



Express Checkout Advanced Features Guide

PayPal Express Checkout Advanced Features Guide

Document Number: 10121.en_US-201204

© 2010-2012 PayPal, Inc. All rights reserved. PayPal is a registered trademark of PayPal, Inc. The PayPal logo is a trademark of PayPal, Inc. Other trademarks and brands are the property of their respective owners.
The information in this document belongs to PayPal, Inc. It may not be used, reproduced or disclosed without the written approval of PayPal, Inc. Copyright © PayPal. All rights reserved. PayPal S.à r.l. et Cie, S.C.A., Société en Commandite par Actions. Registered office: 22-24 Boulevard Royal, L-2449, Luxembourg, R.C.S. Luxembourg B 118 349
Consumer advisory: The PayPal™ payment service is regarded as a stored value facility under Singapore law. As such, it does not require the approval of the Monetary Authority of Singapore. You are advised to read the terms and conditions carefully.

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.



Contents

Preface	7
About This Guide	7
Intended Audience	7
Where to Go for More Information	7
Documentation Feedback	7
Chapter 1 Customizing Express Checkout	9
PayPal Review Page Order Details	9
Special Instructions to Merchant	11
Integrating Order Details into the Express Checkout Flow	11
Providing Gift Options	13
Obtaining Buyer Consent to Receive Promotional Email	15
Overriding Your Customer Service Number	16
Adding a Survey Question	17
PayPal Page Style	18
Custom Page Style	18
Individual Page Style Characteristics	19
Changing the Locale	20
Handling Shipping Addresses	22
Confirmed Address	23
Suppressing the Buyer's Shipping Address	23
Shipping Address Override	24
Automatically Filling Out Shipping and Contact Information	25
Buyer Pays on PayPal	28
Express Checkout Redirect to Let Buyers Pay on PayPal	28
Chapter 2 Express Checkout on Mobile Devices	29
About the Express Checkout Experience on Mobile Devices	29
Buyer Pays on Your Site	29
Buyer Pays on PayPal	30
Mobile Platforms Supported by Express Checkout	30
About Mobile Express Checkout Integration	31

Integrating Express Checkout With Your Mobile Website	31
Enabling PayPal Account Optional Checkout on Mobile Devices	34
Request Fields Supported by Express Checkout on Mobile Devices	36
NVP Request Fields Supported by Express Checkout on Mobile Devices	37
SOAP Request Fields Supported by Express Checkout on Mobile Devices	39
Locales Supported by Express Checkout on Mobile Devices	40
Locale Codes Not Supported by Express Checkout on Mobile Devices	40
Handling Locales Not Supported by Express Checkout on Mobile Devices.	40
Features Not Supported by Express Checkout on Mobile Devices	41
Chapter 3 Handling Recurring Payments	43
How Recurring Payments Work	43
Limitations	43
Recurring Payments Terms	44
Options for Creating a Recurring Payments Profile	45
Specifying the Regular Payment Period	45
Including an Optional Trial Period	46
Specifying an Initial Payment	46
Maximum Number of Failed Payments	46
Billing the Outstanding Amount	47
Identifying Items as Digital or Physical Goods.	47
Recurring Payments With the Express Checkout API	47
Initiating the Processing Flow With SetExpressCheckout	49
Redirecting the Buyer to PayPal.	50
Getting Buyer Details Using GetExpressCheckoutDetails.	51
Creating the Profiles With CreateRecurringPaymentsProfile	51
Recurring Payments Profile Status.	52
Getting Recurring Payments Profile Information.	53
Modifying a Recurring Payments Profile	53
Updating Addresses	54
Updating the Billing Amount	54
Billing the Outstanding Amount of a Profile	55
Recurring Payments Notifications	55
Chapter 4 Reference Transactions	57
Introduction to Reference Transactions	57
Reference Transactions	57
Billing Agreements.	58



API Operations for Reference Transactions	58
Setting Up a Billing Agreement Using the Express Checkout API	59
Initiating a Payment Using a Reference Transaction	60
About Cancelling Agreements and Getting the Billing Address	61
Canceling a Billing Agreement.	62
Obtaining the Most Recent Billing Address	62
Chapter 5 Implementing Parallel Payments	63
About Parallel Payments	63
What Is and What Is Not Supported	63
Post-Integration Experience	63
Name-Value Pair Syntax Supporting Parallel Payments	65
Integrating Parallel Payments by Using the NVP API	66
Integrating Parallel Payments by Using the SOAP API	68
Best Practices for Online Travel Agencies Implementing Parallel Payments	72
Styles of Payment	72
Payment Details	73
Handling Errors	74
Chapter 6 Integrating giropay with Express Checkout	75
giropay Page Flows	75
giropay Payment Page Flow	75
Canceled or Unsuccessful giropay Payment Page Flow	76
giropay Integration	76
Initiate the Flow with SetExpressCheckout	76
Redirect the Buyer to PayPal	76
Complete the Transaction	77
Receive Transaction Status Notification	77
Chapter 7 Implementing the Instant Update API	79
About the Instant Update API	79
Integration Steps.	79
Post-Integration Checkout Experience	80
How the Callback Works in the Express Checkout Flow.	82
Following Instant Update API Best Practices	83
Setting Up the Callback	84
GetExpressCheckoutDetails and DoExpressCheckoutPayment Changes	85

Other Callback Considerations	85
Using the Callback	86
SetExpressCheckout	86
Callback Request	88
Callback Response	89
Chapter 8 Payment Review91
Handling Payment Review	91
Chapter 9 Express Checkout Dynamic Image Integration95
Dynamic Images	95
Configuring the Dynamic Image	95
Set Up the Default Image	96
Set Up Image for Dynamic Use	96
Change the Locale.	97
Provide Incentive Eligibility Feedback to Buyer	97
Choose the Image	97
Dynamic Image Command Reference	98
Dynamic Image Parameters	98
Locale Codes and Priorities	98
Chapter 10 Immediate Payment	101
Overview of Immediate Payment	101
About Immediate Payment For Third-Party Checkout	101
Integrating Immediate Payment for Third-Party Checkout	103
The Call to SetExpressCheckout	103
The Call to DoExpressCheckoutPayment	104
About Immediate Payment For Express Checkout	104
Revision History	107



Preface

About This Guide

This document describes advanced features of Express Checkout integration.

Intended Audience

This document is for merchants and developers implementing Express Checkout who want to learn about its extra features, which can enhance their integrations with PayPal.

Where to Go for More Information

- [Express Checkout Integration Guide](#)
- [Name-Value Pair API Developer Guide](#)
- [SOAP API Developer Reference](#)
- [Merchant Setup and Administration Guide](#)

Documentation Feedback

Help us improve this guide by sending feedback to:
documentationfeedback@paypal.com



1

Customizing Express Checkout

You can specify options in Express Checkout API requests that change the appearance, behavior, and flow of the checkout process.

PayPal Review Page Order Details

When a buyer logs in to PayPal to check out, you can present the buyer with detailed information about each item being purchased. PayPal order details are available with API version 53.0 or later.

NOTE: The `DoExpressCheckoutPayment` request includes the same order details as `SetExpressCheckout`. PayPal recommends that you submit the same parameters in both API calls.

The following diagram shows all of the details that you can include:

Descriptions	Amount
<p>10% Decaf Kona Blend Coffee \$19.90</p> <p>Item description: Size: 8.8-oz</p> <p>Item number: 623083</p> <p>Item price: \$9.95</p> <p>Quantity: 2</p>	
<p>Coffee Filter Bags \$79.40</p> <p>Item description: Size: Two 24-piece...</p> <p>Item number: 623084</p> <p>Item price: \$39.70</p> <p>Quantity: 2</p>	
<hr/>	
Item total	\$99.30
Tax	\$2.58
Shipping and handling	\$5.99
Shipping discount	-\$3.00
Insurance	\$1.00
<hr/>	
Total \$105.87 USD	

(1) – Item name. The item name can identify this item to distinguish it from other line items in the order.

(2) – Item number. Each item can be further identified by an item number. If the item is an eBay auction item, it is recommended that you provide the eBay item number in this field.

(3) – Item description. This field identifies which of several items the buyer is purchasing. For example, you may be offering an item in different sizes. Knowing the size helps the buyer decide whether the one they selected was appropriate. If the item is an eBay auction item, it is recommended that you provide the phrase “eBay item” in this field.

(4) – Item unit price. This field specifies exactly how much one unit of the item costs. It can be a positive or negative value but not zero.

(5) – Item unit quantity. This field identifies the number of units the buyer is ordering.

PayPal calculates the value in the Amount (6) column as the product of line-item unit price and line-item unit quantity.

You can also show other detailed information about the order:

(7) – Item total and tax, which are the total of all items in the order and the tax, respectively.

(8) – Shipping and handling, which is the sum of the shipping and handling amounts.

NOTE: You must determine actual shipping and handling amounts.

(9) – Shipping discount. If the buyer is receiving a discount on shipping, the value appears as a credit in this field.

(10) – Insurance. This field shows the insurance fee when there is insurance on shipping.

(11) – Total. This is the total of the order, including shipping, handling, tax, and other price adjustment-related items.

NOTE: The **Enter gift certificate, reward, or discount** link enables the buyer to redeem certificates, rewards, or discounts that PayPal issues. The link does not enable the buyer to redeem incentives that you issue.

Special Instructions to Merchant

You can allow the buyer to send you special instructions about an order. This feature is especially helpful to buyers who want to customize merchandise. A buyer also might want to tell you to ship their order at a later date because they are out of the country.

NOTE: Users of this feature should be sure to read the instructions the buyer sends.

This feature appears as the link on the Review your information page. When the buyer clicks **Add**, a Note to seller text box opens in which the buyer can enter special instructions to the merchant and click **Save**. The instructions are returned in the responses to `GetExpressCheckoutDetails` and `DoExpressCheckoutPayment`.

Integrating Order Details into the Express Checkout Flow

To integrate order details into the checkout flow, pass any of the following Express Checkout parameters to `SetExpressCheckout`.

NVP Field	SOAP Field	Description and Comments
<code>L_PAYMENTREQUEST_n_NAMEm</code>	Name	Item name.
<code>L_PAYMENTREQUEST_n_NUMBERm</code>	Number	Item number.
<code>L_PAYMENTREQUEST_n_DESCm</code>	Description	Item description. The <code>DESC</code> (NVP) and <code>OrderDescription</code> (SOAP) fields still exist for backwards compatibility. However, <code>L_DESCn</code> and <code>Description</code> enable you to provide a more precise description for each different item purchased, such as hiking boots or cooking utensils rather than one general purpose description such as camping supplies.

NVP Field	SOAP Field	Description and Comments
L_PAYMENTREQUEST_n_AMT <i>m</i>	Amount	Item unit price. PayPal calculates the product of the item unit price and item unit quantity (below) in the Amount column of the cart review area. The item unit price can be a positive or a negative value, but not 0. You may provide a negative value to reflect a discount on an order, for example.
L_PAYMENTREQUEST_n_QTY <i>m</i>	Quantity	Item unit quantity.
ITEMAMT	ItemTotal	Sum of costs of all items in this order.
TAXAMT	TaxTotal	Sum of tax for all items in this order.
SHIPPINGAMT	ShippingTotal	Total shipping cost for this order (8). PayPal calculates the sum of the shipping cost and the handling cost. Although you may change the value later, try to pass in a shipping amount that is reasonably accurate.
PAYMENTREQUEST_n_HANDLINGAMT	HandlingTotal	Total handling cost for this order.
PAYMENTREQUEST_n_SHIPDISCAMT	ShippingDiscount	Shipping discount for this order. You specify this value as a negative number.
PAYMENTREQUEST_n_INSURANCEAMT	InsuranceTotal	Total shipping insurance cost for this order.
PAYMENTREQUEST_n_AMT	OrderTotal	Total of order, including shipping, handling, tax, and any other billing adjustments such as a credit due.

If you pass the generic order description parameter (PAYMENTREQUEST_n_DESC) along with any two of the following line-item parameters, the order description value does not display.

- L_PAYMENTREQUEST_n_NAME*m*
- L_PAYMENTREQUEST_n_NUMBER*m*
- L_PAYMENTREQUEST_n_DESC*m*

If you pass in unit price information (L_PAYMENTREQUEST_n_AMT*m*) without passing in the unit quantity (L_PAYMENTREQUEST_n_QTY*m*), the unit price does not display. To show both values, you must pass in values for both parameters. You can pass in a value of 1 even if the item purchase is uncountable.

The following example shows how to set line-item parameters in the call to SetExpressCheckout.

Request Parameters

[requiredSecurityParameters]

```
&METHOD=SetExpressCheckout
&RETURNURL=http://...
&CANCELURL=http://...
&PAYMENTREQUEST_0_PAYMENTACTION=Sale
&L_PAYMENTREQUEST_0_NAME0=10% Decaf Kona Blend Coffee
&L_PAYMENTREQUEST_0_NUMBER0=623083
&L_PAYMENTREQUEST_0_DESC0=Size: 8.8-oz
&L_PAYMENTREQUEST_0_AMT0=9.95
&L_PAYMENTREQUEST_0_QTY0=2
&L_PAYMENTREQUEST_0_NAME1=Coffee Filter bags
&L_PAYMENTREQUEST_0_NUMBER1=623084
&L_PAYMENTREQUEST_0_DESC1=Size: Two 24-piece boxes
&L_PAYMENTREQUEST_0_AMT1=39.70
&L_PAYMENTREQUEST_0_QTY1=2
&PAYMENTREQUEST_0_ITEMAMT=99.30
&PAYMENTREQUEST_0_TAXAMT=2.58
&PAYMENTREQUEST_0_SHIPPINGAMT=3.00
&PAYMENTREQUEST_0_HANDLINGAMT=2.99
&PAYMENTREQUEST_0_SHIPDISCAMT=-3.00
&PAYMENTREQUEST_0_INSURANCEAMT=1.00
&PAYMENTREQUEST_0_AMT=105.87
&PAYMENTREQUEST_0_CURRENCYCODE=USD
&ALLOWNOTE=1
```

Response Parameters

[successResponseFields]

```
&TOKEN=EC-17C76533PL706494P
```

Related information:

"Providing Gift Options" on page 13

"Following Instant Update API Best Practices" on page 83

"Setting Up the Callback" on page 84

Providing Gift Options

You can provide the buyer with gift options on PayPal. To use this feature, you must implement line-item details.

