PayPal®

Sandbox User Guide

For Professional Use Only
Currently only available in English.

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

Last Updated: April 2007

PayPal Sandbox User Guide

Document Number: 100007.en_US-200704

Notice of non-liability:
PayPal, Inc. is providing the information in this document to you "AS-IS" with all faults. PayPal, Inc. makes no warranties of any kind (whether express, implied or statutory) with respect to the information contained herein. PayPal, Inc. assumes no liability for damages (whether direct or indirect), caused by errors or omissions, or resulting from the use of this document or the information contained in this document or resulting from the application or use of the product or service described herein. PayPal, Inc. reserves the right to make changes to any information herein without further notice.
PayPal, Inc. does not guarantee that the features described in this document will be announced or made available to anyone in the future.

Preface

This Document

This document describes the PayPal virtual test environment called the Sandbox.

Intended Audience

This document is written for merchants and developers who want to test their PayPal-based applications before using them in production.

Organization of This Document

Chapter 1, "Overview to the PayPal Sandbox," describes the Sandbox in general and lists the differences between the Sandbox and the production PayPal services.

Chapter 2, "Accessing the PayPal Sandbox and Email," describes how to gain access to the Sandbox and your Sandbox email.

Chapter 3, "Setting up Test Users," describes the different kinds of business roles and corresponding test accounts you need to set up on the Sandbox for effective testing of your PayPal-based applications.

Chapter 4, "Testing PayPal Website Features," describes how you can use the Sandbox (https://www.sandbox.paypal.com) to test features that correspond to the features on the main PayPal interactive site https://www.paypal.com/.

Chapter 5, "Testing PayPal NVP APIs," describes how to test the Express Checkout NVP API in the Sandbox.

Chapter 6, "Testing Error Conditions," describes how to use negative testing to simulate API error codes, address verification errors, and credit card validation errors.

Chapter 7, "Technical Support," provides information on contacting Technical Support.

Notational Conventions

This document uses typefaces to identify the characteristics of text. These typefaces and the characteristics they imply are described below:

Typeface	How Used
serif italics	A document title.
	A term being discussed or defined. For example: A file is a readable or writable stream of characters …
	Boolean values (not keywords). For example: The function returns true if it encounters an error.
`monospaced`	Pathnames or file names that appear in body text frames.
	Code-related names that appear in body text frames. Such names are used for functions, callbacks, arguments, data structures, and fields. For example: `AbstractResponseType` is the SOAP response type definition on which all PayPal API response methods are based.
	Components of Internet protocol requests and responses, such as HTTPS and `FORM` variables. For example: The PayPal system uses a `method=POST` request to return IPN status variables related to subscriptions, such as `txn_type`.
Serif bold	User interface names, such as window names or menu selections. For example: On the **Profile** page, click Email to confirm your email address.
San-serif oblique	Placeholders used in the context of a format or programming standard or formal descriptions of PayPal system syntax. Placeholders indicate values or names that the reader should provide. Example: For example, amount is the variable for a single-item shopping cart, but amount_X is the name of the variable for a multi-item shopping cart. *amount_3* is the item amount for the third item in a multiple-item shopping cart.

To convey additional information, this document may also apply color and underlining to words or phrases that use the typefaces described above. Such use is described below:

Text attribute	How Used
xxxxxx	Hypertext link to a page in the current document or to another document in the set.
xxxxxx	Hypertext link to a URL or that initiates a web action, such as sending mail.

Revision History

Revision history for *PayPal Sandbox User Guide*.

TABLE P.1 Revision History

Date	Description
April 2007	Added chapters on negative testing and testing APIs using the Express Checkout NVP API. Added new section on handling pending transactions and made miscellaneous changes.
July 2006	Correction of variable name ‘ipn_test,” which should be “test_ipn”.
June 2006	Correction of Sort Code necessary to test UK accounts in Sandbox. Proper Sort Code is 609204.
December 2005	Miscellaneous corrections.

1 Overview to the PayPal Sandbox

The PayPal Sandbox is a self-contained environment within which you can prototype and test your PayPal applications. The PayPal Sandbox is an almost identical copy of the live PayPal website. Its purpose is to give developers a shielded environment for testing and integration purposes and to help avoid problems that might occur while testing PayPal integration solutions on the live site. Before moving any PayPal-based application into production, you should test the application in the Sandbox to ensure that it functions as you intend and within the guidelines and standards set forth by the PayPal Developer Network (PDN).

Get Started Quickly: Integration Center

PayPal's Integration Center has step-by-step details for getting started with the PayPal Software Development Kits (SDKs), Website Payments Pro, Express Checkout, Website Payments Standard, Authorization & Capture, Instant Payment Notification, and more.

Visit the Integration Center at:https://www.paypal.com/integration.

At a Glance: Differences between the Sandbox and Live PayPal

The following table compares the Sandbox and Live PayPal. This is an at-a-glance view of the differences from the perspective of an in-house or third-party developer for a business.

You can also use this table as a checklist.

TABLE 1.1 Differences between Developer Central, Sandbox, and Live PayPal

	PayPal Developer Central and Sandbox	Live PayPal Website and API Service
Type of PayPal Accounts	Depending on the feature you want to develop and test, you need a Personal, Business, or Premier account.	Personal, Business, or Premier account
Developer Central: URL and site logo in upper left corner	https://developer.paypal.com PayPal Developer Central	

TABLE 1.1 Differences between Developer Central, Sandbox, and Live PayPal

	PayPal Developer Central and Sandbox	Live PayPal Website and API Service
Site logos in upper left corner	https://www.sandbox.paypal.com PayPal Sandbox	https://www.paypal.com PayPal

NVP API Servers	`https://api.sandbox.paypal.com/nvp/`	For API Certificate security:
		`https://api.paypal.com/nvp/`
		For API Signature security:
		`https://api-35.paypal.com/nvp/`

SOAP API Servers	`https://api.sandbox.paypal.com/2.0/`	For API Certificate security:
		`https://api.paypal.com/2.0/`
		For API Signature security:
		`https://api-3t.paypal.com/2.0/`

Business roles	You fill all roles you need to test: merchant, buyer, and seller.	Real-world people fill these roles.
Company and people's names and postal addresses	Completely fictitious. Before you begin working with the Sandbox, create the details for all the business roles you must fulfill. The Sandbox simulates verification of postal addresses and names.	Real companies' and people's names and postal addresses.
Email addresses and email inboxes	The Sandbox has a special-purpose email inbox for your testing, contained in the Sandbox itself.	Real email address and inbox for each business role
Bank account and credit card numbers	The Sandbox creates bank accounts, credit card numbers, and CVV2 numbers you need in order to develop and test; all of which are fictitious and only used within the Sandbox. The Sandbox simulates the verification of these numbers. Transactions do not affect real accounts and actual money is never exchanged.	Actual verification of bank account numbers, credit card numbers, and CVV2 numbers
Social Security Number for Billing Agreements	111-*nn-nnnn*	Real social security numbers
PayPal transactions	The Sandbox creates all fictitious bank accounts, credit card numbers, and CVV2 numbers you need for development and testing. The Sandbox simulates the verification of these numbers.	Live transactions, cleared by live PayPal processes

TABLE 1.1 Differences between Developer Central, Sandbox, and Live PayPal

	PayPal Developer Central and Sandbox	Live PayPal Website and API Service
Fraud detection	Fraud detection is not enabled for the Sandbox.	Full protection through PayPal's fraud detection
Digital certificates	After you request digital certificates for use with the PayPal Web Services API, the Sandbox automatically generates them. They are available for immediate downloading.	To safeguard your and your customers' security, requests for digital certificates for use with the Live PayPal Web Services API must be verified by PayPal before they are issued. You are notified in email when your request has been approved.
PayPal Merchant Features supported	All features of the live PayPal website, except closing an account, auction features, monthly statements, shipping preferences, and PayPal Shops.	
Technical Support	Developer Technical Support is available from Developer Central. No telephone support.	Customer Service is available either by email or by telephone. See "Contacting Customer Service for Live PayPal Website Help" on page 49.

2 Accessing the PayPal Sandbox and Email

To access the PayPal Sandbox, sign up for a Developer Central account. After becoming a member of PayPal Developer Central, you access the Sandbox either directly or by launching it after you log in to Developer Central.

Depending on the PayPal feature you want to test with an application, you need to set up different types of PayPal accounts: PayPal Personal, Business, or Premier account. See "Planning the Types of Test Accounts You Need" on page 17.

Signing Up for Developer Central

You do not need an existing, live PayPal account in order to sign up for Developer Central.

1. Go to https://developer.paypal.com.
2. Click **Sign Up Now**.
3. Follow the on-screen instructions.
4. After you sign up, PayPal sends login instructions to the email address you used to sign-up. If you have mail filtering enabled in your mail software, the email sent by PayPal might be filtered out or stored in a folder where you are not expecting it to be. For instance, with Microsoft Outlook mail software, your filtering might cause the email to be stored in "Junk" or "Spam."

