds will be transferred once the payment is approved • COMPLETED – The payment was successful • INCOMPLETE – Some transfers succeeded and some failed for a parallel payment or, for a delayed chained payment, secondary receivers have not been paid • ERROR – The payment failed and all attempted transfers failed or all completed transfers were successfully reversed • REVERSALERROR – One or more transfers failed when attempting to reverse a payment • PROCESSING – The payment is in progress • PENDING – The payment is awaiting processing
sender_email	Sender's email address.
action_type	Whether the Pay API is used with or without the SetPaymentOptions and ExecutePayment API operations. Possible values are: • PAY – If you are not using the SetPaymentOptions and ExecutePayment API operations • CREATE – If you are using the SetPaymentOptions and ExecutePayment API operations
payment_request_date	The date on which the payment request was initiated.
reverse_all_parallel_payments_on_error	Whether the payment request specified to reverse parallel payments if an error occurs. Possible values are: • true – Each parallel payment is reversed if an error occurs • false – Only incomplete payments are reversed (default)
transaction[n].id	The transaction ID, where [n] is a number from 0 to 5. For simple, single receiver payments, this number will be 0. Numbers larger than 0 indicate the payment to a particular receiver in chained and parallel payments.
transaction[n].status	The transaction status, where [n] is a number from 0 to 5. For simple single-receiver payments, this number will be 0. Numbers larger than 0 indicate the payment to a particular receiver in chained and parallel payments. Possible values are: • SUCCESS – The sender's transaction has completed • PENDING – The transaction is awaiting further processing • CREATED – The payment request was received; funds will be transferred once approval is received • PARTIALLY_REFUNDED– Transaction was partially refunded • DENIED – The transaction was rejected by the receiver • PROCESSING – The transaction is in progress • REVERSED – The payment was returned to the sender
transaction[n].id_for_sender	The transaction ID for the sender, where [n] is a number from 0 to 5. For simple, single receiver payments, this number will be 0. Numbers larger than 0 indicate the payment to a particular receiver in chained and parallel payments.

Variable	Description
`transaction[n].status_for _sender_txn`	The transaction status, where [n] is a number from 0 to 5. For simple single-receiver payments, this number will be 0. Numbers larger than 0 indicate the payment to a particular receiver in chained and parallel payments. Possible values are: • `SUCCESS` – The sender's transaction has completed • `PENDING` – The transaction is awaiting further processing • `CREATED` – The payment request was received; funds will be transferred once approval is received • `PARTIALLY_REFUNDED` – Transaction was partially refunded • `DENIED` – The transaction was rejected by the receiver • `PROCESSING` – The transaction is in progress • `REVERSED` – The payment was returned to the sender
`transaction[n].refund_id`	The identification number for the refund
`transaction[n].refund_am ount`	The amount that was refunded.
`transaction[n].refund_ac count_charged`	The email address of the debit account of the refund.
`transaction[n].receiver`	The receiver's email address for the transaction
`transaction[n].invoiceId`	The invoice number for this transaction
`transaction[n].amount`	The payment amount of the transaction
`transaction[n].is_primar y_receiver`	Whether there is a primary receiver for this transaction, which indicates whether the transaction is a chained payment. Possible values are: • `true` – There is a primary receiver (chained payment) • `false` – There is no primary receiver (simple or parallel payment)
`return_url`	The URL to which the sender's browser is redirected after approving a payment on paypal.com. Use the pay key to identify the payment as follows: `payKey=${payKey}`.
`cancel_url`	The URL to which the sender's browser is redirected if the sender cancels the approval for a payment on paypal.com. Use the pay key to identify the payment as follows: `payKey=${payKey}`.
`ipn_notification_url`	The URL to which all IPN messages for this payment are sent.
`pay_key`	The pay key that identifies this payment. This is a token that is assigned by the Pay API after a `PayRequest` message is received and can be used in other Adaptive Payments APIs as well as the `cancelURL` and `returnURL` to identify this payment. The pay key is valid for 3 hours.
`memo`	A note associated with the payment.

Variable	Description
fees_payer	The payer of PayPal fees. Possible values are: • SENDER – Sender pays all fees (for personal, implicit simple/parallel payments; do not use for chained or unilateral payments) • PRIMARYRECEIVER – Primary receiver pays all fees (chained payments only) • EACHRECEIVER – Each receiver pays their own fee (default, personal and unilateral payments) • SECONDARYONLY – Secondary receivers pay all fees (use only for chained payments with one secondary receiver)
trackingId	The tracking ID that was specified for this payment in the PaymentDetailsRequest message.
preapproval_key	The preapproval key returned after a PreapprovalRequest, or the preapproval key that identifies the preapproval key sent with a PayRequest.
reason_code	Whether this transaction is a chargeback, partial, or reversal. Possible values are: • Chargeback Settlement – Transaction is a chargeback • Admin reversal – Transaction was reversed by PayPal administrators • Refund – Transaction was partially or fully refunded

Preapproval Message Variables

PayPal generates an IPN message that contains information about a preapproval in response to the Adaptive Payments Preapproval API operation.