NOTE: Gift options are available with API Version 61.0 or later.

You can enable any of the following gift options:

- Gift message – This feature displays a text box in which the buyer can enter a gift message.
- Gift receipt – This feature provides a checkbox for the buyer to check if they would like a gift receipt included.

- Gift wrap – This feature provides a checkbox for the buyer to check if they would like to have the gift wrapped. The gift wrap feature can include a label describing the gift wrapping, for example, “Decorator box and bow.” Optionally, you can provide the amount to be charged to the buyer for gift wrapping.

The following `SetExpressCheckout` request example sets these options:

Request Parameters

[requiredSecurityParameters]

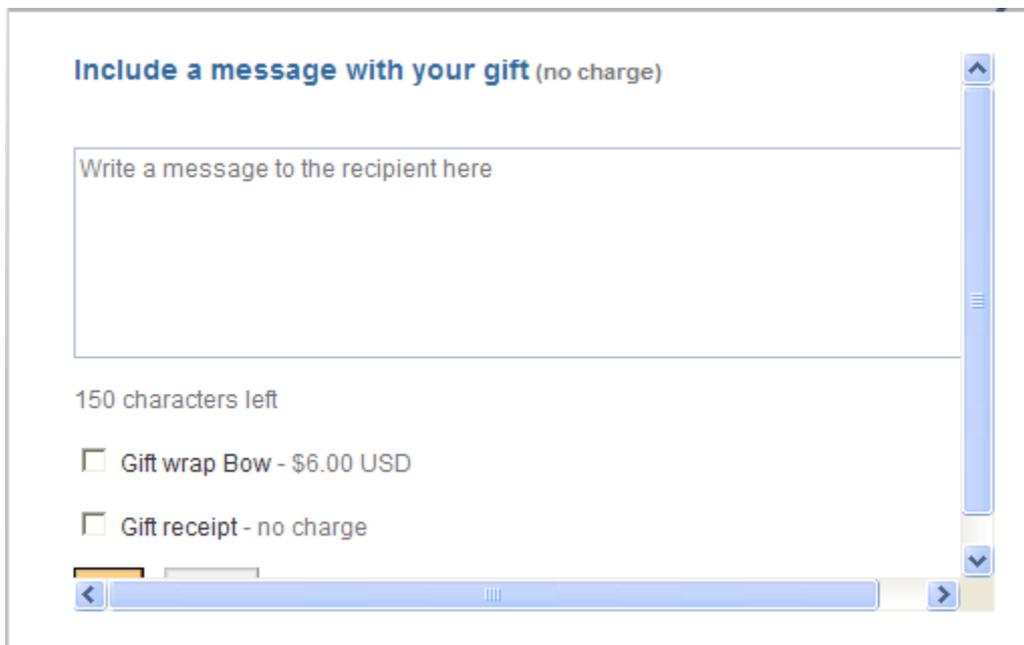
```
&METHOD=SetExpressCheckout
&RETURNURL=https://...
&CANCELURL=https://...
&PAYMENTREQUEST_0_AMT=10.00
&PAYMENTREQUEST_0_CURRENCYCODE=USD
&PAYMENTREQUEST_0_PAYMENTACTION=Sale
&GIFTMESSAGEENABLE=1
&GIFTRECEIPTENABLE=1
&GIFTWRAPENABLE=1
&GIFTWRAPNAME="Bow and Ribbon"
&GIFTWRAPAMOUNT=6.00
```

Response Parameters

[successResponseFields]

```
&TOKEN=EC-17C76533PL706495P
```

The following figure shows how the gift options appear to the buyer.



The screenshot shows a web interface for gift options. At the top, there is a heading "Include a message with your gift (no charge)". Below this is a text input field with the placeholder text "Write a message to the recipient here". Underneath the input field, it says "150 characters left". There are two checkboxes with labels: "Gift wrap Bow - \$6.00 USD" and "Gift receipt - no charge". The interface is contained within a scrollable area with a vertical scrollbar on the right and a horizontal scrollbar at the bottom.

NOTE: You can also configure this option from your profile. For details, see the [Merchant Setup and Administration Guide](#).

Related information:

"PayPal Review Page Order Details" on page 9

Obtaining Buyer Consent to Receive Promotional Email

You can obtain the buyer's consent to receive email promotions on PayPal pages. PayPal returns the email address in the response to `GetExpressCheckoutDetails` and `DoExpressCheckoutPayment`.

NOTE: Obtaining buyer consent to receive promotional email is available with API Version 61.0 or later.

To obtain the buyer's email address, set the `BUYEREMAILOPTINENABLE` field to 1 in the call to `SetExpressCheckout`.

The following request example sets this field:

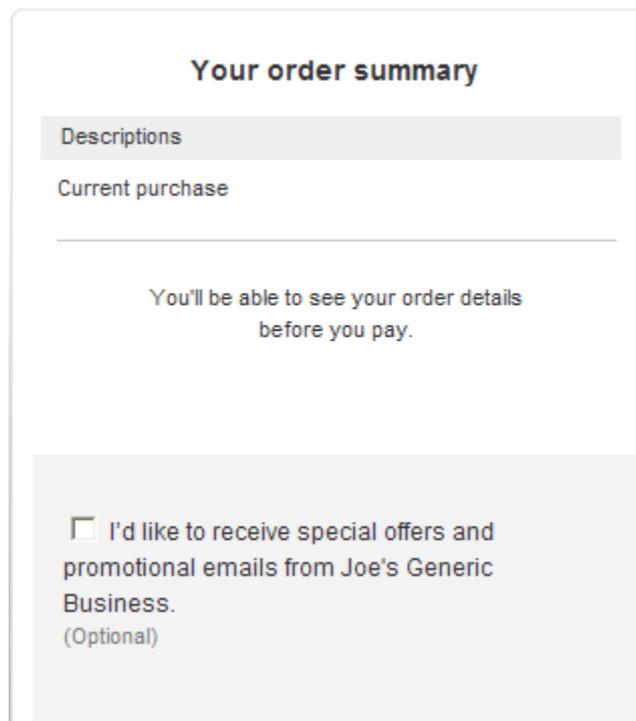
Request Parameters

```
[requiredSecurityParameters]
&METHOD=SetExpressCheckout
&RETURNURL=https://...
&CANCELURL=https://...
&PAYMENTREQUEST_0_AMT=10.00
&PAYMENTREQUEST_0_CURRENCYCODE=USD
&PAYMENTREQUEST_0_PAYMENTACTION=Sale
&BUYEREMAILOPTINENABLE=1
```

Response Parameters

```
[successResponseFields]
&TOKEN=EC-17C76533PL706495P
```

The following figure shows how this appears to the buyer in the cart review area.



The screenshot shows a 'Your order summary' page. At the top, it says 'Your order summary'. Below that is a section titled 'Descriptions' with a sub-section 'Current purchase'. A horizontal line separates this from the text 'You'll be able to see your order details before you pay.' Below that is a checkbox with the text 'I'd like to receive special offers and promotional emails from Joe's Generic Business. (Optional)'. The checkbox is currently unchecked.

NOTE: You can also configure this feature in your profile. For details, see the [Merchant Setup and Administration Guide](#).

Overriding Your Customer Service Number

You can display your Customer Service number to the buyer on Express Checkout pages by configuring it in your profile. You can override this number by specifying another number in the `SetExpressCheckout` request.

NOTE: This feature is available with API Version 61.0 or later.

Displaying your Customer Service number to the buyer enables you to quickly answer the buyer's questions through a telephone call. To override the Customer Service number configured in your profile with a different number on Express Checkout pages, set the `CUSTOMERSERVICENUMBER` field in the call to `SetExpressCheckout`.

The following request example sets this field:

Request Parameters

```
[requiredSecurityParameters]
&METHOD=SetExpressCheckout
&RETURNURL=https://...
&CANCELURL=https://...
&PAYMENTREQUEST_0_AMT=10.00
&PAYMENTREQUEST_0_CURRENCYCODE=USD
&PAYMENTREQUEST_0_PAYMENTACTION=Sale
&CUSTOMERSERVICENUMBER=1-800-FLOWERS
```

Response Parameters

```
[successResponseFields]
&TOKEN=EC-17C76533PL706494P
```

NOTE: For details on configuring the Customer Service number on the PayPal Profile page, see the [Merchant Setup and Administration Guide](#).

Adding a Survey Question

You can add one survey question to the PayPal pages. PayPal returns the choice that the buyer selected in the response to `GetExpressCheckoutDetails` and `DoExpressCheckoutPayment`.

NOTE: This feature is available with API Version 61.0 or later.

The survey question displays in the format of a text string on the PayPal **Review your information** page. The buyer responds by selecting from choices in a drop-down menu.

To enable the display of the survey question and choices, set the `SURVEYENABLE` field to 1 in the call to `SetExpressCheckout`.

- Set the `SURVEYENABLE` field to 1 in the call to `SetExpressCheckout`.
- Set `SURVEYQUESTION` to the string containing your question.
- Provide at least two `L_SURVEYCHOICE`n options from which the buyer can select one.

The following request example sets these fields:

Request Parameters

```
[requiredSecurityParameters]
&METHOD=SetExpressCheckout
&RETURNURL=https://...
&CANCELURL=https://...
&PAYMENTREQUEST_0_AMT=10.00
&PAYMENTREQUEST_0_CURRENCYCODE=USD
&PAYMENTREQUEST_0_PAYMENTACTION=Sale
&SURVEYENABLE=1
&SURVEYQUESTION="How did you hear about us?"
&L_SURVEYCHOICE0="friend"
&L_SURVEYCHOICE1="newspaper ad"
```

Response Parameters

[successResponseFields]

&TOKEN=EC-17C76533PL706495P

The survey question appears to the buyer in the cart review area.

NOTE: You can also configure this feature in your profile. For details, see the [Merchant Setup and Administration Guide](#).

PayPal Page Style

You can change the overall appearance of the PayPal pages by defining a custom page style or by customizing individual page style characteristics.

You define a custom page style in your profile and pass the resulting page style name when you call `SetExpressCheckout`. Typically you customize individual page style characteristics in your profile as well. However, you can also call `SetExpressCheckout` and pass in individual page characteristics dynamically.

Custom Page Style

When your buyer logs in to PayPal to check out, you can make the PayPal pages the buyer sees appear to have a similar look and feel to those on your website. You can customize any of these page characteristics and save the results as a Page Style Name. You can define up to three unique Page Style Names, in which you can specify the following characteristics:

- Header image
- Header border color
- Header background color
- Page background color

For instructions on how to customize page styles and create Page Style Names, see the [Merchant Setup and Administration Guide](#).

To set a custom page style in a call to `SetExpressCheckout`:

1. Include the optional `PAGESTYLE` parameter in the call to `SetExpressCheckout`.
2. Set `PAGESTYLE` to the Page Style Name you defined in your account.

The following example sets `PAGESTYLE` to the Page Style Name:

Request Parameters

```
[requiredSecurityParameters]
&METHOD=SetExpressCheckout
&RETURNURL=https://...
&CANCELURL=https://...
&PAYMENTREQUEST_0_AMT=10.00
&PAYMENTREQUEST_0_CURRENCYCODE=USD
&PAYMENTREQUEST_0_PAYMENTACTION=Sale
&PAGESTYLE=TestMerchant
```

Response Parameters

```
[successResponseFields]
&TOKEN=EC-17C76533PL706494P
```

Individual Page Style Characteristics

Typically, you create a custom page style for the PayPal pages using the custom payment pages in your Account Profile. In cases where you do not want to use the Account Profile option, you can specify these individual page style characteristics using variables in your program code:

- Logo image — a URL to an image of your logo.
- Cart review area border color — your principal identifying color. PayPal blends your color to white in a gradient fill that borders the cart review area.

To define your logo image:

1. Create a logo image up to 90 pixels wide by 60 pixels high and save it in a valid graphics format, such as .gif, .jpg, or .png.
2. Store the URL to the image on a secure (https) server so your buyer's web browser does not display a message that the payment contains insecure items.
3. Assign the URL to the LOGOIMG parameter in the call to `SetExpressCheckout`.

To display the border in your principal identifying color, set the `CARTBORDERCOLOR` parameter to the 6-digit hexadecimal value of that color in the call to `SetExpressCheckout`.

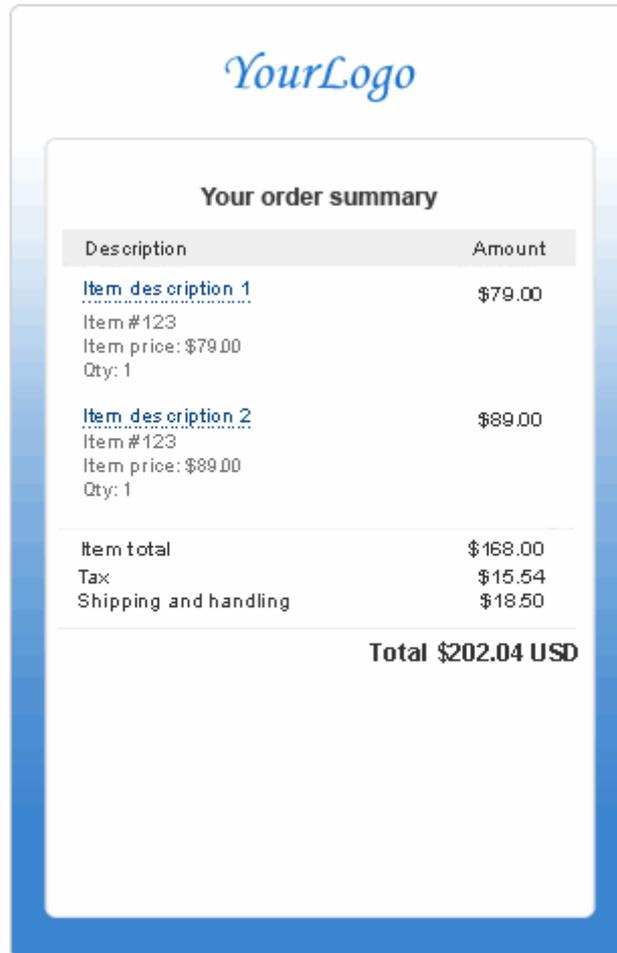
The following example sets a custom header image and adds a border color.