Getting to the Sandbox

You have two ways to get to the PayPal Sandbox: direct access or logging in to Developer Central and then launching the Sandbox.

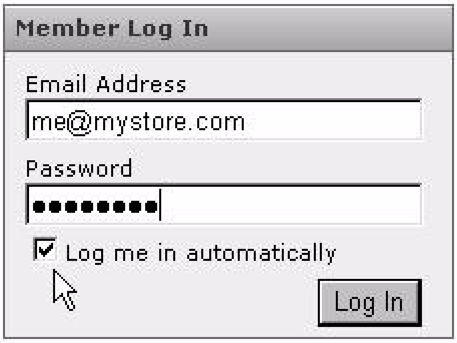
Regardless of whether you get to the Sandbox directly or by launching it, log in to the Sandbox as a test user of the necessary type to test the feature you want to test: Personal, Business, or Premier account.

Accessing the Sandbox Directly

When you log in to Developer Central, you have the option to get to the Sandbox directly at https://www.sandbox.paypal.com/. At the Developer Central login page, click **Log me in automatically, as shown in** Figure 2.1, "Login for PayPal Developer Central" on page 14.

NOTE: To access the Sandbox directly, you must enable cookies in your browser.

FIGURE 2.1 Login for PayPal Developer Central



Member Log In

Email Address: me@mystore.com

Password: ●●●●●●●●

☑ Log me in automatically

Launching the Sandbox

To launch the Sandbox:

1. Log in to Developer Central.
2. Click the **Sandbox** tab.
3. Select the test user that represents the feature you want to test.
4. Click **Launch Sandbox**.

Managing Your Developer Central Profile

Except for your registered email address, you can change all the information associated with your Developer Central account, including your password. (Your email address is your login username.)

To manage your Developer Central profile:

1. Log in to Developer Central.
2. Click **profile** in the upper left of the main Developer Central page.
3. Change any of details, except your email address.
4. Click **Save**

Sandbox Email

When certain kinds of transactions occur in the live PayPal system, PayPal sends email messages to participants in the transaction. From these email messages, the recipient or initiator of an event or transaction can verify that the event took place and that the monetary amounts associated with the event are correct. With the live PayPal system, email messages are sent to the real email addresses of the participants.

PayPal Sandbox email, however, is a self-contained email system in the Sandbox itself. Each developer registered with PayPal Developer Central has a single inbox to which all email messages for test users are sent. You see email messages addressed only to test users associated with your Developer Central account: the Sandbox test accounts you set up.

To access your email inbox in the Sandbox:

1. Log in to Developer Central.
2. Click the **Email** tab.

   Up to 30 of the latest email messages are listed in your inbox. The subject line of email messages you have not read are in bold. Click a subject line to read the message.

3 Setting up Test Users

Depending on the business application you are developing and testing, you need different types of test accounts. There are two types of test accounts: Personal and Business.

Planning the Types of Test Accounts You Need

Determine the types of test accounts you need to test the applications you are developing. In addition, determine the number of different accounts you need. You might need several different Personal or Business PayPal test accounts to test your application.

The information you need to supply when you create test accounts is as follows:

- Mailing address
- Email address and password for the test PayPal account. You can use the same password (not email address) for all your test accounts so that you can more easily remember it.
- Security questions and answers. You can use the same security questions and answers for all your test accounts so that you can more easily remember them.

IMPORTANT: The email address and password of a test account should never be those a real account on the live PayPal site. The same applies to your answers to the security questions. All of this data should be fictional.

- Personal or Business account
- Your agreement to the terms of using the Sandbox

For Business accounts, you also need to supply the following fictitious information for testing:

- Business name and address
- Customer service contact information
- Business owner contact information
- Business owner address
- Social Security Number to sign up for Website Payments Pro

Managing Test Accounts

In Developer Central, you can view, work with, or launch the Sandbox for all your test accounts. You can also create new accounts or remove test email addresses from your view.

- To work with test accounts, log in to Developer Central, and click the **Sandbox** tab.

- To create a new account, click the Create Account link in the upper left corner of the list of accounts.
- To work with the account, select the account by clicking the radio button associated with it on the left.

You can start the Sandbox for the selected account by clicking **Launch Sandbox**. When you logged in to Developer Central, you might have set the **Log me in automatically** checkbox to allow direct access to https://www.sandbox.paypal.com/, in which case you do not have to launch the Sandbox to access it.

NOTE: The **Delete** button does not delete the test account. It removes the test account from your list of accounts, but the email address for the test account is still on file for the Sandbox. You cannot reuse an email address that is still on file for the Sandbox.

Creating a Personal Account

To create a test user with a Personal account:

1. Log in to Developer Central.
2. Click the **Sandbox** tab.
3. Click the **Create Account** link.
4. On the next page, select the **Personal Account** radio button. Then click **Continue**.
5. Next, enter the account information for the new test user's PayPal Sandbox account. The email address entered on this page serves as the PayPal ID for that test user. Use a fictitious email address for each new test user, so that the PayPal ID is unique. Remember that passwords are case-sensitive.
6. Click the **Sign Up** button.
7. On the next page (confirming email address), click the **Continue** button. This goes to the **My Account** > **Overview** tab.

   To confirm the email address for the new test user, see the steps in "Confirming an Email Address" on page 19.

Creating a Business Account

To create a test user with a Business account:

1. Log in to Developer Central.
2. Click the **Sandbox** tab.
3. Click the **Create Account** link.
4. On the next page, select the **Business Account** radio button. Then click the **Continue** button.

5. Next, enter account information for the business represented by the new PayPal account.
6. Click **Continue** when done.
7. Enter the user information for the new test user's PayPal Sandbox account.

   Click the **Sign Up** button when done.
8. Click **Continue**. This takes you to the **My Account** > **Overview** tab.

Confirming an Email Address

To confirm an email address for a test user:

1. Log in to Developer Central.
2. Click the **Email** tab.
3. On the next page, click the email link to open the email window.
4. In the window that appears, find the email for the new test user with a subject line of "Activate Your PayPal Account!"
5. Open the email and click the confirmation link.

"Unverified" Account Status

At this point your new test users exist and have confirmed email addresses. However, they have an "Unverified" status. To set a user to a status of "Verified", add a bank account for it. For more information, see "Adding a Bank Account" on page 20.

Adding a Funding Source

To test transactions, you must add a source of funds to your buyer test account. The following sections describe your choices:

1. "Adding a Bank Account" on page 20. You can add bank accounts, but they will not contain funds unless you use Send Money to send the bank account holder money.
2. "Adding a Credit Card " on page 21. For testing, this is the most efficient way to add funds.

NOTE: No money or funds are actually transferred in the Sandbox; however, to protect confidentiality, you should not use actual credit card numbers or bank accounts if you allow other people to log in to your Developer Central account.

Adding a Bank Account

You add a bank account to the Sandbox test account representing a customer or buyer so that you can test transactions between the buyer's account and another account; typically, the other account is a business account that represents yourself as a merchant. Adding a bank account also changes the account status from "Unverified" to "Verified."

The bank account is a source of funds for the test user's PayPal account, and thus for transactions between that test user and other test users. A test user can have multiple bank accounts, but at least one is required in order to verify the test user.

The Sandbox automatically generates bank account and sort code numbers when you add a bank account.

For Australia, Canada, Germany, or UK

Use the automatically generated bank account information only for test US bank accounts. To add test Canadian, German, or UK bank account information, follow these guidelines.

TABLE 3.1 Australian German, Canadian, and UK Test Bank Account Information

Australia	Canada	Germany	UK
BSB Number: 242-200	**Transit Number:** 00001	**Routing Number:** 37020500	**Bank Account Number:** Any 8-digit number
Account Number: any random number	**Institution Number:** 311	**Bank Account Number:** Any 10-digit number	**Sort Code:** 609204 or 700709
	Bank Account Number: Any one-digit to 12-digit number	**Sort Code:** Any 8-digit number	

Steps for All Countries

To add a bank account for a new test user:

1. After logging into Developer Central, launch the Sandbox as the test user.

2. Navigate to **My Account** > **Overview**.

3. In the Overview window, click the **Add Bank Account** link in the Activate Account box on the left.

4. In the Add Bank Account window:

   – Enter a fictitious bank name. If you copy-and-paste the automatically generated bank account number as the name of the bank, that account number will be more visible to you for use in later testing.
   – Except for UK or German test bank accounts, leave all other automatically generated information as is.

- Make a note of the test bank account number, because it will be handy to have when you do your testing.
- Click **Add Bank Account**.

5. In the resulting success page, click **Continue** at the bottom.

   The **My Account** > **Overview** page opens.