Variable	Description
transaction_type	The type of transaction. For a preapproval, this variable returns Adaptive Payment Preapproval. **NOTE:** If this variable is set to Adaptive Payment Pay or Adjustment, refer to the Pay Message Variable section.
preapproval_key	The preapproval key returned after a PreapprovalRequest.
approved	Whether the preapproval request was approved. Possible values are: • true – The preapproval was approved • false – The preapproval was denied
cancel_url	The URL to which the sender's browser is redirected if the sender decides to cancel the preapproval as requested. Use the preapproval key to identify the payment as follows: preapprovalKey=${preapprovalKey}
current_number_of_payments	The current number of payments made for this preapproval.

Variable	Description
`current_total_amount_of_all_payments`	The current total of payments made for this preapproval.
`current_period_attempts`	The current number of attempts this period for this preapproval.
`currencyCode`	The currency code. Possible values are: ● Australian Dollar – `AUD` ● Brazilian Real – `BRL` **NOTE:** The Real is supported as a payment currency and currency balance only for Brazilian PayPal accounts. ● Canadian Dollar – `CAD` ● Czech Koruna – `CZK` ● Danish Krone – `DKK` ● Euro – `EUR` ● Hong Kong Dollar – `HKD` ● Hungarian Forint – `HUF` ● Israeli New Sheqel – `ILS` ● Japanese Yen – `JPY` ● Malaysian Ringgit – `MYR` **NOTE:** The Ringgit is supported as a payment currency and currency balance only for Malaysian PayPal accounts. ● Mexican Peso – `MXN` ● Norwegian Krone – `NOK` ● New Zealand Dollar – `NZD` ● Philippine Peso – `PHP` ● Polish Zloty – `PLN` ● Pound Sterling – `GBP` ● Singapore Dollar – `SGD` ● Swedish Krona – `SEK` ● Swiss Franc – `CHF` ● Taiwan New Dollar – `TWD` ● Thai Baht – `THB` ● U.S. Dollar – `USD`
`date_of_month`	The day of the month on which a monthly payment is to be made. A number between 1 and 31 indicates the day of the month. A value of 0 indicates that the payment can be made on any day.

Variable	Description
day_of_week	The day of the week that a weekly payment is to be made. Possible values are: • NO_DAY_SPECIFIED • SUNDAY • MONDAY • TUESDAY • WEDNESDAY • THURSDAY • FRIDAY • SATURDAY
starting_date	First date for which the preapproval is valid.
ending_date	Last date for which the preapproval is valid. Time is currently not supported.
max_total_amount_of_all_payments	The preapproved maximum total amount of all payments.
max_amount_per_payment	The preapproved maximum amount of all payments.
max_number_of_payments	The maximum number of payments that is preapproved.
payment_period	The payment period. Possible values are: • NO_PERIOD_SPECIFIED • DAILY • WEEKLY • BIWEEKLY • SEMIMONTHLY • MONTHLY • ANNUALLY
pin_type	Whether a personal identification number (PIN) is required. It is one of the following values: • NOT_REQUIRED – A PIN is not required • REQUIRED – A PIN is required
sender_email	The sender's email address.

Adaptive Accounts IPN Messages

PayPal sends an IPN message whenever a PayPal account is created using the
CreateAccount API operation.

Field	Description
notify_version	Message's version number
first_name	Account holder's first name
last_name	Account holder's last name
verify_sign	Encrypted string used to validate the authenticity of the transaction
charset	Character set
account_key	Account key returned by the CreateAccount API operation
confirmation_code	Confirmation code

Example IPN message in response to CreateAccount request

```
notify_version=UNVERSIONED&first_name=Bonzo&verify_sign=AXGvd8phm-
RhbVH.QG69I2sFCPP9AWhNtLiyAHVfld6At5D25Wf4lvCu&charset=gb2312&last_name=Zai
us&account_key=AA-3PW39611LE744914C&confirmation_code=08052490912533831876
```

Payment Review Using Notifications

If PayPal places a payment under review, PayPal sends messages containing the payment_status and pending_reason variables. In addition, it sends a protection_eligibility variable that identifies the kind of protection in force for the transaction.

The initial IPN or PDT message indicates that the payment is under review:

- The payment_status variable is set to Pending.
- The pending_reason variable is set to PaymentReview.

A subsequent IPN message indicates the resolution of the payment review:

- If the transaction was successful and the payment was accepted, PayPal sends a notification whose payment_status variable is set to Completed.
- If the transaction failed and the payment was rejected, PayPal sends a notification whose payment_status variable is set to Reversed.

NOTE: PDTs are only available for the initial status.