Request Parameters

```
[requiredSecurityParameters]
&METHOD=SetExpressCheckout
&RETURNURL=https://...
&CANCELURL=https://...
&PAYMENTREQUEST_0_AMT=10.00
&PAYMENTREQUEST_0_MAXAMT=...
&PAYMENTREQUEST_0_PAYMENTACTION=Sale
&LOGOIMG=https://www.../YourLogo.gif
&CARTBORDERCOLOR=0000CD
```

Response Parameters*[successResponseFields]*`&TOKEN=EC-17C76533PL706494P`

The figure below shows the cart review area with a custom logo and border color around it that were set in the call to `SetExpressCheckout`.



Changing the Locale

You can change the locale of PayPal pages to match the language on your website.

To change the language displayed on the PayPal pages, set the `LOCALECODE` parameter to one of the following allowable values in the `SetExpressCheckout` call:

- AU – Australia
- AT – Austria
- BE – Belgium

- BR – Brazil
- CA – Canada
- CH – Switzerland
- CN – China
- DE – Germany
- ES – Spain
- GB – United Kingdom
- FR – France
- IT – Italy
- NL – Netherlands
- PL – Poland
- PT – Portugal
- RU – Russia
- US – United States
- The following 5-character codes are also supported for languages in specific countries:
 - da_DK – Danish (for Denmark only)
 - he_IL – Hebrew (all)
 - id_ID – Indonesian (for Indonesia only)
 - jp_JP – Japanese (for Japan only)
 - no_NO – Norwegian (for Norway only)
 - pt_BR – Brazilian Portuguese (for Portugal and Brazil only)
 - ru_RU – Russian (for Lithuania, Latvia, and Ukraine only)
 - sv_SE – Swedish (for Sweden only)
 - th_TH – Thai (for Thailand only)
 - tr_TR – Turkish (for Turkey only)
 - zh_CN – Simplified Chinese (for China only)
 - zh_HK – Traditional Chinese (for Hong Kong only)
 - zh_TW – Traditional Chinese (for Taiwan only)

The following example sets LOCALCODE to ES (Spain).

Request Parameters

```
[requiredSecurityParameters]
&METHOD=SetExpressCheckout
&RETURNURL=https://...
&CANCELURL=https://...
&PAYMENTREQUEST_0_AMT=10.00
&PAYMENTREQUEST_0_CURRENCYCODE=EUR
&PAYMENTREQUEST_0_PAYMENTACTION=Sale
&LOCALCODE=ES
```

Response Parameters

`[successResponseFields]`

`&TOKEN=EC-17C76533PL706494P`

The following figure shows the **Log in to PayPal** page when the `is` is set to ES:

The following figure shows the PayPal **Choose a way to pay** page when the `LOCALECODE` is set to ES:

Handling Shipping Addresses

You can specify several shipping address options that affect the PayPal pages.

In your `SetExpressCheckout` request, you can specify the following options:

- Require a confirmed address.
- Do not display the shipping address on the review page.
- Display an alternative address on the review page.

Confirmed Address

A *confirmed address* is a shipping address that PayPal has established as belonging to the PayPal account holder. To be protected by PayPal's Seller Protection Policy, you must require the shipping address to be a confirmed address.

NOTE: Because many buyers prefer to ship to a non-confirmed address (they may, for example, be shipping a gift to someone), PayPal does not recommend requiring a confirmed address unless you are selling high-risk merchandise. If you prefer that confirmed addresses be used, then do not set `ADDROVERRIDE`.

To require a confirmed address for the shipping address, ensure that the shipping address matches the address on record with PayPal. You can do this through your account profile, as described in the [Merchant Setup and Administration Guide](#). Alternately, you can set a flag in the call to `SetExpressCheckout`, as follows:

1. Include the optional `REQCONFIRMSHIPPING` parameter in the call to `SetExpressCheckout`.
2. Set `REQCONFIRMSHIPPING` to 1.

The following example shows how to require the shipping address to be a confirmed address.

NOTE: The value of `REQCONFIRMSHIPPING` overrides the setting in your Merchant Account Profile.

Request Parameters

```
[requiredSecurityParameters]
&METHOD=SetExpressCheckout
&RETURNURL=https://...
&CANCELURL=https://...
&PAYMENTREQUEST_0_AMT=10.00
&PAYMENTREQUEST_0_PAYMENTACTION=Sale
&REQCONFIRMSHIPPING=1
&PAYMENTREQUEST_0_SHIPTOSTREET=1 Main St
&PAYMENTREQUEST_0_SHIPTOCITY=San Jose
&PAYMENTREQUEST_0_SHIPTOSTATE=CA
&PAYMENTREQUEST_0_SHIPTOCOUNTRYCODE=US
&PAYMENTREQUEST_0_SHIPTOZIP=95131
&PAYMENTREQUEST_0_SHIPTOPHONENUM=408-967-4444
```

Response Parameters

```
[successResponseFields]
&TOKEN=EC-6UA07551EA393551U
```

Suppressing the Buyer's Shipping Address

You can suppress the display of the buyer's shipping address on the PayPal pages. You might want to do this in these cases:

- You are selling a product or service that does not require shipping.

- You prefer to handle addresses completely on your own and do not want to let buyers choose from their PayPal address book.

To suppress the display of the buyer's shipping address, set the `NOSHIPPING` parameter to 1 in the call to `SetExpressCheckout`. No shipping address displays on Express Checkout pages.

The following example suppresses the shipping address.

Request Parameters

[requiredSecurityParameters]

```
&METHOD=SetExpressCheckout
&RETURNURL=https://...
&CANCELURL=https://...
&PAYMENTREQUEST_0_AMT=10.00
&PAYMENTREQUEST_0_CURRENCYCODE=USD
&PAYMENTREQUEST_0_PAYMENTACTION=Sale
&NOSHIPPING=1
&PAYMENTREQUEST_0_SHIPTONAME=J Smith
&PAYMENTREQUEST_0_SHIPTOSTREET=1 Main St
&PAYMENTREQUEST_0_SHIPTOCITY=San Jose
&PAYMENTREQUEST_0_SHIPTOSTATE=CA
&PAYMENTREQUEST_0_SHIPTOCOUNTRYCODE=US
&PAYMENTREQUEST_0_SHIPTOZIP=95131
&PAYMENTREQUEST_0_SHIPTOPHONENUM=408-967-4444
```

Response Parameters

[successResponseFields]

```
&TOKEN=EC-17C76533PL706494P
```

The PayPal Review your information page does not display a shipping address when `NOSHIPPING` is set to 1.

Shipping Address Override

You can override the buyer's shipping address stored on PayPal. You would want to do this if, for example, your website registration already requested the buyer's shipping address.

Overriding the shipping address stored on PayPal replaces it with one you specify in the call to `SetExpressCheckout`. The buyer cannot edit the overridden address.

NOTE: If you prefer to override addresses, PayPal recommends that you do not require confirmed addresses, as described in [Confirmed Address](#).

To override the shipping address:

1. Set the `ADDROVERRIDE` parameter to 1 in the call to `SetExpressCheckout`.
2. Set the following shipping address parameters in the call to `SetExpressCheckout` to the address values you want to use for the new address.
 - `PAYMENTREQUEST_0_SHIPTONAME`
 - `PAYMENTREQUEST_0_SHIPTOSTREET`

- PAYMENTREQUEST_0_SHIPTOCITY
- PAYMENTREQUEST_0_SHIPTOSTATE (Optional)
- PAYMENTREQUEST_0_SHIPTOCOUNTRYCODE
- PAYMENTREQUEST_0_SHIPTOZIP
- PAYMENTREQUEST_0_SHIPTOSTREET2 (Optional)

The following example overrides the shipping address with the address values shown.

Request Parameters

[requiredSecurityParameters]

```
&METHOD=SetExpressCheckout
&RETURNURL=https://...
&CANCELURL=https://...
&PAYMENTREQUEST_0_AMT=10.00
&PAYMENTREQUEST_0_CURRENCYCODE=USD
&PAYMENTREQUEST_0_PAYMENTACTION=Sale
&ADDROVERRIDE=1
&PAYMENTREQUEST_0_SHIPTOSTREET=1 Second St
&PAYMENTREQUEST_0_SHIPTOSTREET2=Ste 210
&PAYMENTREQUEST_0_SHIPTOCITY=San Jose
&PAYMENTREQUEST_0_SHIPTOSTATE=CA
&PAYMENTREQUEST_0_SHIPTOCOUNTRYCODE=US
&PAYMENTREQUEST_0_SHIPTOZIP=95131
&PAYMENTREQUEST_0_SHIPTOPHONENUM=408-967-4444
```

Response Parameters

[successResponseFields]

```
&TOKEN=EC-57K68322WE343022B
```

The PayPal **Review your information** page shows the shipping address parameter values you specified in the SetExpressCheckout request.

Automatically Filling Out Shipping and Contact Information

PayPal can automatically fill out form fields for the buyer based on the buyer's shipping contact information passed in the call to SetExpressCheckout.

When you pass the buyer's shipping address, telephone number and email address in the call to SetExpressCheckout, PayPal automatically fills out this information in the debit or credit card form fields on the PayPal Choose a way to pay page.

After the call to SetExpressCheckout, the buyer is redirected to the PayPal. On the **Choose a way to pay page**, buyers having a PayPal account can log in with their email address and password. Buyers who do not have an account can use their debit or credit card to pay and will have their shipping and contact information filled out.

See the following SetExpressCheckout example:

Request Parameters*[requiredSecurityParameters]*

```
&METHOD=SetExpressCheckout
&RETURNURL=https://...
&CANCELURL=https://...
&PAYMENTREQUEST_0_AMT=10.00
&PAYMENTREQUEST_0_PAYMENTACTION=Sale
&PAYMENTREQUEST_0_SHIPTOSTREET=1 Main Street
&PAYMENTREQUEST_0_SHIPTOCITY=San Jose
&PAYMENTREQUEST_0_SHIPTOSTATE=CA
&PAYMENTREQUEST_0_SHIPTOCOUNTRYCODE=US
&PAYMENTREQUEST_0_SHIPTOZIP=95131
&PAYMENTREQUEST_0_EMAIL=jsmith01@example.com
&PAYMENTREQUEST_0_SHIPTOPHONENUM=408-559-5948
```

Response Parameters*[successResponseFields]*

```
&TOKEN=EC-6UA07551EA393551U
```

The figure below shows the Pay with debit or credit card section expanded with the buyer shipping and contact fields filled out.

▶ **Pay with my PayPal account**

▼ **Pay with a debit or credit card**
If you don't have a PayPal account

Country

Credit card number

Payment types    

Expiration date mm / yy

CSC
[What is this?](#)

Billing information J Smith
1 Main St
San Jose, CA 95131
United States
[Change](#)

Contact information 408-559-5948
jsmith01@example.com
[Change](#)

▶ Save your information with PayPal

This transaction will appear on your statement as JOESGENERIC.

[Review and Continue](#)

Secure payments by **PayPal**

[Cancel and return to Joe's Generic Business.](#)

Buyer Pays on PayPal

With Express Checkout, you can shorten your checkout flow to let buyers complete their purchases on PayPal. Then, you can skip your order confirmation page.

Generally, buyers select payment methods as the last step before they complete their purchases. If you collect no additional information after buyers return from PayPal, you can skip the confirm-order page on your website. If you collect additional information that does not affect the payment, PayPal recommends that you collect it *after* buyers complete their purchases.

The `useraction` URL parameter in your redirect to PayPal determines whether buyers complete their purchases on PayPal or on your website. If you set `useraction` to `commit`, PayPal sets the button text to **Pay Now** on the PayPal **Review your information** page. This text lets buyers know that they complete their purchases if they click the button.



After PayPal redirects buyers to your site, call `GetExpressCheckoutDetails` and `DoExpressCheckoutPayment` to have PayPal complete the payment successfully. Call `DoExpressCheckoutPayment` without waiting for buyer interaction. Use information in the `GetExpressCheckoutDetails` response to fill out your order confirmation page.

Express Checkout Redirect to Let Buyers Pay on PayPal

The following sample code shows the redirect to let buyers pay on PayPal:

```
https://www.paypal.com/cgi-bin/webscr?cmd=_express-checkout&useraction=commit&token=valueFromSetExpressCheckoutResponse
```

2

Express Checkout on Mobile Devices

Express Checkout on mobile devices runs in mobile browsers with pages optimized for smaller mobile screens and mobile keyboards. If you have an Express Checkout implementation, you can take advantage of the mobile checkout experience with only minimal programming changes.

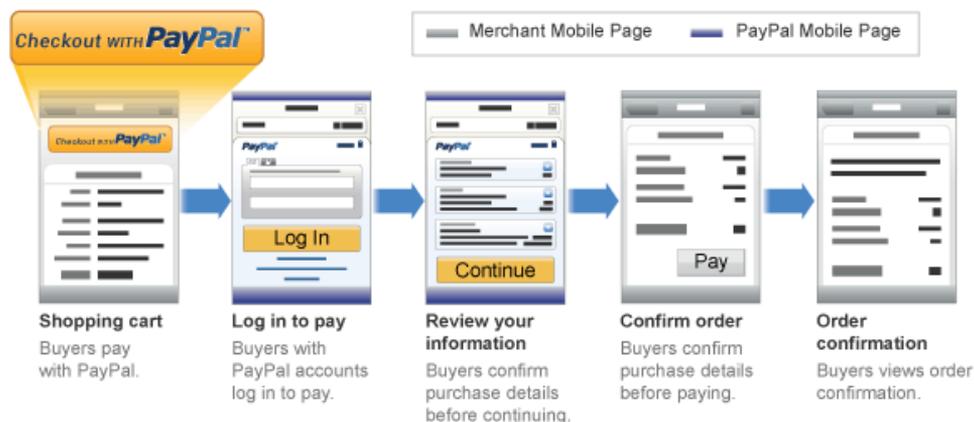
About the Express Checkout Experience on Mobile Devices

On mobile devices, Express Checkout provides payment pages tailored for faster checkout and for smaller mobile screens and keyboards. You can either set up the experience so that the buyer pays on your site or on PayPal

- [Buyer Pays on Your Site](#)
- [Buyer Pays on PayPal](#)

Buyer Pays on Your Site

The Express Checkout experience on mobile devices begins on your mobile website when a buyer is ready to pay you.