6. Click the **Confirm Bank Account** link in the **Activate Account** box at the left side.

7. In the Confirm Bank Account window, click **Submit**.

   To create additional bank accounts for an existing test user:

1. After logging into Developer Central, launch the Sandbox as the test user.

2. Navigate to **My Account** > **Profile**.

3. Under the **Financial Information** header, click the **Bank Accounts** link.

4. In the **Bank Account** window, click **Add**.

5. In the **Add Bank Account** window:

   - Enter a fictitious bank name. Using the automatically generated bank account number as the name of the bank will make that account number visible to you for use in testing later.
   - Except for UK or German test bank accounts, leave all other automatically generated information as is.
   - Make a note of the test bank account number, because it will be handy to have when you do your testing.
   - Click **Add Bank Account**.

6. In the resulting success window, click the **Continue** button at the bottom.

   The **My Account** > **Overview** page opens.

7. Click the **Confirm Bank Account** link in the **Activate Account** box at the left side.

8. In the **Confirm Bank Account** window, click **Submit**.

Adding a Credit Card

A credit card is a source of funds for the test user's PayPal account, and thus can be used for transactions between that test user and other test users. A test user can have multiple credit cards.

Test credit card numbers cannot be used to pay for real-world transactions.

NOTE: Skip this step if you signed up for a Business Account. Business Accounts automatically get a credit card.

Adding a Credit Card for a New Test User

To add a credit card account for a new test user:

1. After logging into Developer Central, launch the Sandbox as the test user.
2. Navigate to **My Account** > **Overview**.
3. In the **Overview** window, click the **Add Credit Card** link in the **Activate Account** box on the left.
4. In the **Add Credit Card** window, leave all information as it is (automatically generated) and click **Add Credit Card**.

Adding More Credit Cards for an Existing Test User

To create additional credit card accounts for an already existing test user:

1. Log in to the Sandbox as the test user.
2. Navigate to **My Account** > **Profile**.
3. Under the Financial Information header, click the Credit Cards link.
4. In the **Credit Cards** window, click the **Add** button.
5. In the **Add Credit Card** window, leave the automatically generated information as is.
6. Make a note of the credit card number for your use in later testing.
7. Click **Add Credit Card**.

Generating a Credit Card Number to Test PayPal Account Optional

To obtain a test credit card number for testing PayPal Account optional:

1. Log in to the Sandbox as the test user.
2. Navigate to **My Account** > **Profile**.
3. Under the **Financial Information** header, click the **Credit Cards** link.

   Make a note of the credit card number for your use in later testing.

Signing Up for Website Payments Pro

To sign up for Website Payments Pro, create a Business account, as described in "Creating a Business Account" on page 18."

To complete the application for Website Payments Pro, you must enter a Social Security Number. You can enter a Social Security Number in the following format:

`111`*xxxxxx*

where *x* is any digit.

NOTE: The SSN you enter must not have already been recorded for some other user of the Sandbox.

4 Testing PayPal Website Features

This chapter describes PayPal products features you can test in the Sandbox without PayPal APIs:

- **Website Payments with Buy Now Buttons**: Use the Sandbox to test accepting PayPal as a payment mechanism on a website.
- **Shopping Cart Purchases**: Use the Sandbox to test the purchase of multiple items in a single transaction using a single payment.
- **Instant Payment Notification (IPN)**. Use the Sandbox to test IPN for updates and payment notifications.
- **Refunds:** Use the Sandbox to test refunding payments from a test buyer.
- **Subscriptions**: Use the Sandbox to test subscription buttons.

IMPORTANT: To execute test transactions on Sandbox you need to complete a purchase as a test buyer with your buyer test account. Typically, you go through your website purchase flow as a buyer. You must ensure that you execute your test on `www.sandbox.paypal.com` instead of `www.paypal.com`.

Website Payments with Buy Now Button

You can use the Sandbox to familiarize yourself with the PayPal **Buy Now** button, with which you can associate PayPal with a specific item you sell on your website.

To create a test **Buy Now** button:

1. After logging into Developer Central, launch the Sandbox as the test seller account.
2. Go to the **Merchant Services** tab.
3. Select the **Buy Now Buttons** link under the **Key Features** heading to get to the Button Factory. You can also search the Help for “Button Factory.”
4. Follow the online instructions to create a Buy Now button. For more information, see the *Website Payments Standard Integration Guide*.
5. Copy and paste the code into your web page file wherever you would like the button image to appear. Typically, the button should be located next to the description of the item or service. Your web page does not have to be published to your web server for you to check the button placement; it can be on you own local hard drive.

IMPORTANT: You must change the form action to redirect to the Sandbox, using the following URL:

```
https://www.sandbox.paypal.com/cgi-bin/webscr"
method="post"
```

Use the PayPal Help link to answer related questions, such as “How do I make a Buy Now Button compatible with the Shopping Cart feature?” For general information, see https://www.paypal.com/pdn-item. For general information about shopping cart purchases, see https://www.paypal.com/shoppingcart. For general information about subscriptions, see https://www.paypal.com/pdn-recurring.

Encrypted Website Payments

The Sandbox also supports Encrypted Website Payments (EWP), as does the PayPal SDK console.

For information about what EWP is and how to use it, see the *Website Payments Standard Integration Guide*

For information about using the PayPal SDK console to generate EWP HTML, see the *PayPal SDK Guide* for any of the supported platforms.

Testing Payments with Buy Now Button

For the purposes of testing the Buy Now button, your web page does not need to be published to your web server. It can reside on your local hard drive. However, you do need to be logged in to the Sandbox.

1. Log in to Developer Central, click the **Sandbox** tab, select the desired test user, and click **Launch Sandbox**.
2. Open the HTML file containing the Buy Now Button.
3. Click the **Buy Now** Button.
4. Log in using your test buyer account.
5. Follow the on-screen instructions to complete your test payment.

Verifying a Test Payment

1. Log in to Developer Central.
2. Click the **Email** tab.

   Your Sandbox inbox shows payment confirmation email messages for the seller and buyer.
3. To further verify that the payment was successful:
   - Check your web server for IPN notifications related to the payment.
   - Launch the Sandbox as your test buyer or seller account and navigate to **My Account** > **Overview** to see the transaction in your **Recent Activity**.

Handling Pending Transactions

Transactions typically are credited to your PayPal account instantly after the buyer completes the transaction; however, the user might select a payment method that is not completed instantly. In these cases, the transaction goes into a pending state and the transaction is completed after a couple of days. The following sections describe how to set up pending status transactions that can either be completed or canceled.

Creating a Pending Transaction

1. Log in to Developer Central, click the **Sandbox** tab, select the desired test user, and click **Launch Sandbox**.
2. Open the HTML file containing the Pay Button.
3. Click the Pay Button.
4. Log in using your test buyer account.
5. On the **Review Purchase Page** click on the link **Change** under funding method.
6. Select **Electronic Funds Transfer** as funding method and click **Continue**.
7. Click **Pay** to create the transaction.

To verify the creation of the transaction, see "Verifying a Test Payment" on page 26.

Completing or Canceling a Pending Transaction

1. In the transactions log, click the **Details** link (in the Details column).
2. In the Transaction Detail window, there are two links to simulate actual bank clearing. These links appear only in the Sandbox:
   - **Clear Transaction:** Click to complete the transaction.
   - **Fail Transaction:** Click to cancel the transaction.

Instant Payment Notification (IPN)

You can use the Sandbox to test Instant Payment Notification, such as the PayPal **Buy Now** button or reversals.

Setting up IPN in the Sandbox

For information about implementing IPN, see the following:

- Technical overview at https://www.paypal.com/cgi-bin/webscr?cmd=p/xcl/rec/ipn-techview-outside
- *Order Management Integration Guide*

test_ipn Variable

The Sandbox sets the variable `test_ipn` with a value of 1 in the HTTP response back to your IPN page. The purpose of this variable is to clearly differentiate between live and Sandbox IPN, so you can write your processing programs to work with either live or Sandbox IPN. If the test_ipn variable is not present in the HTTP response, you are working with Live PayPal.

To enable IPN for a test user:

1. Log in to Developer Central, click the **Sandbox** tab, select the desired test user, and click **Launch Sandbox**.
2. Click the **Profile** subtab.
3. Click the **Instant Payment Notification Preferences** link in the **Selling Preferences** column.
4. Click **Edit**.
5. Click the checkbox and enter the URL at which you would like to receive your IPN notifications.
6. Click **Save**.
7. Install IPN on your web server. You might want to start with one of PayPal's source code samples available at https://www.paypal.com/ipn under the **Code Samples** section. There are source code samples for the several programming languages and development environments. For further information, see the *Order Management Integration Guide*.

Verifying a Test Refund

To verify a test refund, you must have already made a test payment.