With Express Checkout on mobile devices, the buyer:

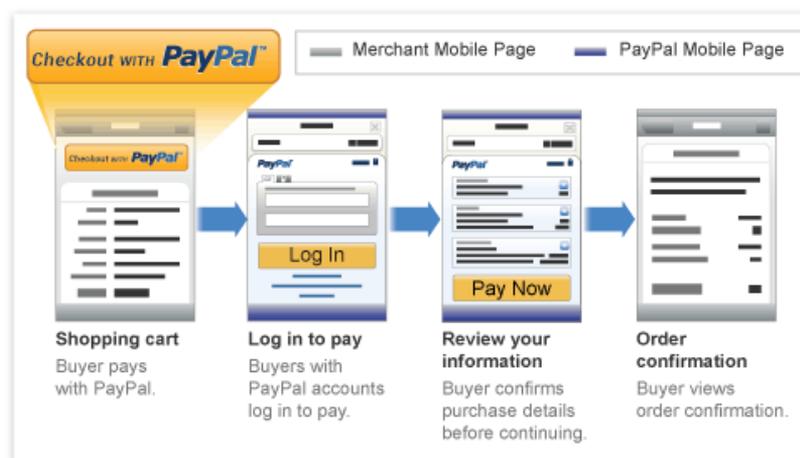
1. Clicks **Checkout with PayPal** on your mobile website or from your app.
2. Enters login credentials on the mobile PayPal login page, and then clicks **Log in**.
3. Reviews payment details on the mobile PayPal review page, and then clicks **Continue**.

4. Confirms the order and pays on your mobile website or app.
5. Views order confirmation on your mobile site or app.

NOTE: To pay with debit or credit cards, buyers click the “pay by card” at the bottom of the login page to enter their billing information; this flow is called *PayPal account optional* checkout.

Buyer Pays on PayPal

You can shorten the Express Checkout experience on mobile devices by letting buyers pay on PayPal.



When buyers pay on PayPal, the buyer:

1. Clicks **Checkout with PayPal** on your mobile website or from your app.
2. Enters login credentials on the mobile PayPal login page, and then clicks **Log in**.
3. Reviews payment details on the mobile PayPal review page, and then presses **Pay Now**.
4. Views order confirmation on your mobile site or app.

Related information:

"Enabling PayPal Account Optional Checkout on Mobile Devices" on page 34

Mobile Platforms Supported by Express Checkout

Express Checkout supports the mobile checkout experience on specific mobile devices and their embedded mobile browsers only.

Supported Mobile Devices	Software Versions
Android-based devices	All
iPhone® (2), 3G, 3GS, 4	iPhone OS 2.x, 3.x; iOS 4.0
iPod touch® 2G, 3G	iPhone OS 2.x, 3.x
iPad, iPad 2	All
BlackBerry (Research in Motion)	BlackBerry OS version 4.6 and higher

About Mobile Express Checkout Integration

PayPal supports several implementations of Mobile Express Checkout. You can provide a complete mobile website, or you can create a mobile phone app in which the checkout button is integrated into the app itself or is on your mobile website.

Your preferred integration determines what you must do to convert an existing Express Checkout integration to provide a mobile experience. Typically, your decision depends on whether you create a native app for the devices you want to support:

- If you are not providing an app, you must modify your pages to the recommended size for your device. When a buyer uses Express Checkout to check out, PayPal attempts to determine whether the buyer is using a mobile device, and the kind of device, and redirects the mobile browser to PayPal's mobile Express Checkout webpages. The call to `SetExpressCheckout` occurs as a result of the buyer clicking **Checkout with PayPal**.
- If you are providing an app, you must include the Mobile Express Checkout Library in your app and initialize it. For information about the Mobile Express Checkout Library, see the *Mobile Express Checkout Library Developer Guide and Reference* for your device's operating system: [iOS](#) or [Android](#). The library also provides additional functions that you can call from your app.

IMPORTANT: You cannot call `SetExpressCheckout` or any PayPal API directly from your app. If the page that hosts the **Pay with PayPal** button is on your app and not on a secure webpage on your server, you must make the call from your secure server and pass the response to your app; otherwise, the credentials used to make the call are not protected.

Integrating Express Checkout With Your Mobile Website

To integrate Express Checkout on you mobile website, specify return and cancel URLs on your mobile website in your call to `SetExpressCheckout` and specify the command in the

redirect as `cmd=_expresscheckout-mobile`. Not all features of Express Checkout are supported on mobile websites.

Your mobile website should conform to the standards of the browser in which it runs; for example, Safari on an iPad or Chrome on an Android phone. It must contain a page or pages to which PayPal redirects the buyer's browser when the payment flow completes or is canceled. You specify these pages in the `RETURNURL` and `CANCELURL` fields, respectively, of your call to `SetExpressCheckout`.

The following steps assume you already have a working Express Checkout integration that you want to "port" to your mobile website. If you are providing an app, you must include the Mobile Express Checkout Library in your app and initialize it. For information about the Mobile Express Checkout Library, see the *Mobile Express Checkout Library Developer Guide and Reference* for your device's operating system: [iOS](#) or [Android](#).

To support Mobile Express Checkout integration:

1. Change the command in your redirect to the Express Checkout flow to `_expresscheckout-mobile`.

If the buyer checks out on PayPal, your redirect will include:

```
https://www.paypal.com/cgi-bin/webscr?
cmd=_express-checkout-mobile
&token=valueFromSetExpressCheckoutResponse
```

If the buyer checks out on your website, also set `useraction=commit`:

```
https://www.paypal.com/cgi-bin/webscr?
cmd=_express-checkout-mobile
&useraction=commit
&token=valueFromSetExpressCheckoutResponse
```

NOTE: While it is not strictly required that you change the `cmd` value, PayPal recommends that you make this change to improve performance.

2. Change the `RETURNURL` field in your call to `SetExpressCheckout` to the page you want your buyer to return to on your mobile website.

```
RETURNURL=https://mobileWebsitePage.html
```

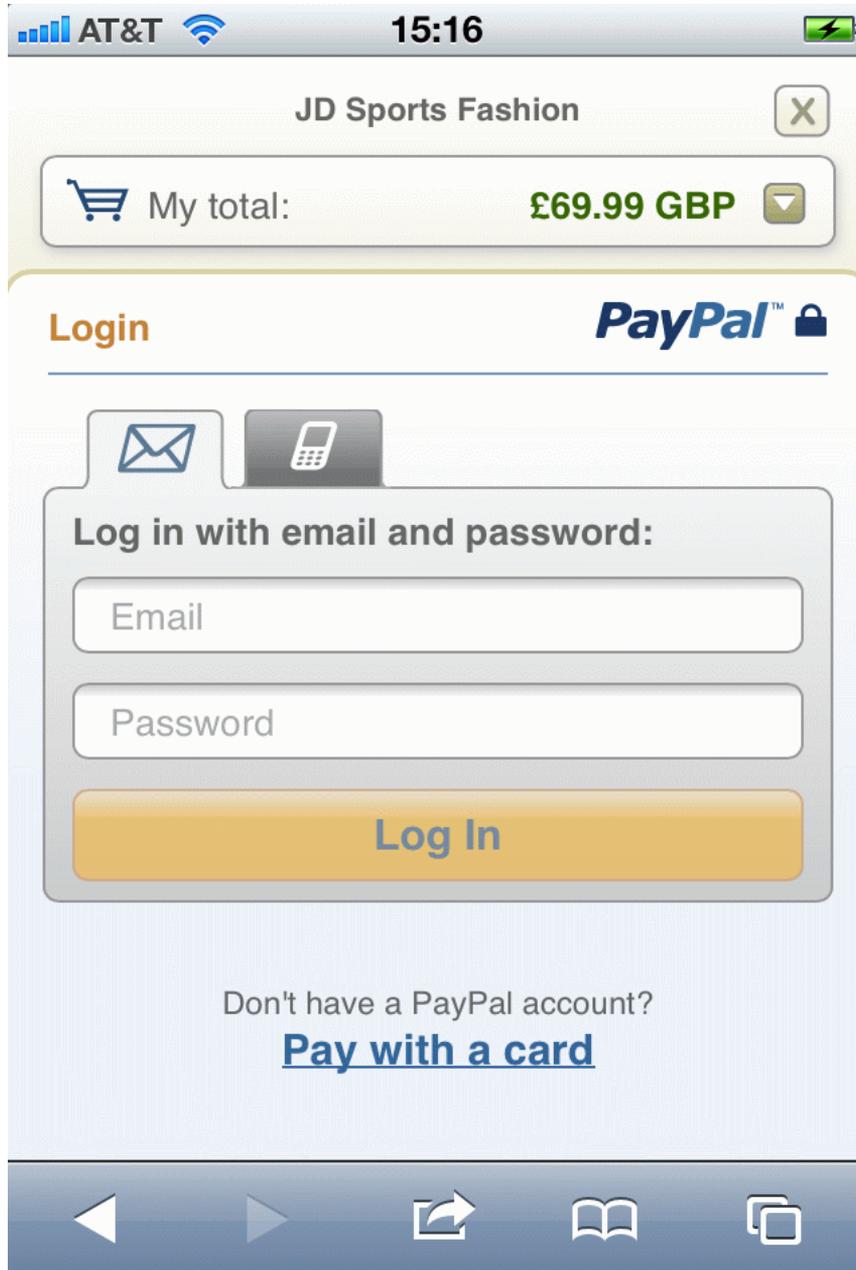
3. Change the `CANCELURL` field in your call to `SetExpressCheckout` to the page you want your buyer to return to if they cancel out of the Express Checkout payment flow.

```
CANCELURL=https://mobileWebsitePage.html
```

4. Remove any non-supported fields from your `SetExpressCheckout` and `DoExpressCheckoutPayment` calls. Include only fields for supported features.

Result:

When you redirect to PayPal and the buyer pays on PayPal, the following **Login** page appears on an iOS device:



When the buyer logs in to PayPal, the following **Review** page appears on an iOS device:

My total: \$19.95 USD

Review

Payment method:
Bank xx-3509 \$19.95 USD

Ship to:
Test User
1 Main St
San Jose, CA 95131
United States

Other info:

Please include information below:

Note to seller
Checkout using iPad

Pay Now

Related information:

"Features Not Supported by Express Checkout on Mobile Devices" on page 41

Enabling PayPal Account Optional Checkout on Mobile Devices

PayPal *account optional* checkout enables buyers to pay using debit and credit cards on your mobile website without logging in to PayPal, whether or not the buyer has a PayPal account. You cannot use this kind of checkout with recurring payments or reference transactions because these features require buyers to log in to PayPal to complete the initial payment.

To enable account optional checkout using Express Checkout, you must set the **PayPal Account Optional** preference to **On** in **Profile > Website Payment Preferences** after you log in to PayPal.

To enable account optional checkout, modify your SetExpressCheckout request to include the following information:

1. Specify that the buyer is not required to have a PayPal account:

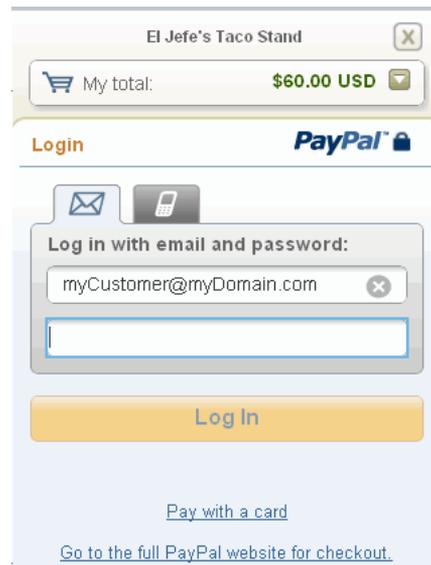
`SOLUTIONTYPE=Sole`

2. Specify whether the buyer lands on the checkout billing page or the PayPal mobile login page to complete the payment.

- `LANDINGPAGE=Billing` places the buyer on the billing page:

The screenshot shows a mobile checkout interface for 'El Jefe's Taco Stand'. At the top, there is a shopping cart icon and 'My total: \$60.00 USD'. Below this is a 'Pay with a card' section, powered by PayPal. The form includes a dropdown menu for 'United States', a 'Card number' input field, a 'Card type' selection with options for VISA, MasterCard, and Discover, an 'Expiration' field with '01' and '2011' selected, a 'Security code' field with an information icon, a 'Zip code' field, and an 'Email address' field. A 'Phone number' field is partially visible at the bottom.

- `LANDINGPAGE=Login` places the buyer on the PayPal login page:



From the PayPal mobile login page, the buyer can choose to log in to PayPal, pay with a debit or credit card, or log in to PayPal using the non-mobile payment flow. The buyer enters debit or credit card information on the billing page after clicking **Pay with a card**.

3. Fill in all required fields and optional fields in the `SetExpressCheckout` request and call the `SetExpressCheckout` API operation.
4. Redirect your buyer's browser to PayPal's mobile endpoint.

Related information:

"About the Express Checkout Experience on Mobile Devices" on page 29

Request Fields Supported by Express Checkout on Mobile Devices

Express Checkout on mobile devices supports a subset of the API request fields that Express Checkout supports when it runs on personal computers. PayPal ignores API request fields for features that Express Checkout does not support when it runs on mobile devices.

NVP Request Fields Supported by Express Checkout on Mobile Devices

Express Checkout on mobile devices supports only these NVP fields:

NVP SetExpressCheckout Request Fields Supported on Mobile Devices

AMT	RETURNURL	CANCELURL	TOKEN
MAXAMT	DESC	CUSTOM	INVNUM
REQCONFIRMSHIPPING	NOSHIPPING	ADDROVERRIDE	LOCALECODE
			<p>The following values are not supported:</p> <ul style="list-style-type: none"> • AD – Andorra • AL – Albania • AM – Armenia • BR – Brazil • GE – Georgia • VA – Holy See • IN – India • MC – Monaco • MD – Moldova • UA – Ukraine
PAYMENTACTION	EMAIL	SOLUTIONTYPE value must be mark	CHANNELTYPE value must be merchant
BRANDNAME			

NVP Address Type Fields Supported by Express Checkout on Mobile Devices

STREET	STREET2	CITY
STATE	COUNTRYCODE	ZIP
SHIPTOPHONENUM		

Express Checkout on Mobile Devices

Request Fields Supported by Express Checkout on Mobile Devices

NVP Payment Details Type Fields Supported by Express Checkout on Mobile Devices

PAYMENTREQUEST_n_AMT	AMT (<i>deprecated</i>)	PAYMENTREQUEST_n_CURRENCYCODE
CURRENCYCODE (<i>deprecated</i>)	PAYMENTREQUEST_n_ITEMAMT	ITEMAMT (<i>deprecated</i>)
PAYMENTREQUEST_n_SHIPPINGAMT	SHIPPINGAMT (<i>deprecated</i>)	PAYMENTREQUEST_n_INSURANCEAMT
INSURANCEAMT (<i>deprecated</i>)	PAYMENTREQUEST_n_SHIPDISCAMT	SHIPPINGDISCAMT (<i>deprecated</i>)
PAYMENTREQUEST_n_HANDLINGAMT	HANDLINGAMT (<i>deprecated</i>)	PAYMENTREQUEST_n_TAXAMT
TAXAMT (<i>deprecated</i>)	PAYMENTREQUEST_n_DESC	DESC (<i>deprecated</i>)
PAYMENTREQUEST_n_CUSTOM	CUSTOM (<i>deprecated</i>)	PAYMENTREQUEST_n_INVNUM
INVNUM (<i>deprecated</i>)	BUTTONSOURCE	PAYMENTREQUEST_n_NOTIFYURL
NOTIFYURL (<i>deprecated</i>)	PAYMENTREQUEST_n_NOTETEXT	NOTETEXT (<i>deprecated</i>)
PAYMENTREQUEST_n_TRANSACTIONID	TRANSACTIONID (<i>deprecated</i>)	PAYMENTREQUEST_n_ALLOWEDPAYMENTMETHOD
ALLOWEDPAYMENTMETHOD (<i>deprecated</i>)	PAYMENTREQUEST_n_PAYMENTACTION	PAYMENTACTION (<i>deprecated</i>)
PAYMENTREQUEST_n_PAYMENTREQUESTID	PAYMENTREQUESTID (<i>deprecated</i>)	

NVP Payment Details Item Type Fields Supported by Express Checkout on Mobile Devices

L_PAYMENTREQUEST_n_NAME <i>m</i>	L_NAME <i>n</i> (<i>deprecated</i>)	L_PAYMENTREQUEST_n_DESC <i>m</i>
L_DESC <i>n</i> (<i>deprecated</i>)	L_PAYMENTREQUEST_n_AMT <i>m</i>	L_AMT <i>n</i> (<i>deprecated</i>)
L_PAYMENTREQUEST_n_NUMBER <i>m</i>	L_NUMBER <i>n</i> (<i>deprecated</i>)	L_PAYMENTREQUEST_n_QTY <i>m</i>
L_QTY <i>n</i> (<i>deprecated</i>)	L_PAYMENTREQUEST_n_TAXAMT <i>m</i>	L_TAXAMT <i>n</i> (<i>deprecated</i>)
L_PAYMENTREQUEST_n_ITEMWEIGHTVALUE <i>m</i> and L_PAYMENTREQUEST_n_ITEMWEIGHTUNIT <i>m</i>	L_ITEMWEIGHTVALUE <i>n</i> and L_ITEMWEIGHTUNIT <i>n</i> (<i>deprecated</i>)	L_PAYMENTREQUEST_n_ITEMURL <i>m</i>
L_ITEMURL <i>n</i> (<i>deprecated</i>)		

SOAP Request Fields Supported by Express Checkout on Mobile Devices

Express Checkout on mobile devices supports only these SOAP fields:

SOAP SetExpressCheckout Request Fields Supported on Mobile Devices

OrderTotal	ReturnURL	CancelURL	Token
MaxAmount	OrderDescription	Custom	InvoiceID
Address	ReqConfirmShipping	NoShipping	AddressOverride
LocaleCode	PaymentAction	BuyerEmail	SolutionType
The following values are not supported:			value must be mark
<ul style="list-style-type: none"> • AD – Andorra • AL – Albania • AM – Armenia • BR – Brazil • GE – Georgia • VA – Holy See • IN – India • MC – Monaco • MD – Moldova • UA – Ukraine 			
ChannelType	BrandName		
value must be merchant			

SOAP AddressType Fields Supported by Express Checkout on Mobile Devices

Street1	Street2	CityName
StateOrProvince	Country	PostalCode
Phone		

SOAP PaymentDetailsType Fields Supported by Express Checkout on Mobile Devices

OrderTotal	ItemTotal	ShippingTotal
InsuranceTotal	ShippingDiscount	HandlingTotal
TaxTotal	OrderDescription	Custom
InvoiceID	ButtonSource	NotifyURL
ShipToAddress	PaymentDetailsItem	NoteText
TransactionId	AllowedPaymentMethodType	PaymentAction
PaymentRequestID		

SOAP PaymentDetailsItemType Fields Supported by Express Checkout on Mobile Devices

Name	Description	Amount
Number	Quantity	Tax
ItemWeight	ItemURL	

Locales Supported by Express Checkout on Mobile Devices

You can set the locale for Express Checkout pages on mobile devices to any of the countries generally supported by PayPal, with a few exceptions. PayPal displays the **Login** page in the default language for the country that you specify. The default locale is United States.

Locale Codes Not Supported by Express Checkout on Mobile Devices

The following locale codes are not supported by Express Checkout on mobile devices:

Country or Region	Code
Albania	AL
Andorra	AD
Armenia	AM
Brazil	BR
Georgia	GE
Holy See (Vatican City State)	VA
India	IN
Moldova, Republic of	MD
Monaco	MC
Ukraine	UA

Handling Locales Not Supported by Express Checkout on Mobile Devices

If you attempt use use an unsupported locale, the following actions occur:

- **Merchants** – If you set the locale code to an unsupported country, PayPal switches buyers to the full website checkout experience.
- **Buyers** – If a buyer logs in with a PayPal account for an unsupported country, PayPal stops the mobile checkout experience and the buyer cannot pay.

Features Not Supported by Express Checkout on Mobile Devices

Before implementing Express Checkout on mobile devices, examine the list of features that are not supported.

Express Checkout **does not support** the following features when it runs on mobile devices:

- **SMS security keys for mobile PayPal login**, in which buyers can sign up for an extra layer of security with a secure token. Buyers add a one-time secure code to their password or mobile PIN when they log in to PayPal. Buyers can use secure codes from hardware tokens but not from software tokens sent by SMS to the buyer's mobile phone.
- **Forgotten email or password**, in which buyers can request their email address or password during checkout, before logging in to PayPal.
- **Parallel payments**, which lets buyers pay multiple merchants in a single checkout session.
- **Dynamic images**, which lets PayPal update the images on your website for the **Checkout With Express Checkout** button and PayPal acceptance mark automatically to coincide with PayPal campaigns.
- **Buyer experience enhancements**, which lets you offer gift wrap, ask a survey question, or display your customer service number during checkout.
- **Custom payment page styles**, which lets you change the overall appearance of the **Review your information** page with a custom page style or with individual page style characteristics.
- **Instant Update API**, which is a callback that sends you the buyer's shipping address during checkout on PayPal and lets you respond with your actual shipping charges and handling charges.
- **International addresses**, which lets buyers add international addresses during checkout after logging in to PayPal.
- **Promotional offers**, which lets buyers pay with coupons, incentives, and buyer credit.
- **Dynamic currency conversion**, which lets you list an item in one currency and then accept payment in a different currency.
- **Keep me logged in**, in which buyers remain logged in to PayPal between transactions.

Related information:

"Integrating Express Checkout With Your Mobile Website" on page 31

3

Handling Recurring Payments

Set up a recurring payment to handle subscription and other payments that occur on a fixed schedule.

- [Recurring Payments Demo](#)

How Recurring Payments Work

To view a video that demonstrates how to set up Recurring Payments, navigate to: [Recurring Payments Demo](#)

When you support recurring payments for a buyer, you create a *recurring payments profile*. The profile contains information about the recurring payments, including details for an optional trial period and a regular payment period. Both periods contain information about the payment frequency and payment amounts, including shipping and tax, if applicable.

After creating a profile, PayPal automatically queues payments based on the billing start date, billing frequency, and billing amount. Payments reoccur until the profile expires, there are too many failed payments to continue, or you cancel the profile.

NOTE: When using Express Checkout, the buyer can also cancel a recurring payments profile.

PayPal funds queued payments using the normal payment method hierarchy within the buyer's PayPal account.

After creating a recurring payments profile, you can view profile details or cancel the profile from your PayPal account. You can also access recurring payments reports from the PayPal **Business Overview** page.

Also, after creating a recurring payments profile, you can use the Recurring Payments API to do the following:

- Get information details about a recurring payments profile.
- Change the status of a recurring payments profile.
- Update the details of the recurring payments profile.
- Bill the outstanding amount of the recurring payments profile.

Limitations

The current release of the Recurring Payments API has the following limitations:

- A profile can have at most one optional trial period and a single regular payment period.
- The profile start date may not be earlier than the profile creation date.

Recurring Payments with Express Checkout also has the following limitations:

- To be able to create a recurring payments profile for the buyer, you must ensure that the buyer's PayPal account includes an active credit card.
- You can create a maximum of 10 recurring payments profiles during checkout.
- You can increase the profile amount by only 20% in each 180-day interval after you create the profile.

Recurring Payments Terms

The following table lists and describes terms that are commonly used in the context of PayPal recurring payments.

Recurring Payments Terms

Term	Definition
Recurring payments profile	Your record of a recurring transaction for a single buyer. The profile includes all information required to automatically bill the buyer a fixed amount of money at a fixed interval.
Billing cycle	<p>Make one payment per billing cycle. Each billing cycle has 2 components:</p> <ul style="list-style-type: none"> • The billing period specifies the unit to calculate the billing cycle (such as days or months). • The billing frequency specifies the number of billing periods that make up the billing cycle. <p>For example, if the billing period is Month and the billing frequency is 2, the billing cycle is 2 months. If the billing period is Week and the billing frequency is 6, PayPal schedules the payments every 6 weeks.</p>
Regular payment period	The main subscription period for this profile, which defines a payment amount for each billing cycle. The regular payment period begins after the trial period, if you specify a trial period for the profile.
Trial period	An optional subscription period before the regular payment period begins. A trial period may not have the same billing cycles and payment amounts as the regular payment period.
Payment amount	The amount the buyer pays for each billing cycle.
Outstanding balance	If a payment fails for any reason, PayPal adds that amount to the profile's outstanding balance.
Profile ID	An alphanumeric string (generated by PayPal) that uniquely identifies a recurring profile. You can specify the Profile ID in the <code>TransactionSearch</code> API operation to obtain all payments associated with the identified profile.

Options for Creating a Recurring Payments Profile

You can create a recurring payments profile that allows a regular payment period, an optional trial period, an initial payment, and other options.

Specifying the Regular Payment Period

Each recurring payments profile has a regular payment period that defines the amount and frequency of the payment. The following table lists the required fields for specifying the regular payment period.

Required fields for specifying a regular payment period

NVP	SOAP	Description
PROFILESTARTDATE	RecurringPaymentsProfileDetails.BillingStartDate	The date when billing for this profile begins. NOTE: The profile may take up to 24 hours for activation.
BILLINGPERIOD	ScheduleDetails.PaymentPeriod.BillingPeriod	The unit of measure for the billing cycle. Must be one of the following: <ul style="list-style-type: none"> • Day • Week • SemiMonth • Month • Year
BILLINGFREQUENCY	ScheduleDetails.PaymentPeriod.BillingFrequency	The number of billing periods that make up one billing cycle. The combination of billing frequency and billing period must be less than or equal to one year. NOTE: If the billing period is SemiMonth, the billing frequency must be 1.
AMT	ScheduleDetails.PaymentPeriod.Amount	Amount to bill for each billing cycle.

You can optionally include a value for TOTALBILLINGCYCLES (SOAP field ScheduleDetails.PaymentPeriod.TotalBillingCycles), which specifies the total number of billing cycles in the regular payment period. If you either do not specify a value or specify the value 0, the payments continue until PayPal (or the buyer) cancels or suspends the profile. If the value is greater than 0, the regular payment period continues for the specified number of billing cycles.

You can also specify an optional shipping amount or tax amount for the regular payment period.

Including an Optional Trial Period

You can optionally include a trial period in the profile by specifying the following fields in the `CreateRecurringPaymentsProfile` request. The following table lists the required fields for creating an optional trial period.

Required fields for specifying a trial period

NVP	SOAP
TRIALBILLINGPERIOD	<code>ScheduleDetails.TrialPeriod.BillingPeriod</code>
TRIALBILLINGFREQUENCY	<code>ScheduleDetails.TrialPeriod.BillingFrequency</code>
TRIALAMT	<code>ScheduleDetails.TrialPeriod.Amount</code>
TRIALTOTALBILLINGCYCLES	<code>ScheduleDetails.TrialPeriod.TotalBillingCycles</code>

Specifying an Initial Payment

You can optionally specify an initial non-recurring payment when the recurring payments profile is created by including the following fields in the `CreateRecurringPaymentsProfile` request:

Required fields for specifying an initial payment

NVP	SOAP
INITAMT	<code>ScheduleDetails.ActivationDetails.InitialAmount</code>
FAILEDINITAMTACTION	<code>ScheduleDetails.ActivationDetails.FailedInitAmountAction</code>

By default, PayPal does not activate the profile if the initial payment amount fails. To override this default behavior, set the `FAILEDINITAMTACTION` field to `ContinueOnFailure`. If the initial payment amount fails, `ContinueOnFailure` instructs PayPal to add the failed payment amount to the outstanding balance due on this recurring payment profile.