1. Log in to Developer Central.
2. Click the **Email** tab.

   Your Sandbox inbox shows refund confirmation email messages for the seller and buyer.
3. To further verify that the refund was successful:
   - Check your Web server for IPN notifications related to the refund.
   - Launch the Sandbox as your test buyer or seller account and navigate to **My Account** > **Overview** to see the transaction in your **Recent Activity**.

Transferring Funds to a Test Account

Before a test user can exchange funds with other test users in transactions, you must manually transfer funds to the test user's account.

To transfer funds to a test user's PayPal account:

1. After logging into Developer Central, launch the Sandbox as the test user.
2. Navigate to **My Account** > **Add Funds**.
3. Click the Transfer funds from a Bank Account link.
4. On the Add Funds by Electronic Funds Transfer page:
   - Select the bank account from which the funds are coming in the From drop-down list.
   - Enter the amount to transfer in the Amount box.
   - Click Continue.
5. On the resulting Add Funds Confirmation page, click **Submit**.

   Navigate to **My Account** > **Overview** to see that the transfer transaction is listed.

Clearing or Failing Test eCheck Transactions

When you use eCheck to transfer funds or send payments, the transaction appears as pending until you manually clear or fail it. Manual clearing is only necessary in the Sandbox.

To clear or fail test eCheck transactions:

1. In the transactions log, click the **Details** link (in the Details column).
2. In the Transaction Detail window, there are two links to simulate actual bank clearing. These links appear only in the Sandbox:
   - **Clear Transaction:** Click to complete the transaction.
   - **Fail Transaction:** Click to cancel the transaction.
3. Click **Return to Log** to see the transfer completed and the money in the Sandbox account.

   The **My Account** > **Overview** page opens.
4. Click the **View Limits** links on the **My Account** > **Overview** page to see the spending limits for the current test user.

Sending Funds to a Seller

To purchase goods or services, a PayPal user must send funds to a seller. In the PayPal Sandbox, you can simulate the actions of a buyer by manually initiating the payment of funds. You must use a Personal account that represents a paying customer.

To send funds from one test user to another:

1. Log in to Developer Central, click the **Sandbox** tab, select the desired test user, and click **Launch Sandbox**.
2. Navigate to the **Send Money** tab.
3. On the Send Money page, enter the email address (PayPal account name) for the test user in Recipient's Email box.
4. Enter the amount to send to the seller test user in the Amount box.
5. Select the currency for the funds in the **Currency** drop-down list. (Note: **Auction** is not an option in the drop-down list.)
6. Select the reason for sending the funds in the **Type** drop-down list.
7. Enter text in the Subject box, if you want to. This text is the subject of the email sent to the recipient about the transfer of funds.
8. Enter text in the **Note** memo box. This text appears in the body of the notification email.
9. Click **Continue**. This does not send the money; a confirmation step follows.
10. On the **Check Payment Details** page, review the transaction details for correctness. You can click **More Funding Options** to change the source of fund used for payment.
11. Click **Send Money**. This triggers the actual transfer of funds.
12. Your Sandbox email inbox contains all the email messages that are sent to the test user sending the money and the test user receiving the money. See "Sandbox Email" on page 14.

    Log in as the seller test user and navigate to the My Account > Overview tab to see the transaction for the recipient's account.

Billing A Customer

PayPal Business users can bill another PayPal user for the purchase of goods or services. In PayPal terminology, the feature to bill a customer is called Request Money. In the PayPal Sandbox, you can manually initiate a request for funds from another test user. One test account is the biller. The other test account is the customer.

To request funds from another test user:

1. After logging into Developer Central, launch the Sandbox as the test user requesting the funds.
2. Navigate to the **Request Money** tab.
3. On the **Request Money** page, enter the email address (PayPal login name) for the test user being billed in the **Recipient's Email** box.
4. Enter the billed amount in the **Amount** box.
5. Select the currency for the funds in the **Currency** drop-down list.
6. Select the reason for the request for funds (billing) in the **Type** drop-down list. (Note: **Auction** is not an option in the drop-down list.)
7. Enter text in the **Subject** box. This text is the subject of the email sent to the recipient regarding the sent funds.
8. Enter text in the **Note** memo box. This text appears in the body of the notification email.
9. Click **Continue**.
10. On the **Request Money – Confirm** page, click **Request Money**. This triggers the actual request for funds.
11. Navigate to the **My Account** > **Overview** tab. The request for money should be listed.
12. Log in as the billed test user and navigate to the **My Account** > Overview tab to see the transaction for the billed user's account. The transaction for the request for money appears on the My Account > Overview tab with Pay and Cancel buttons. Click Pay, and in the confirmation window, click Send Money. This completes the transfer of requested funds.

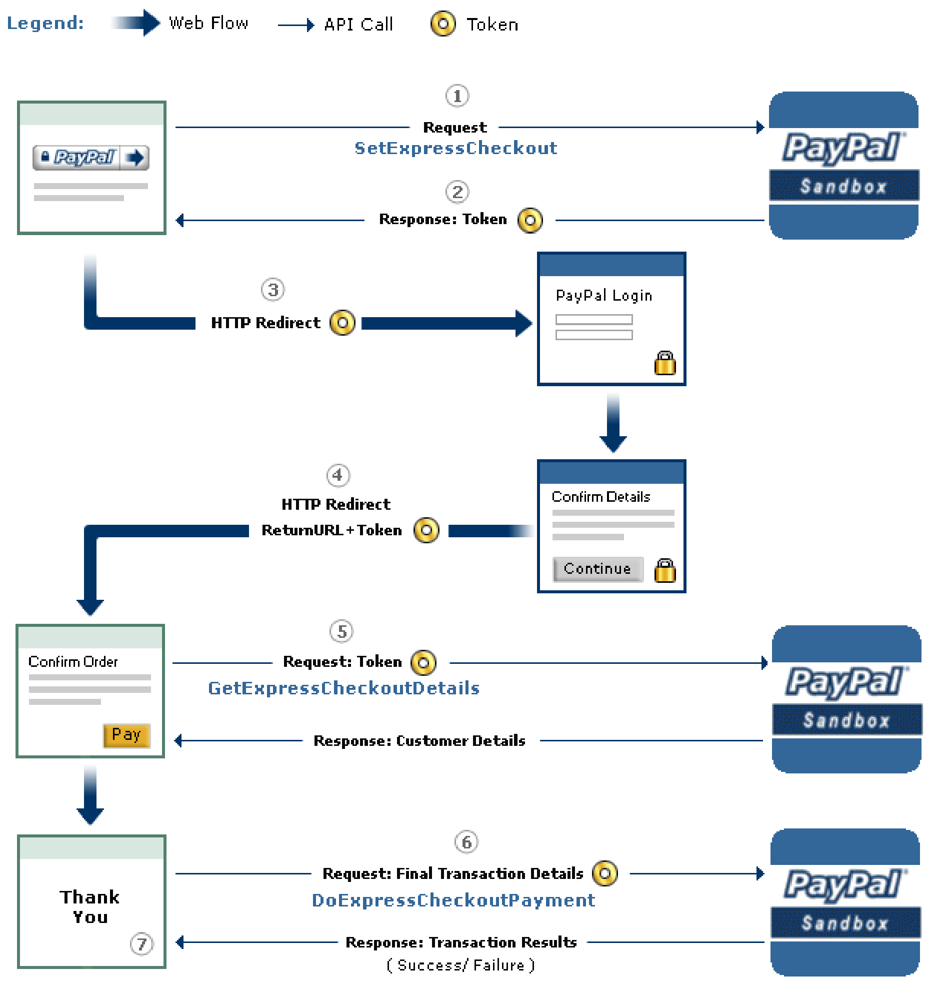
    To view the email messages sent to both test users, go to your Sandbox email. For details about your Sandbox email, see "Sandbox Email" on page 14.

5 Testing PayPal NVP APIs

This chapter describes how to test the Express Checkout name-value pair (NVP) API in the Sandbox. For more sophisticated examples, PayPal recommends you use the PayPal SDK that matches your environment, such as PHP or ASP. You can also use this chapter for ideas on how to establish a general testing procedure for PayPal APIs called from your site.

Testing Express Checkout

The following diagram shows the Express Checkout flow, which uses the Sandbox as the API server. The pages on the left represent your site.



NOTE: For information about Express Checkout, see the *Express Checkout Integration Guide* and the *PayPal NVP API Developer Guide and Reference*.

The following steps match the circled numbers in the diagram. Perform the actions in each step to test Express Checkout.