If you do not set `FAILEDINITAMTACTION` or set it to `CancelOnFailure`, PayPal creates the recurring payment profile. However, PayPal places the profile into a pending status until the initial payment completes. If the initial payment clears, PayPal notifies you by Instant Payment Notification (IPN) that it has activated the pending profile. If the payment fails, PayPal notifies you by IPN that it has canceled the pending profile.

If you created the profile using Express Checkout, the buyer receives an email stating that PayPal cleared the initial payment or canceled the pending profile.

Maximum Number of Failed Payments

By including the `MAXFAILEDPAYMENTS` field in the `CreateRecurringPaymentsProfile` request, you set the number of failed payments allowed before PayPal automatically suspends the profile. PayPal sends you an IPN message when the number of failed payments reaches the maximum number specified.

Billing the Outstanding Amount

If a payment fails for any reason, PayPal adds the billing amount (including shipping and tax, if applicable) to the profile's outstanding balance. Use the `AUTOBILLOUTAMT` field in the `CreateRecurringPaymentsProfile` request to specify whether PayPal should add the outstanding amount to the payment amount for the next billing cycle.

Whether or not you choose to include the outstanding amount with the payment for the next billing cycle, you can also use the `BillOutstandingAmount` API to programmatically collect that amount at any time.

Identifying Items as Digital or Physical Goods

Set all the payment details item fields in the following table in the `CreateRecurringPaymentsProfile` request. If all items are digital goods, be sure to set the item category field to `Digital` to get the discount rate for digital goods.

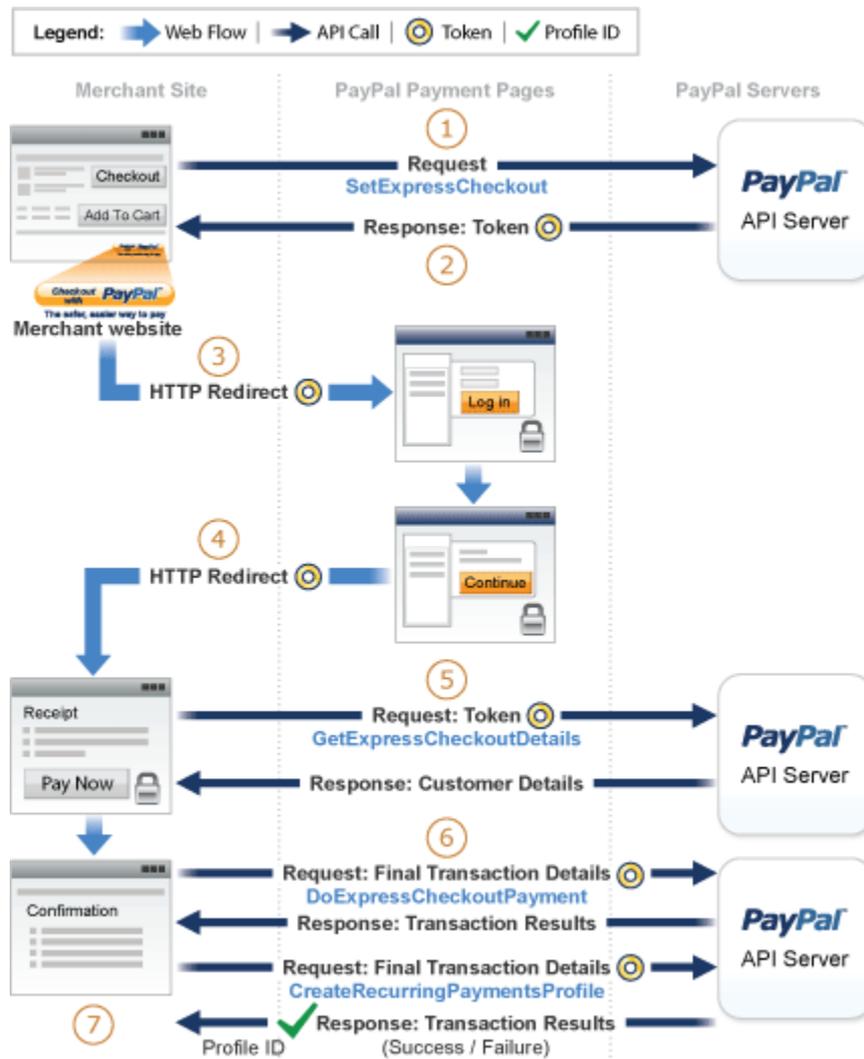
Required fields for specifying item details

NVP	SOAP	Description
<code>L_PAYMENTREQUEST_0_ITEMCATEGORYn</code>	<code>ItemCategory</code>	For digital goods, this field must be set to <code>Digital</code> . <ul style="list-style-type: none"> • Digital • Physical
<code>L_PAYMENTREQUEST_0_NAMEn</code>	<code>Name</code>	Item name.
<code>L_PAYMENTREQUEST_0_AMTn</code>	<code>Amount</code>	Cost of item.
<code>L_PAYMENTREQUEST_0_QTYn</code>	<code>Quantity</code>	Item quantity.

Recurring Payments With the Express Checkout API

During the checkout flow, you can create one or more recurring payments and mix recurring payments with other purchases.

The following diagram illustrates the typical processing flow to create recurring payments during checkout.



The circled numbers in the diagram correspond to the numbered steps in the following table:

Recurring payments processing flow

Step	Merchant...	PayPal...
1	Calls <code>SetExpressCheckout</code> , setting up one or more billing agreements in the request.	
2		Returns a token, which identifies the transaction, to the merchant.

Step	Merchant...	PayPal...
3	Redirects buyer's browser to: <code>https://www.paypal.com/cgi-bin/webscr?cmd=_express-checkout</code> <code>&token=<token returned by SetExpressCheckout ></code>	
		Displays login page and allows buyer to select payment options and shipping address.
4		Redirects buyer's browser to <code>returnURL</code> passed to <code>SetExpressCheckout</code> if buyer agrees to payment description.
5	Calls <code>GetExpressCheckoutDetails</code> to get buyer information (optional).	
		Returns <code>GetExpressCheckoutDetails</code> response.
	Displays merchant review page for buyer.	
6	Calls <code>DoExpressCheckoutPayment</code> if the order includes one-time purchases as well as a recurring payment. Otherwise, skip this step.	
		Returns <code>DoExpressCheckoutPayment</code> response
	Calls <code>CreateRecurringPaymentsProfile</code> one time for each recurring payment item included in the order.	
		Returns <code>ProfileID</code> in <code>CreateRecurringPaymentsProfile</code> response for each profile successfully created.
7	Displays successful transaction page.	

Initiating the Processing Flow With `SetExpressCheckout`

As in the Express Checkout flow, the `SetExpressCheckout` request notifies PayPal that you are:

- Initiating an order that can be either a one-time purchase, up to 10 recurring payments, or a mixture of a one-time purchase and recurring payments
- Initiating the processing flow to create one or more billing agreements for recurring payments with no associated one-time purchase or recurring payment

NOTE: You can also initiate the processing flow using `SetCustomerBillingAgreement` for orders that contain only a single recurring payment.

Typically, you set the amount of the payment for an Express Checkout transaction when you call the `SetExpressCheckout` request. You confirm the amount in the

DoExpressCheckoutPayment request. If, however, you set the amount of the payment to 0 in the SetExpressCheckout request, you can create a billing agreement without initiating a payment.

NOTE: To create a billing agreement without purchase, use Version 54.0 or higher, of the PayPal API.

To set up one or more billing agreements for recurring payments, modify the SetExpressCheckout request as follows:

1. Add an L_BILLINGTYPE n field for each billing agreement you want to create; n is a value in the range of 0 to 9, inclusive. Set the value of each field to RecurringPayments.

```
L_BILLINGTYPE0=RecurringPayments
```

2. Add an L_BILLINGAGREEMENTDESCRIPTION n field to correspond to each L_BILLINGTYPE n field you pass; n is a value in the range of 0 to 9, inclusive. Set the value of each field to the description of the goods or services associated with that billing agreement, for example:

```
L_BILLINGAGREEMENTDESCRIPTION0=Time Magazine subscription
```

3. If there is no associated purchase, set PAYMENTREQUEST_0_AMT to 0.

```
PAYMENTREQUEST_0_AMT=0
```

4. (Optional) Set MAXAMT to the average expected transaction amount.

PayPal uses the value you pass to validate the buyer's payment method for recurring payments. If you do not specify a value, the default is 25.00.

The SetExpressCheckout response provides a token that uniquely identifies the transaction for subsequent redirects and API calls.

Redirecting the Buyer to PayPal

After you receive a successful response from SetExpressCheckout, add the TOKEN from the SetExpressCheckout response as a name/value pair to the following URL, and redirect your buyer to it:

```
https://www.paypal.com/cgi-bin/webscr?cmd=_express-checkout&
token=<value_returned_by_SetExpressCheckoutResponse>
```

When redirecting the buyer to PayPal, PayPal recommends that you use the HTTPS response 302 "Object Moved" with the URL as the value of the Location header in the HTTPS response. Make sure that you use an SSL-enabled server to prevent browser warnings about a mix of secure and insecure graphics.

Getting Buyer Details Using GetExpressCheckoutDetails

The `GetExpressCheckoutDetails` method returns information about the buyer, including name and email address stored on PayPal. You can optionally call this API after PayPal redirects the buyer's browser to the `ReturnURL` you specified in the `SetExpressCheckout` request.

The `GetExpressCheckoutDetails` request has one required parameter, `TOKEN`, which is the value returned in the `SetExpressCheckout` response.

PayPal does not return the values you specified for the following parameter fields in the `GetExpressCheckoutDetails` response unless the transaction includes a purchase. PayPal ignores the fields if you set up a billing agreement for a recurring payment that is not immediately charged.

- `PAYMENTREQUEST_n_DESC`
- `PAYMENTREQUEST_n_CUSTOM`
- `PAYMENTREQUEST_n_INVNUM`

Creating the Profiles With CreateRecurringPaymentsProfile

After your buyer has agreed to the recurring payments billing agreement on your confirmation page, you must call `CreateRecurringPaymentsProfile` to create the profile. If you are creating multiple recurring payments profiles, you must call `CreateRecurringPaymentsProfile` once for each profile you plan to create.

If the transaction includes a mixture of a one-time purchase and recurring payments profiles, call `DoExpressCheckoutPayment` to complete the one-time purchase transaction. Then call `CreateRecurringPaymentsProfile` for each recurring payment profile you plan to create.

IMPORTANT: PayPal does not create the recurring payments profile until you receive a success response from the `CreateRecurringPaymentsProfile` call.

To obtain the best rates for digital goods, set values for the following required payment details item fields:

- `L_PAYMENTREQUEST_0_NAME`*n*
- `L_PAYMENTREQUEST_0_AMT`*n*
- `L_PAYMENTREQUEST_0_QTY`*n*
- `L_PAYMENTREQUEST_0_ITEMCATEGORY`*n* (you must set the value to `Digital`)

NOTE: The payment details item fields are available with version 69.0 and later of the `CreateRecurringPaymentsProfile` API.

The `CreateRecurringPaymentsProfile` response contains a Profile ID, which is an encoded string that uniquely identifies the recurring payments profile.

The following is a `CreateRecurringPaymentsProfile` example that enables you to obtain the best rates for digital goods items.

Request Parameters

```
[requiredSecurityParameters]
&METHOD=CreateRecurringPaymentsProfile
TOKEN=EC-13W99099UU817504D
&PROFILESTARTDATE:20XX-03-05T03:00:00
&DESC=Movie clips
&BILLINGPERIOD=Month
&BILLINGFREQUENCY=12
&AMT=1.00
&CURRENCYCODE=USD
&EMAIL=payername@bbb.net
&L_PAYMENTREQUEST_0_ITEMCATEGORY0=Digital
&L_PAYMENTREQUEST_0_NAME0=Kitty Antics
&L_PAYMENTREQUEST_0_AMT0=1.00
&L_PAYMENTREQUEST_0_QTY0=1
```

Response Parameters

```
[successResponseFields]
&PROFILEID=I-G7ALAX8095JY
&PROFILESTATUS=Active
```

Recurring Payments Profile Status

The recurring payments actions you may take depend on the status of the profile.

A recurring payments profile can have one of the following status values:

- `ActiveProfile`
- `PendingProfile`
- `ExpiredProfile`
- `SuspendedProfile`
- `CancelledProfile`

If PayPal successfully creates the profile, the profile has an `ActiveProfile` status. However, if a non-recurring initial payment fails and you set `FAILEDINITAMTACTION` to `CancelOnFailure` in the `CreateRecurringPaymentsProfile` request, PayPal creates the profile with a status of `PendingProfile`. The profile remains in this status until the initial payment either completes successfully or fails.

A profile has a status of `ExpiredProfile` when PayPal completes the total billing cycles for the optional trial and the regular payment periods.

You can suspend or cancel a profile by using the `ManageRecurringPaymentsProfileStatus` API. You can also reactivate a suspended profile. If PayPal has already reached the maximum number of failed payments, however, you must increase the number of failed payments before reactivating the profile.

NOTE: You can also suspend, cancel, or reactive a recurring payments profile through the PayPal website.

For recurring payments profiles created with the Express Checkout API, the buyer receives an email about the change in status of their recurring payment.

Getting Recurring Payments Profile Information

Use the `GetRecurringPaymentsProfileDetails` API to obtain information about a profile.

NOTE: You can also get information about recurring payments profiles from the PayPal website.

Along with the information that you specified in the `CreateRecurringPaymentsProfile` request, `GetRecurringPaymentsProfileDetails` also returns the following summary information about the profile:

- Profile status
- Next scheduled billing date
- Number of billing cycles completed in the active subscription period
- Number of billing cycles remaining in the active subscription period
- Current outstanding balance
- Total number of failed billing cycles
- Date of the last successful payment received
- Amount of the last successful payment received

Modifying a Recurring Payments Profile

Use the `UpdateRecurringPaymentsProfile` API to modify a recurring payments profile.

NOTE: You can also modify recurring payments profiles from the PayPal website.

You can modify only the following specific information about an active or suspended profile:

- Subscriber name or address
- Past due or outstanding amount
- Whether to bill the outstanding amount with the next billing cycle
- Maximum number of failed payments allowed
- Profile description and reference
- Number of additional billing cycles
- Billing amount, tax amount, or shipping amount

NOTE: You cannot modify the billing frequency or billing period of a profile. You can modify the number of billing cycles in the profile.

You can modify the following profile information during the trial period or regular payment period.

- Billing amount
- Number of billing cycles

NOTE: For recurring payments with the Express Checkout API, PayPal does not allow certain updates, such as billing amount, within 3 days of the scheduled billing date.

The profile changes take effect with the next payment after the call to update the profile. Say, for example, the buyer has made 1 trial payment out of a total of 3. You call `UpdateRecurringPaymentsProfile` to increase the number of billing cycles to 5. This provides the buyer with 4 additional trial payments. If you call `UpdateRecurringPaymentsProfile` during the regular payment period, the changes take effect with the buyer's next scheduled regular payment.

For complete details, see the *Name-Value Pair Developer Guide and Reference* or the *SOAP API Reference*.

Updating Addresses

When you update the subscriber shipping address, you must enter all of address fields, not just those that are changing:

To update the subscriber's street address, for example, specify all the address fields listed in the *Name-Value Pair Developer Guide and Reference* or *SOAP API Reference*. Do not specify only the street address field.

Updating the Billing Amount

For profiles created using Express Checkout, you can increase the recurring payment total amount by 20% maximum in a fixed 180-day interval after profile creation. The 20% maximum is based on the total amount of the profile at the beginning of the 180-day interval, including any shipping or tax amount.

If, for example, you create a profile on March 10 with a total amount of \$100, during the 180-day interval from March 10 to September 6, you can increase the amount to a maximum of \$120 (120% of \$100).

Suppose that during the first 180-day interval, you increased the payment amount to \$110. During the next 180-day interval (starting September 7), you can only increase the amount of the payment to a maximum of \$132 (120% of \$110).

Billing the Outstanding Amount of a Profile

Use the `BillOutstandingAmount` API to immediately bill the buyer for the current past due or outstanding amount for a recurring payments profile.

NOTE: You can also bill the buyer for the current past due or outstanding amount for a recurring payments profile from the PayPal website.

To bill the outstanding amount:

- The profile status must be active or suspended.

NOTE: The `BillOutstandingAmount` API does not reactivate a suspended profile. You need to call `ManageRecurringProfileStatus` to do this.

- The profile must have a non-zero outstanding balance.
- The amount of the payment cannot exceed the outstanding amount for the profile.
- The `BillOutstandingAmount` call cannot be within 24 hours of a regularly scheduled payment for this profile.

NOTE: An error occurs when another outstanding balance payment is already queued.

PayPal informs you by IPN about the success or failure of the outstanding payment. For profiles created using Express Checkout, the buyer receives an email notification of the payment.

Recurring Payments Notifications

PayPal notifies you of recurring payments events through IPN and email. Typically, however, you can call `GetTransactionDetails` to obtain the information you need.

PayPal notifies you of certain events through IPN. For recurring payments profiles created using Express Checkout, PayPal also notifies buyers of specific events by email. The following table indicates when PayPal generates IPN and emails:

Recurring payments IPN messages and email

Event	IPN	Buyer Email
Profile successfully created	Yes	Yes
Profile creation failed	Yes	Yes
Profile canceled from paypal.com interface	Yes	Yes
Profile status changed using API	No	Yes
Profile updated using API	No	Yes
Initial payment either succeeded or failed	Yes	Yes
Payment either succeeded or failed (during either trial period or regular payment period)	Yes	Yes
Outstanding payment either succeeded or failed	Yes	Yes
Maximum number of failed payments reached	Yes	No

NOTE: API transactions such as `ManageRecurringPaymentsProfileStatus` do not trigger IPN notification. The API response immediately provides the success or failure of the call.

4

Reference Transactions

A reference transaction is a financial transaction from which subsequent transactions can be derived. For example, a buyer can make a purchase on your site and the PayPal transaction ID becomes the ID of a reference transaction that can later be used to initiate another transaction.

Introduction to Reference Transactions

Recurring payments using reference transactions enables you to handle payments for varying amounts of money on a varying schedule. The buyer can sign up for recurring payments as part of the Express Checkout flow or sign up before making a purchase. Consider the following examples:

- A buyer purchases a mobile phone and selects a carrier. The buyer pays for the phone using Express Checkout and agrees to pay monthly for the service.
- A buyer downloads a music video and pays for it with Express Checkout. During checkout, the buyer agrees to make a payment whenever the account balance exceeds a specified minimum amount.
- A buyer agrees to pay monthly for utilities, such as gas or electricity. The buyer makes no payment at the time of agreement. PayPal deducts payments monthly for actual use.

These examples represent payment transactions that reoccur periodically, either for a specific time period or when transactions reach a certain threshold. The amount of each transaction varies by use. These kinds of recurring payments are well-suited for processing with reference transactions, which are used in the same way as reference transactions in credit card processing. When implementing recurring payments using reference transactions, you control transaction initiation and the transaction amount. You can also include payment for multiple items in one checkout.

NOTE: PayPal must enable you to use reference transactions. Contact PayPal for details.

Reference Transactions

A reference transaction is a financial transaction from which you can derive subsequent transactions. For example, a buyer makes a purchase on your site, and you use the PayPal transaction ID or *reference transaction ID* later to initiate another transaction. In the music video example, you might use the transaction generated by the purchase to initiate a monthly transaction for renting videos from the service. Similarly, you can use the transaction for the first download to generate another transaction for other downloads.

Billing Agreements

You must have the buyer's agreement to obtain funds from the buyer's account. The buyer must log in to PayPal once to agree, but after that, PayPal does not require the buyer to log in. This agreement represents a *billing agreement* between you and the buyer. PayPal maintains the agreement.

The buyer can agree during the Express Checkout flow, as in the case of purchasing a mobile phone purchase or downloading a music video. Or the buyer can agree before initiating a payment, as in the case of signing up for automatic utility payments.

PayPal creates a billing agreement ID that represents the agreement. The billing agreement ID is also associated with a transaction. You can use the billing agreement ID to derive a subsequent transaction in the same way that you use a reference transaction ID.

A reference transaction must have occurred within the past 365 days because the ID may not be available after a year. The billing agreement ID does not establish a time frame. It is good until canceled by the buyer. A buyer may have more than one billing agreement ID for your site. This could occur if the buyer established separate agreements for different kinds of service. If you use a reference transaction ID to initiate a reference transaction, you must make sure that the ID is associated with the correct billing agreement.

To avoid confusion, PayPal recommends that you use the billing agreement ID whenever it is available, instead of the reference transaction ID.

API Operations for Reference Transactions

Several API operations implement recurring payments using reference transactions:

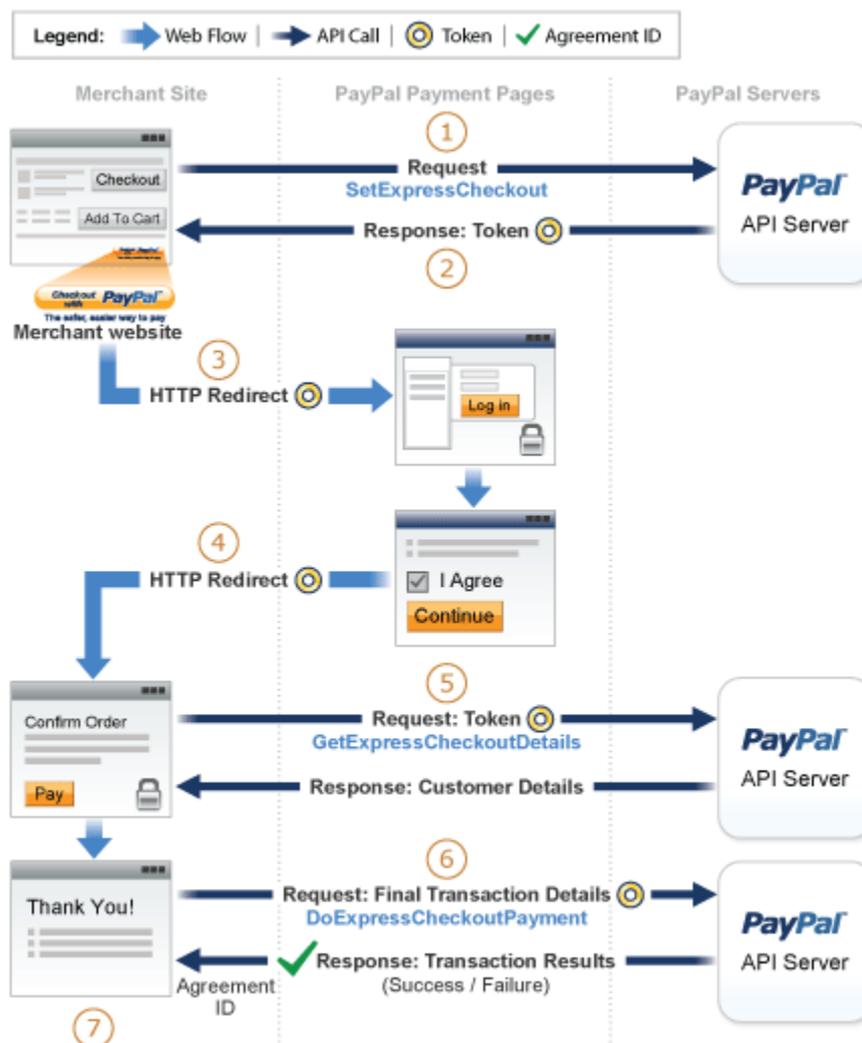
- Express Checkout API operations enable you to set up a billing agreement as part of the checkout processing, as in the examples of a mobile phone purchase or a music video download. These operations eliminate the need for a separate step to create the billing agreement.
- The `DoReferenceTransaction` API operation initiates a payment, which is sometimes called a *merchant pull* payment because you initiate the payment, not the buyer. After a billing agreement has been established, this is the only API operation you must call.
- The `CreateBillingAgreement` API operation enables you to set up a billing agreement without requiring a purchase. Use it in place of `DoExpressCheckoutPayment` to allow a buyer to set up a billing agreement before making a payment, as in the example of a buyer preparing to make a monthly utility payment.
- You can use the `BAUpdate` API (NVP: `METHOD=BillAgreementUpdate`) to enable a buyer to cancel a billing agreement while on your site. The buyer also can cancel a billing agreement by logging in to PayPal directly; thus, this API operation may not be required.
- You can use `BAUpdate` (NVP: `METHOD=BillAgreementUpdate`) to obtain the buyer's most recent billing address.

Setting Up a Billing Agreement Using the Express Checkout API

You must set up a billing agreement using the Express Checkout API before you can use a reference transaction.

Specify 0 as the amount of the purchase in `SetExpressCheckout` and call `CreateBillingAgreement` if you only want to obtain a billing agreement ID for later use. (Do not call `DoExpressCheckoutPayment`, because no payment is involved.) Otherwise, call `DoExpressCheckoutPayment` to obtain the billing agreement ID and to initiate an Express Checkout payment.

The following diagram shows the flow for obtaining a billing agreement ID during an Express Checkout payment:



The numbered steps correspond to the circled numbers. Your site must take the specified actions at each step, as follows:

1. Call `SetExpressCheckout` and pass a `BillingAgreementDetails` element that contains the following information:
 - A billing type. PayPal recommends that you use `MerchantInitiatedBillingSingleAgreement` if you need only one billing agreement between you and the buyer. Use `MerchantInitiatedBilling` to create a billing agreement between you and the buyer each time you call `DoExpressCheckoutPayment`.
 - A description of the goods or services associated with the agreement.
 - Optionally, the kind of PayPal payment you require.
 - Optionally, a string that you can use for any purpose.
2. PayPal returns a token, which you use in the subsequent steps.
3. Redirect the buyer's browser to PayPal, which allows the buyer to log in or set up an account to pay for the purchase. PayPal presents the buyer with a confirmation page.
4. PayPal redirects the buyer's browser to your return page.
5. Call `GetExpressCheckoutDetails` to obtain information about the buyer and the buyer's checkout status. PayPal returns checkout details.
6. Call `DoExpressCheckoutPayment` to initiate the purchase.
7. If the buyer accepts the billing agreement, PayPal returns information about the purchase and a billing agreement ID. Save the ID for reference transactions or recurring payments.

To obtain a billing agreement ID without also initiating a purchase, modify the previous procedure as follows:

- In step 1, specify 0 for the amount of purchase in the payment details.
- (Optional) Set `MAXAMT` to the expected average transaction amount (default is 25.00). PayPal uses the value to validate the buyer's payment method for future payments.
- Skip step 5.
- In step 6, call `CreateBillingAgreement` instead of `DoExpressCheckoutPayment`.

If you offer a store account, you can control whether PayPal creates the billing agreement based on buyer signup. To skip the creation of a billing agreement ID if the buyer does not sign up, set `SKIPBACREATION` to `true` in the call to `DoExpressCheckout`.

The numbered steps correspond to the circled numbers. Your site must take the specified actions at each step, as follows:

Initiating a Payment Using a Reference Transaction

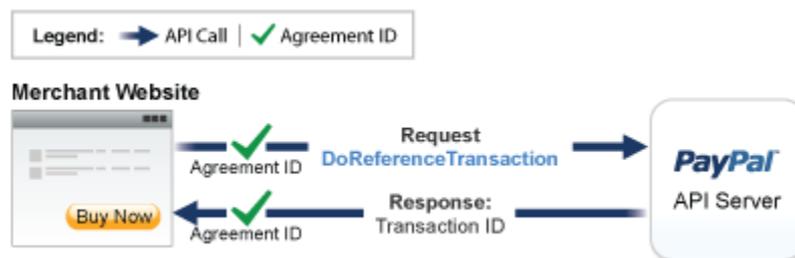
After you establish a billing agreement, you can initiate a payment, which withdraws funds from the buyer's PayPal account without manual intervention. Call `DoReferenceTransaction` to use a reference transaction.