1. Invoke a form on your site that calls the `SetExpressCheckout` API on the Sandbox. To invoke the API, set form fields whose names match the NVP names of the fields you want to set, specify their corresponding values, and then post the form to `https://api-3t.sandbox.paypal.com/nvp`, as shown in the following example:

```
<form method=post action=https://api-3t.sandbox.paypal.com/nvp>
   <input type=hidden name=USER value= API_username>
   <input type=hidden name=PWD value= API_password>
   <input type=hidden name=SIGNATURE value= API_signature>
   <input type=hidden name=VERSION value=2.3>
   <input type=hidden name=PAYMENTACTION value=Authorization>
   <input name=AMT value=19.95>
   <input type=hidden name=RETURNURL
      value=http://www.YourReturnURL.com>
   <input type=hidden name=CANCELURL
      value=http://www.YourCancelURL.com>
   <input type=submit name=METHOD value=SetExpressCheckout>
</form>
```

NOTE: The API username is a Sandbox business test account for which a signature exists. See the Test Certificates tab of the Sandbox to obtain a signature.

2. PayPal responds with a message, such as the one shown below. Note the status, which should include `ACK` set to `Success,` and a token that is used in subsequent steps.

```
TIMESTAMP=2007%2d04%2d05T23%3a23%3a07Z
&CORRELATIONID=63cdac0b67b50
&ACK=Success
&VERSION=2%2e300000
&BUILD=1%2e0006
&TOKEN=TOKEN=EC%2d1NK66318YB717835M
```

3. If the operation was successful, use the token and redirect your browser to the Sandbox, as follows:

```
https://www.sandbox.paypal.com/cgi-bin/webscr?cmd=_express-checkout
&TOKEN=EC-1NK66318YB717835M
```

NOTE: You may need to replace hexadecimal codes with ASCII codes; for example, you may need to replace `%2d` in the token with a hyphen (-).

4. Log into the Sandbox and confirm details. You must log in to Developer Central before you log in to a Sandbox test account. You log in to the test account that represents the buyer, not the *API_username* business test account that represents you as the merchant.

   When you confirm, the Sandbox redirects your browser to the return URL you specified when calling `SetExpressCheckout`, as in the following example:

```
http://www.YourReturnURL.com/
                    ?token=EC-1NK66318YB717835M&PayerID=7AKUSARZ7SAT8
```

5. Invoke a form on your site that calls the `GetExpressCheckout` API on the Sandbox:

```
<form method=post action=https://api-3t.sandbox.paypal.com/nvp>
   <input type=hidden name=USER value= API_username>
   <input type=hidden name=PWD value= API_password>
   <input type=hidden name=SIGNATURE value= API_signature>
   <input type=hidden name=VERSION value=2.3>
   <input name=TOKEN value=EC-1NK66318YB717835M>
   <input type=submit name=METHOD value=GetExpressCheckoutDetails>
</form>
```

If the operation was successful, the `GetExpressCheckout` API returns information about the payer, such as the following information:

```
TIMESTAMP=2007%2d04%2d05T23%3a44%3a11Z
&CORRELATIONID=6b174e9bac3b3
&ACK=Success
&VERSION=2%2e300000
&BUILD=1%2e0006
&TOKEN=EC%2d1NK66318YB717835M
&EMAIL=YourSandboxBuyerAccountEmail
&PAYERID=7AKUSARZ7SAT8
&PAYERSTATUS=verified
&FIRSTNAME=...
&LASTNAME=...
&COUNTRYCODE=US
&BUSINESS=...
&SHIPTONAME=...
&SHIPTOSTREET=...
&SHIPTOCITY=...
&SHIPTOSTATE=CA
&SHIPTOCOUNTRYCODE=US
&SHIPTOCOUNTRYNAME=United%20States
&SHIPTOZIP=94666
&ADDRESSID=...
&ADDRESSSTATUS=Confirmed
```

6. Invoke a form on your site that calls the `DoExpressCheckoutPayment` API on the Sandbox:

```
<form method=post action=https://api-3t.sandbox.paypal.com/nvp>
   <input type=hidden name=USER value= API_username>
   <input type=hidden name=PWD value= API_password>
   <input type=hidden name=SIGNATURE value= API_signature>
   <input type=hidden name=VERSION value=2.3>
   <input type=hidden name=PAYMENTACTION value=Authorization>
   <input type=hidden name=PAYERID value=7AKUSARZ7SAT8>
   <input type=hidden name=TOKEN value= EC%2d1NK66318YB717835M>
   <input type=hidden name=AMT value= 19.95>
   <input type=submit name=METHOD value=DoExpressCheckoutPayment>
```

```
        </form>
```

7. If the operation was successful, the response should include `ACK` set to `Success`, as follows:

```
TIMESTAMP=2007%2d04%2d05T23%3a30%3a16Z
&CORRELATIONID=333fb808bb23
&ACK=Success
&VERSION=2%2e300000
&BUILD=1%2e0006
&TOKEN=EC%2d1NK66318YB717835M
&TRANSACTIONID=043144440L487742J
&TRANSACTIONTYPE=expresscheckout
&PAYMENTTYPE=instant
&ORDERTIME=2007%2d04%2d05T23%3a30%3a14Z
&AMT=19%2e95
&CURRENCYCODE=USD
&TAXAMT=0%2e00
&PAYMENTSTATUS=Pending
&PENDINGREASON=authorization
&REASONCODE=None
```

6 Testing Error Conditions

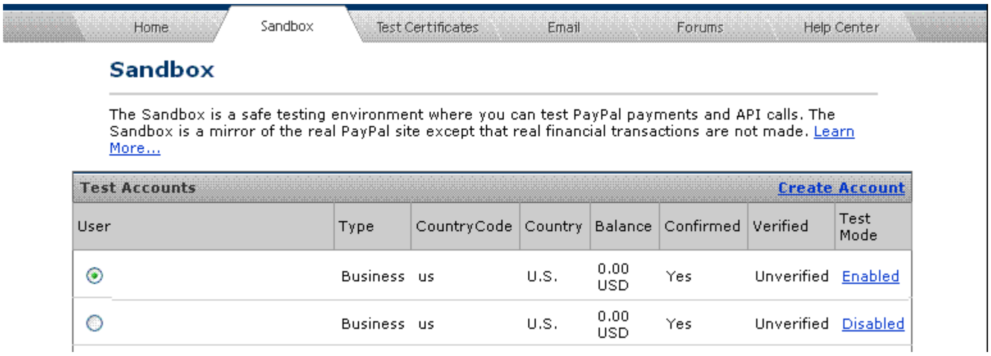
In default operation, the Sandbox mimics the live PayPal site as closely as possible, which means that an error can be replicated only by creating the exact conditions and sequence of events to raise an error. This *positive test* environment is well-suited for testing logic that follows the typical error-free path; however, it can be difficult to raise error conditions and test logic to handle errors.

The Sandbox can be set to allow *negative testing*, which enables you to simulate an error. You can test against the following kinds of errors:

- errors that result from calling a PayPal API
- address verification and credit card validation errors that occur when using Virtual Terminal or calling `DoDirectPayment`.

You raise an error condition by setting a value in a field passed to an API or setting a value in a field submitted to Virtual Terminal. The value triggers a specific error condition. Negative testing is available only in the Sandbox; you cannot force or simulate an error on the live site.

You must create a Business test account and enable negative testing; otherwise, setting a value in the API or transaction will not raise an error unless the error would be raised in the default positive test environment. To enable negative testing, set **Test Mode** to **Enabled**. The following screen shows two Business accounts. The first test account enables negative testing; the second account disables negative testing.



Sandbox

The Sandbox is a safe testing environment where you can test PayPal payments and API calls. The Sandbox is a mirror of the real PayPal site except that real financial transactions are not made. Learn More...

Test Accounts — Create Account

	User	Type	CountryCode	Country	Balance	Confirmed	Verified	Test Mode
◉		Business	us	U.S.	0.00 USD	Yes	Unverified	Enabled
○		Business	us	U.S.	0.00 USD	Yes	Unverified	Disabled

To test Virtual Terminal, you must set risk controls for address verification and credit card security, respectively, to **Decline** or **Accept and Report** depending on the kind of negative testing you want to perform. If you do not set the appropriate risk controls, default processing occurs, which is to accept the transaction.

Severe error conditions, such as bad arguments or invalid login, preempt negative testing because the error cannot be handled by either negative testing or positive testing. In these cases, the error condition for positive testing is raised, regardless of whether the account was enabled for negative testing.

API Testing

For APIs, you trigger an error condition by setting a field to the value of the error you want to trigger. The value you specify depends on the kind of field:

- for amount-related fields, specify a value as a number with two digits to the right of the decimal point; for example, 107.55 triggers PP API error 10755
- for other kinds of fields, specify the actual PP API error; for example, 10755 triggers PP API error 10755

The following table identifies the API, the NVP name or SOAP element of the field that triggers the error, and a description of how to set the value in the field:

TABLE 6.1 API Fields That Trigger Error Conditions

API Name	NVP Field Name	SOAP Element	Description
`RefundTransaction`	`AMT`	`Amount`	Specify the error code to trigger as all digits in a number with two digits to the right of the decimal point; for example, 107.55 triggers PP API error code 10755.
`GetTransaction Details`	`TRANSACTIONID`	`TransactionID`	Specify the error code to trigger as all digits in the field; for example, an ID of 10755 triggers PP API error code 10755.
`TransactionSearch`	`INVNUM`	`InvoiceID`	Specify the error code to trigger as all digits in the field; for example, an ID of 10755 triggers PP API error code 10755.
`DoDirectPayment`	`AMT`	`OrderTotal`	Specify the error code to trigger as all digits in a number with two digits to the right of the decimal point; for example, 107.55 triggers PP API error code 10755.
`SetExpressCheckout`	`MAXAMT`	`MaxAmount`	Specify the error code to trigger as all digits in a number with two digits to the right of the decimal point; for example, 107.55 triggers PP API error code 10755.
`GetExpressCheckout Details`	`TOKEN`	`Token`	Specify the error code to trigger as all digits in the field; for example, a token value of 10755 triggers PP API error code 10755.

***TABLE 6.1 API Fields That Trigger Error Conditions** (Continued)*

API Name	NVP Field Name	SOAP Element	Description
DoExpressCheckout	AMT	Amount	Specify the error code to trigger as all digits in a number with two digits to the right of the decimal point; for example, 107.55 triggers PP API error code 10755.
DoCapture	AMT	Amount	Specify the error code to trigger as all digits in a number with two digits to the right of the decimal point; for example, 106.23 triggers PP API error code 10623.
DoVoid	AUTHORIZATIONID	AuthorizationID	Specify the error code to trigger as all digits in the field; for example, an ID of 10623 triggers PP API error code 10623.
DoReauthorization	AMT	Amount	Specify the error code to trigger as all digits in a number with two digits to the right of the decimal point; for example, 106.23 triggers PP API error code 10623.
DoAuthorization	AMT	Amount	Specify the error code to trigger as all digits in a number with two digits to the right of the decimal point; for example, 106.23 triggers PP API error code 10623.
MassPay	EMAILSUBJECT	EmailSubject	Specify the error code to trigger as all digits in the field; for example, a subject of 10755 triggers PP API error code 10755.
BillUser	AMT	Amount	Specify the error code to trigger as all digits in a number with two digits to the right of the decimal point; for example, 107.55 triggers PP API error code 10755.
BAUpdate Version 2.4	MPID	MpID	Specify the error code to trigger as all digits in the field; for example, an ID of 10755 triggers PP API error code 10755.
BAUpdate Version 3.0	REFERENCEID	ReferenceID	Specify the error code to trigger as all digits in the field; for example, an ID of 10755 triggers PP API error code 10755.
AddressVerify	—	—	Not supported for negative testing.

NOTE: If the trigger value is not a valid error code for the API being tested, positive testing occurs for the request, which might result in another error occuring.

Negative Testing Using an Amount-Related Trigger Field

Consider an example that sets up testing for error 10623 for `DoAuthorization`, in which the error code is specified in the `AMT` field:

```
METHOD=DoAuthorization
&TRANSACTIONID=O-1GU0288989807143B&
AMT=106.23&
TRANSACTIONENTITY=Order&
VERSION=2.4&
USER=username&
PWD=password&
SIGNATURE=signature
```

The request invokes the following response:

```
TIMESTAMP=2007%2d04%2d04T03%3a10%3a19Z&
CORRELATIONID=447d121150529&
ACK=Failure&
L_ERRORCODE0=10623&
L_SHORTMESSAGE0=Maximum%20number%20of%20authorization%20allowed%20for%20the
%20order%20is%20reached%2e&
L_LONGMESSAGE0=Maximum%20number%20of%20authorization%20allowed%20for%20the%
20order%20is%20reached%2e&
L_SEVERITYCODE0=Error&
VERSION=2%2e400000&
BUILD=1%2e0006
```

Negative Testing Using a Non-Amount Trigger Field

Consider an example that sets up testing for error 10603 for `DoVoid`, in which the error code is specified in the `AUTHORIZATIONID` field:

```
METHOD=DoVoid&
AUTHORIZATIONID=10603&
VERSION=2.4&
USER=username&
PWD=password&
SIGNATURE=signature
```

The request invokes the following response:

```
TIMESTAMP=2007%2d04%2d04T03%3a10%3a22Z&
CORRELATIONID=51b0c5054dee6&
ACK=Failure&
L_ERRORCODE0=10603&
L_SHORTMESSAGE0=The%20buyer%20is%20restricted%2e&
L_LONGMESSAGE0=The%20buyer%20account%20is%20restricted%2e&
L_SEVERITYCODE0=Error
&VERSION=2%2e400000&
BUILD=1%2e0006
```

Negative Testing With Multiple Messages

Consider an example that sets up testing for error 10009 for `RefundTransaction`, which returns 14 possible error message sets:

```
METHOD=RefundTransaction&
TRANSACTIONID=asdf&
REFUNDTYPE=Partial&
AMT=100.09&
VERSION=2.4&
USER=username&
PWD=password&
SIGNATURE=signature
```

The request invokes the following response:

```
TIMESTAMP=2007%2d04%2d04T03%3a10%3a23Z&
CORRELATIONID=81ccc18eaec49&
ACK=Failure&
L_ERRORCODE0=10009&
L_SHORTMESSAGE0=Transaction%20refused&
L_LONGMESSAGE0=You%20can%20not%20refund%20this%20type%20of%20transaction&
L_SEVERITYCODE0=Error&
L_ERRORCODE1=10009&
L_SHORTMESSAGE1=Transaction%20refused&
L_LONGMESSAGE1=You%20are%20over%20the%20time%20limit%20to%20perform%20a%20r
efund%20on%20this%20transaction&
L_SEVERITYCODE1=Error&
L_ERRORCODE2=10009&
L_SHORTMESSAGE2=Transaction%20refused&
L_LONGMESSAGE2=Account%20is%20restricted&
L_SEVERITYCODE2=Error&
...
L_ERRORCODE13=10009&
L_SHORTMESSAGE13=Transaction%20refused&
L_LONGMESSAGE13=The%20partial%20refund%20amount%20must%20be%20less%20than%2
0or%20equal%20to%20the%20remaining%20amount&
L_SEVERITYCODE13=Error&
VERSION=2%2e400000&
BUILD=1%2e0006
```

Testing Using AVS Codes

You can simulate address verification by triggering an AVS error code when you call `DoDirectPayment` or use Virtual Terminal. To specify a code, place `AVS_`*code* in the NVP `SHIPTOSTREET` field or the `Street1` SOAP element when you call `DoDirectPayment`, where *code* is an AVS code, or enter `AVS_`*code* in Address Line 1 when using Virtual Terminal. For example, if you set `123 AVS_A Street` in the NVP `SHIPTOSTREET` field, AVS code `A` is set.

NOTE: `AVS_`*code* is case sensitive; all characters must be uppercase. For example, `AVS_A` is valid trigger; `avs_a` is not.

The following table identifies valid AVS codes, corresponding triggers, and a description of each error condition:

TABLE 6.2 AVS Error Conditions and Triggers

AVS Code	Trigger	Description of Error
A	AVS_A	The address matches but no zip code is specified; results in an error if the “Partial Address Match” risk control is set.
B	AVS_B	The international address matches but no zip code is specified; results in an error if the “Partial Address Match” risk control is set.
D	AVS_D	Exact match (no error). The international address and postal code matches.
F	AVS_F	Exact match (no error). The UK address and postal code matches.
P	AVS_P	The postal code matches but no address is specified; results in an error if the “Partial Address Match” risk control is set.
W	AVS_W	The 9-digit zip code matches but no address is specified; results in an error if the “Partial Address Match” risk control is set.
X	AVS_X	Exact match (no error). The complete address and 9-digit zip code matches.
Y	AVS_Y	Exact match (no error). The complete address and 5-digit zip code matches.
Z	AVS_Z	The 5-digit zip code matches but no address is specified; results in an error if the “Partial Address Match” risk control is set.
N	AVS_N	No address information; results in an error if the “No Address Match” risk control is set.
C	AVS_C	No address information for an international address; results in an error if the “No Address Match” risk control is set.
E	AVS_E	Not allowed for MOTO (internet/phone) transactions.
I	AVS_I	Service unavailable internationally; results in an error if the “Service Unavailable/Unsupported” risk control is set.
G	AVS_G	Service globally unavailable; results in an error if the “Service Unavailable/Unsupported” risk control is set.
R	AVS_R	Retry; results in an error if the “Service Unavailable/Unsupported” risk control is set.

***TABLE 6.2 AVS Error Conditions and Triggers** (Continued)*

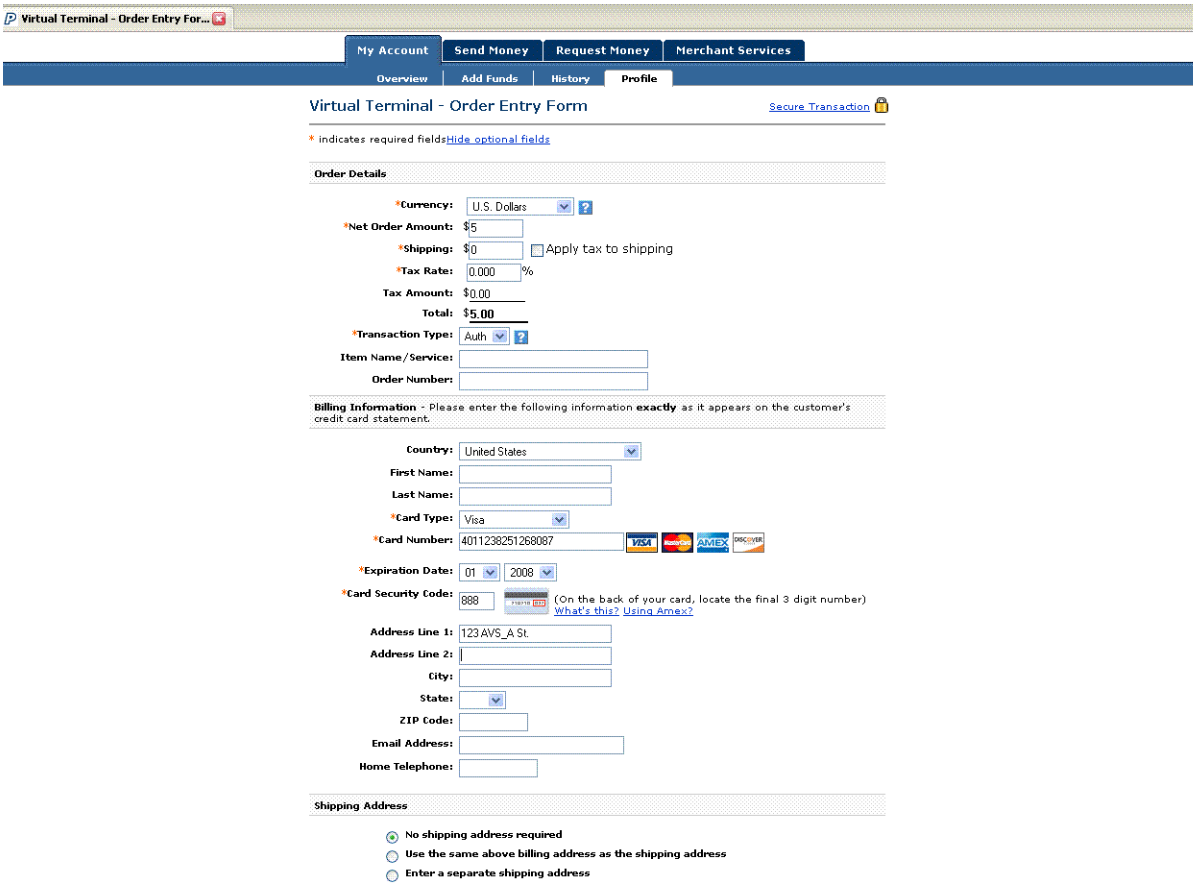
AVS Code	Trigger	Description of Error
S	AVS_S	Service not supported; results in an error if the "Service Unavailable/Unsupported" risk control is set.
U	AVS_U	Service unavailable; results in an error if the "Service Unavailable/Unsupported" risk control is set.

NOTE: The specified AVS code is set, regardless of whether a PP API error code is set. If no AVS code is specified or the AVS risk control is not specified, AVS code X is returned.

Testing an AVS Code Using Virtual Terminal

Consider an example of testing for AVS code A using Virtual Terminal. You enter AVS_A in the **Address Line 1** field:

FIGURE 6.1 Entering an AVS code in the Address Line 1 field



Virtual Terminal - Order Entry Form

* indicates required fieldsHide optional fields

Order Details

*Currency:	U.S. Dollars
*Net Order Amount:	$5
*Shipping:	$0 □ Apply tax to shipping
*Tax Rate:	0.000 %
Tax Amount:	$0.00
Total:	$5.00
*Transaction Type:	Auth
Item Name/Service:	
Order Number:	

Billing Information - Please enter the following information **exactly** as it appears on the customer's credit card statement.

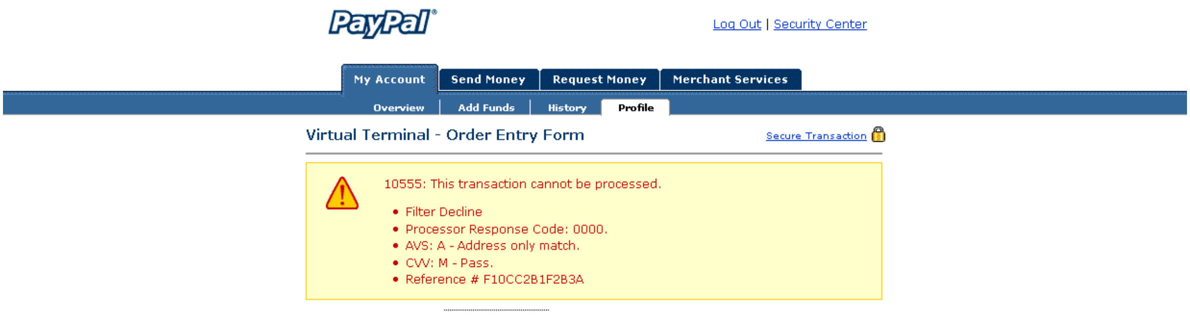
Country:	United States
First Name:	
Last Name:	
*Card Type:	Visa
*Card Number:	4011238251268087
*Expiration Date:	01 2008
*Card Security Code:	888 (On the back of your card, locate the final 3 digit number) What's this? Using Amex?
Address Line 1:	123 AVS_A St.
Address Line 2:	
City:	
State:	
ZIP Code:	
Email Address:	
Home Telephone:	

Shipping Address

● No shipping address required
○ Use the same above billing address as the shipping address
○ Enter a separate shipping address

When you attempt to process the transaction, the following message appears:

FIGURE 6.2 AVS error message from Virtual Terminal



Virtual Terminal - Order Entry Form

10555: This transaction cannot be processed.

- Filter Decline
- Processor Response Code: 0000.
- AVS: A - Address only match.
- CVV: M - Pass.
- Reference # F10CC2B1F2B3A

Testing an AVS Code Using DoDirectPayment

Consider an example that sets up testing for AVS code `A` and error code 10755 in `DoDirectPayment`, for which AVS code `A` indicates no zip code is specified and results in an error if the “Partial Address Match” risk control is set, whether or not other errors occur:

```
METHOD=DoDirectPayment&
CREDITCARDTYPE=VISA&
ACCT=4683075410516684&
EXPDATE=112007&
CVV2=808&
AMT=107.55&
FIRSTNAME=Designer&
LASTNAME=Fotos&
IPADDRESS=255.55.167.002&
STREET=1234%20AVS_A%20Street&
CITY=San%20Jose&
STATE=CA&
COUNTRY=United%20States&
ZIP=95110&
COUNTRYCODE=US&
SHIPTONAME=Louise%20P.%20Flowerchild&
SHIPTOSTREET=1234%20Easy%20Street&
SHIPTOSTREET2=Apt%2022%20bis&
SHIPTOCITY=New%20Orleans&
SHIPTOSTATE=LA&
SHIPTOCOUNTRY=US&
SHIPTOZIP=70114&
PAYMENTACTION=Authorization&
FIZBIN=foo&
VERSION=2.4&
USER=username&
PWD=password&
SIGNATURE=Aq9tJJ3ndj7r32JgX.qAzqOoC1JJAM7erWun-CUZYFDtxffpKWU4ERQG
```

The request invokes the following response:

```
TIMESTAMP=2007%2d04%2d04T03%3a35%3a10Z&
CORRELATIONID=a7cbf2d4d83dc&
ACK=Failure&
L_ERRORCODE0=10555&
L_SHORTMESSAGE0=Filter%20Decline&
L_LONGMESSAGE0=This%20transaction%20cannot%20be%20processed%2e&
L_SEVERITYCODE0=Error&
L_ERRORCODE1=10755&
L_SHORTMESSAGE1=Unsupported%20Currency%2e&
L_LONGMESSAGE1=This%20transaction%20cannot%20be%20processed%20due%20to%20an
%20unsupported%20currency%2e&
L_SEVERITYCODE1=Error&
VERSION=2%2e400000&
BUILD=1%2e0006
```

Testing Using CVV Codes

You can simulate credit card validation by triggering a CVV error code when you call `DoDirectPayment` or use Virtual Terminal. To specify a CVV code, place a trigger value in the NVP `CVV2` field or the `CVV2` SOAP element when you call `DoDirectPayment`, or enter `AVS_`*code* in **Card Security Code** when using Virtual Terminal.

The following table identifies valid CVV codes, corresponding triggers, and a description of each error condition:

TABLE 6.3 CVV Error Conditions and Triggers

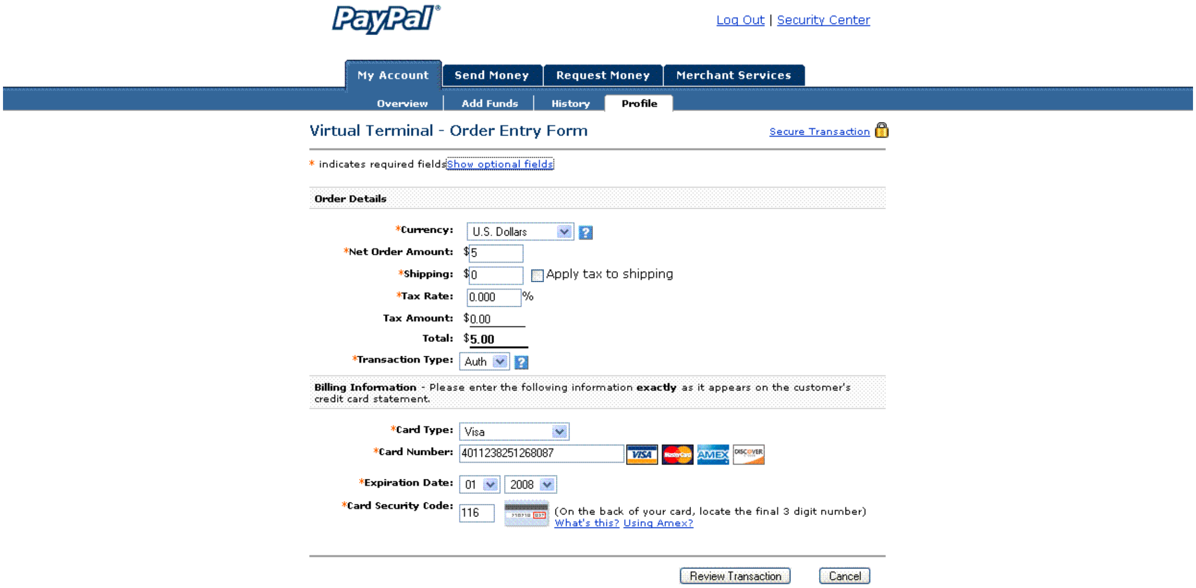
CVV Code	Trigger	Description of Error
M	115	CVV2 matches (no error).
N	116	CVV2 does not match.
U	125	Service unavailable.
S	123	Service not supported.
P	120	Transaction not processed.
X	130	No response.

NOTE: The specified CVV2 code is set, regardless of whether a PP API error code is set. If no CVV2 code is specified, `M` is returned.Virtual Terminal only displays the CVV2 error if the risk control blocks the payment.

Testing a CVV Code Using Virtual Terminal

Consider an example of testing for CVV code `N` using Virtual Terminal. You enter `116` in the **Card Security Code** field:

FIGURE 6.3 Entering a CVV code in the Card Security Code field



Virtual Terminal - Order Entry Form

* indicates required fields Show optional fields

Order Details

*Currency: U.S. Dollars

*Net Order Amount: $5

*Shipping: $0 □ Apply tax to shipping

*Tax Rate: 0.000 %

Tax Amount: $0.00

Total: $5.00

*Transaction Type: Auth

Billing Information - Please enter the following information **exactly** as it appears on the customer's credit card statement.

*Card Type: Visa

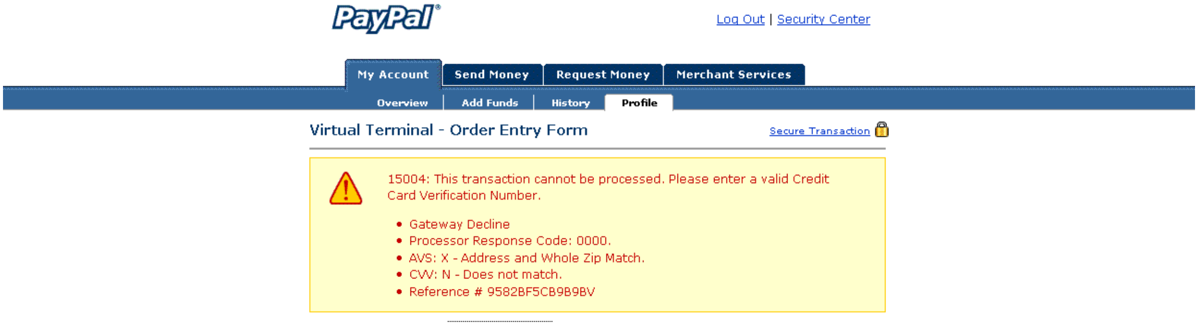
*Card Number: 4011238251268087

*Expiration Date: 01 2008

*Card Security Code: 116 (On the back of your card, locate the final 3 digit number) What's this? Using Amex?

When you attempt to process the transaction, the following message appears:

FIGURE 6.4 CVV error message from Virtual Terminal



Virtual Terminal - Order Entry Form

15004: This transaction cannot be processed. Please enter a valid Credit Card Verification Number.

- Gateway Decline
- Processor Response Code: 0000.
- AVS: X - Address and Whole Zip Match.
- CVV: N - Does not match.
- Reference # 9582BF5CB9B9BV

NOTE: Other errors are also reported in addition to CVV code `N`.

Testing a CVV Code Using DoDirectPayment

Consider an example that sets up testing for CCV code `N` in `DoDirectPayment`, which indicates a mismatch in the card validation code:

```
METHOD=DoDirectPayment&
CREDITCARDTYPE=VISA&
ACCT=4683075410516684&
EXPDATE=112007&
CVV2=116&
AMT=1.55&
FIRSTNAME=Designer&
LASTNAME=Fotos&
IPADDRESS=255.55.167.002&
STREET=1234%20Easy%20Street&
CITY=San%20Jose&
STATE=CA&
COUNTRY=United%20States&
ZIP=95110&
COUNTRYCODE=US&
SHIPTONAME=Louise%20P.%20Flowerchild&
SHIPTOSTREET=1234%20Easy%20Street&
SHIPTOSTREET2=Apt%2022%20bis&
SHIPTOCITY=New%20Orleans&
SHIPTOSTATE=LA&
SHIPTOCOUNTRY=US&
SHIPTOZIP=70114&
PAYMENTACTION=Authorization&
FIZBIN=foo&
VERSION=2.4&
USER=username&
PWD=password&
SIGNATURE=signature
```

The request invokes the following response:

```
TIMESTAMP=2007%2d04%2d04T03%3a35%3a12Z&
CORRELATIONID=2499856319532&
ACK=Failure&
L_ERRORCODE0=15004&
L_SHORTMESSAGE0=Gateway%20Decline&
L_LONGMESSAGE0=This%20transaction%20cannot%20be%20processed%2e%20Please%20e
nter%20a%20valid%20Credit%20Card%20Verification%20Number%2e&
L_SEVERITYCODE0=Error&
VERSION=2%2e400000&
BUILD=1%2e0006
```

7 Technical Support

Depending on the PayPal product you need assistance with, contact either Customer Service or Developer Technical Support.

Contacting Customer Service for Live PayPal Website Help

Use PayPal Help to find an answer to any problem you might encounter with live products such as Website Payments or Instant Payment Notification.

To contact Customer Service about issues with the Live PayPal website:

1. Go to https://www.paypal.com/
2. Click **Help** in the upper right corner of the page.
3. Click **Contact Us** in the lower left of the page.
4. Chose either **Help by Email** or **Help by Phone.**
5. Follow the remaining instructions.

Contacting Developer Technical Support for API Help

For information about PayPal Web Services API, Developer Central, and using the Sandbox, refer to the following resources:

- **Help Center**: In Developer Central, click **Help Center** to access developer manuals and links.
- **Forums**: In Developer Central, click **Forums** to share information with the PayPal developer community.
- **Online Developer Support Portal** at https://paypal.com/mts

Use the **Forums** first to find answers about any questions or problems you might have. Another developer might have already posted information about your question or problem.

To contact Developer Technical Support about the PayPal Web Services API:

1. Log in to your account at https://developer.paypal.com/ by entering your email address and password in the Member Log In box
2. Click **Help Center** at the bottom of the box on the right side of the page.
3. Click **Email PayPal Support**.

4. Complete the form.

Index