Call `DoReferenceTransaction` and include the following information:

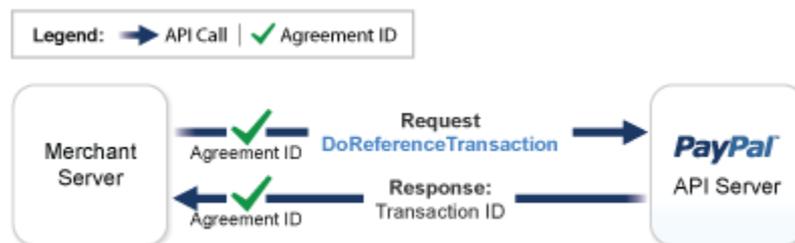
- A reference ID that associates the buyer to a billing agreement. Typically, the ID is the billing agreement ID; however, you can also use the ID associated with a reference transaction.
- A payment action, such as an authorization, order, or final sale.
- Payment details, including the amount of the payment.

PayPal responds with a reference transaction ID that you can keep for future reference and other information about the transaction.

You most likely call `DoReferenceTransaction` in response to the buyer confirming a purchase on your site. The following diagram shows the `DoReferenceTransaction` message flow for a buyer making a purchase from your site.



PayPal does not require that you have a buyer on your site. For example, you could call `DoReferenceTransaction` for each buyer in a batch of buyers whose utility bills are due. The following diagram shows the `DoReferenceTransaction` message flow for a server-initiated recurring payment.



About Cancelling Agreements and Getting the Billing Address

You can use the `BAUpdate` API operation (NVP: `METHOD=BillAgreementUpdate`) to perform two distinct functions: cancel a billing agreement or obtain the buyer's latest billing address.

NOTE: To update a billing agreement (as the operation name implies), you call `BAUpdate` to cancel the agreement. Then, you must create a new billing agreement.

Canceling a Billing Agreement

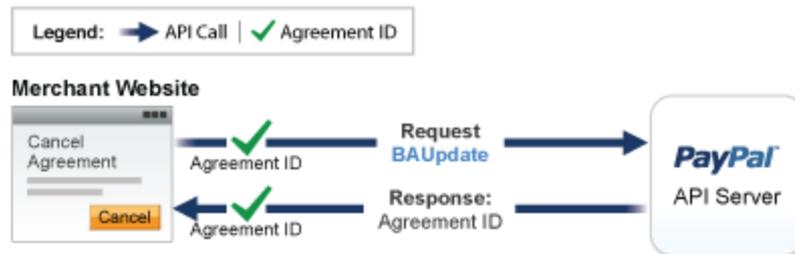
You can use `BAUpdate` (NVP: `METHOD=BillAgreementUpdate`) to cancel a billing agreement.

To cancel an agreement, call `BAUpdate` and include the following information:

- A reference ID that associates the buyer to a billing agreement. Typically, the ID is the billing agreement ID; however, you can also use a ID from a reference transaction.
- An action, which must be `Canceled`, in the `BillingAgreementStatus` field.

PayPal responds with the billing agreement ID and other information about the buyer whose agreement was canceled. Although a buyer can log in to PayPal to manage agreements, the `BAUpdate` API operation enables the buyer to cancel the agreement from your site without logging in to PayPal. You can provide your own page for maintaining agreements with the buyer.

The following diagram shows the message flow to cancel a billing agreement from your page:



Obtaining the Most Recent Billing Address

You can use `BAUpdate` (NVP: `METHOD=BillAgreementUpdate`) to obtain the buyer's most recent billing address.

NOTE: This feature applies to all new and existing reference transaction-based billing agreements and preapproved payment-originated agreements.

To obtain the billing address, call `BAUpdate` without passing a value in the `BillingAgreementStatus` field. The billing address is returned in the `BAUpdate` response.

5

Implementing Parallel Payments

Parallel payments enables a single buyer to pay multiple merchants in a single checkout session. Parallel payments is available with API version 63.0 and higher.

About Parallel Payments

Parallel payments enables buyers to pay multiple merchants on a marketplace in a single Express Checkout session.

An online travel agency marketplace is a typical example of parallel payments in use. The buyer purchases airline tickets and makes reservations from various merchants such as hotels, car rental agencies, and entertainment venues hosted on the site. By implementing parallel payments through Express Checkout, the marketplace host accepts PayPal as a payment method. The host also provides the buyer with a consolidated order on the PayPal **Review your information** page, summarizing expenses, itineraries, and other supporting information. Buyers see travel information, including cancellation fees, directly from the supplier on the **Transaction Details** page and in an email message.

What Is and What Is Not Supported

Parallel payments:

- Supports sales and orders that you later capture with the authorization and capture API operations
 - Supports up to 10 *payments* in one Express Checkout session
- NOTE:** The same merchant can receive multiple payments in one Express Checkout session.
- Does not support use of the Instant Update API (callback)
 - Does not support accelerated boarding; however single-payment transactions are still supported
 - Does not support parallel billing agreements

Post-Integration Experience

After you integrate parallel payments, the PayPal cart review area shows summary information for each payment. The following figure is an example of summary information for an online travel agency with payments to an airline and a hotel.

Your order summary

Description	Amount
<input type="checkbox"/> Vandelay Airlines	\$300.00
<input type="checkbox"/> Central Perk Hotel	\$200.00
Total USD	\$500.00

The following figure shows expanded details on the airline purchase.

Description	Amount
<input type="checkbox"/> Vandelay Airlines Depart San Jose Feb 12 at 12:10 PM ... SJC Terminal 1. Flight time: 7 hour... Item number: Flight 522 Item price: \$50.00 Quantity: 1 Depart Baltimore Feb 15 at 6:13 PM ... BWI Terminal 1. Flight time: 7 hour... Item number: Flight 961 Item price: \$150.00 Quantity: 1	\$300.00
Total USD	\$500.00

The following figure shows expanded details on the hotel payment.

Your order summary

Description	Amount
Insurance: 0.00	
<input type="checkbox"/> Central Perk Hotel	\$200.00
Night(s) stay at 9990 Deereco Road,... King No-Smoking; Check-in after 4:0... Item number: 300 Item price: \$45.00 Quantity: 4 Tax: 20.00 Shipping and handling: 0.00 Insurance: 0.00	
Total USD	\$500.00

Name-Value Pair Syntax Supporting Parallel Payments

The PayPal API uses a special syntax for NVP fields to support parallel payments.

The NVP interface to the PayPal API supports up to a maximum of 10 parallel payments in a transaction. To accommodate this, request fields have the following format, where n is a number in the range 0 to 9 representing a payment.

`PAYMENTREQUEST_n_NVPREQUESTFIELDNAME`

The first numbered field in a list of payments starts with n equal to 0, the second field has n equal to 1, and so forth.

The response name format is:

`PAYMENTREQUEST_n_NVPRESPONSEFIELDNAME`

NOTE: Even if your Express Checkout integration supports single payments only, you must use this format. Specify $n=0$ for single payment with version 63.0 or higher of the Express Checkout API.

The payment information returned in the `DoExpressCheckoutPayment` response has the same basic format; however, the field name starts with `PAYMENTINFO`:

`PAYMENTINFO_n_NVPRESPONSEFIELDNAME`

The NVP API reference documentation shows the proper format and naming for every NVP field that uses this syntax.

Examples:

The following syntax represents the total amount of the first payment:

PAYMENTREQUEST_0_AMT

The following represents the second line of the name for the third payment:

L_PAYMENTREQUEST_2_NAME1

Integrating Parallel Payments by Using the NVP API

To integrate parallel payments by using the NVP API, you need to use the syntax for creating unique NVP request field names and create a unique set of fields for each payment. You also need to set a few required variables.

To integrate parallel payments using the NVP interface to Express Checkout:

1. Create a unique set of NVP request fields for each payment you will be hosting on your marketplace using the syntax `PAYMENTREQUEST_n_NVPREQUESTFIELDNAME` where *n* is a value from 0 - 9.
2. You are required to pass values in the following Payment Details Type fields in the call to `SetExpressCheckout` and `DoExpressCheckoutPayment`. For each of the *n* payments you host:
 - Pass the value `Sale` or `Order` in `PAYMENTREQUEST_n_PAYMENTACTION`.
 - Pass a unique value for `PAYMENTREQUEST_n_PAYMENTREQUESTID`. You will use this value to locate the matching payment response details for that payment.
 - Pass the merchant's Payer ID (secure merchant account ID) or the merchant's email address in `PAYMENTREQUEST_n_SELLERPAYPALACCOUNTID`.
3. Use the Payment Details Item Type fields as appropriate in the call to `SetExpressCheckout` and `DoExpressCheckoutPayment` to pass data about each payment.

Result:

For each payment in the transaction, the `DoExpressCheckoutPayment` response returns:

- A `PAYMENTINFO_n_PAYMENTREQUESTID` value that matches the `PAYMENTREQUEST_n_PAYMENTREQUESTID` value you passed in the `DoExpressCheckoutPayment` request. Use this value to locate the response data for each payment.
- A `PAYMENTINFO_n_SELLERPAYPALACCOUNTID`. This value is whichever one of the following values was passed in:
 - The merchant's email address
 - The merchant's Payer ID (secure merchant account ID)

If errors are returned, check the response data in the PAYMENTERROR field for each payment. It is possible that errors are returned only for a subset of the payments, while other payments are successful. For failed payments, you should ask the buyer for an alternate payment method.

Example

The following is an example SetExpressCheckoutrequest with parallel payments integrated.

Request Parameters

[requiredSecurityParameters]

```
&METHOD=SetExpressCheckout
&RETURNURL=http://...
&CANCELURL=http://...
&PAYMENTREQUEST_0_CURRENCYCODE=USD
&PAYMENTREQUEST_0_AMT=300
&PAYMENTREQUEST_0_ITEMAMT=200
&PAYMENTREQUEST_0_TAXAMT=100
&PAYMENTREQUEST_0_DESC=Summer Vacation trip
&PAYMENTREQUEST_0_INSURANCEAMT=0
&PAYMENTREQUEST_0_SHIPDISCAMT=0
&PAYMENTREQUEST_0_SELLERPAYPALACCOUNTID=seller-139@paypal.com
&PAYMENTREQUEST_0_INSURANCEOPTIONOFFERED=false
&PAYMENTREQUEST_0_PAYMENTACTION=Order
&PAYMENTREQUEST_0_PAYMENTREQUESTID=CART26488-PAYMENT0
&PAYMENTREQUEST_1_CURRENCYCODE=USD
&PAYMENTREQUEST_1_AMT=200
&PAYMENTREQUEST_1_ITEMAMT=180
&PAYMENTREQUEST_1_SHIPPINGAMT=0
&PAYMENTREQUEST_1_HANDLINGAMT=0
&PAYMENTREQUEST_1_TAXAMT=20
&PAYMENTREQUEST_1_DESC=Summer Vacation trip
&PAYMENTREQUEST_1_INSURANCEAMT=0
&PAYMENTREQUEST_1_SHIPDISCAMT=0
&PAYMENTREQUEST_1_SELLERPAYPALACCOUNTID=seller-140@paypal.com
&PAYMENTREQUEST_1_INSURANCEOPTIONOFFERED=false
&PAYMENTREQUEST_1_PAYMENTACTION=Order
&PAYMENTREQUEST_1_PAYMENTREQUESTID=CART26488-PAYMENT1
&L_PAYMENTREQUEST_0_NAME0=Depart San Jose Feb 12 at 12:10PM Arrive in
Baltimore at 10:22PM
&L_PAYMENTREQUEST_0_NAME1=Depart Baltimore Feb 15 at 6:13 PM Arrive in San
Jose at 10:51 PM
&L_PAYMENTREQUEST_0_NUMBER0=Flight 522
&L_PAYMENTREQUEST_0_NUMBER1=Flight 961
&L_PAYMENTREQUEST_0_QTY0=1
&L_PAYMENTREQUEST_0_QTY1=1
&L_PAYMENTREQUEST_0_TAXAMT0=50
&L_PAYMENTREQUEST_0_TAXAMT1=50
&L_PAYMENTREQUEST_0_AMT0=50
&L_PAYMENTREQUEST_0_AMT1=150
```

```
&L_PAYMENTREQUEST_0_DESC0=SJC Terminal 1. Flight time: 7 hours 12 minutes
&L_PAYMENTREQUEST_0_DESC1=BWI Terminal 1. Flight time: 7 hours 38 minutes
&L_PAYMENTREQUEST_1_NAME0=Night(s) stay at 9990 Deereco Road,Timonium, MD
21093
&L_PAYMENTREQUEST_1_NUMBER0=300
&L_PAYMENTREQUEST_1_QTY0=1
&L_PAYMENTREQUEST_1_TAXAMT0=20
&L_PAYMENTREQUEST_1_AMT0=180
&L_PAYMENTREQUEST_1_DESC0=King No-Smoking; Check in after 4:00 PM; Check
out by 1:00 PM
```

Response Parameters

[successResponseFields]

```
&TOKEN=EC-17C76533PL706494P
```

Integrating Parallel Payments by Using the SOAP API

Parallel payments uses the `PaymentDetailsType` structure to pass data for each merchant receiving payment. You can pass up to a 10 structures in a single call to `SetExpressCheckout` and `DoExpressCheckoutPayment`.

NOTE: Be sure to use structure fields that are not marked as ‘deprecated’ in the SOAP API reference documentation.

To integrate parallel payments by using the SOAP interface to Express Checkout:

1. Create `PaymentDetails` as an array of `PaymentDetailsType` structures, one for each payment you host on your marketplace.
2. You are required to pass values in the following `PaymentDetailsType` fields in the call to `SetExpressCheckout` and `DoExpressCheckoutPayment`.
 - Pass the value `Sale` or `Order` in `PaymentAction`.
 - Pass a unique value in `PaymentRequestID`. You will use this value to locate the matching response details for that payment.
 - Pass the merchant’s Payer ID (secure merchant account ID) or the merchant’s email address in the `SellerDetailsType.PayPalAccountId` field.
3. Use `PaymentDetailsType` and `PaymentDetailsItemType` fields, as appropriate, in the call to `SetExpressCheckout` and `DoExpressCheckoutPayment` to pass data about each payment.

Result:

For each payment in the transaction, the `DoExpressCheckoutPayment` response returns a `PaymentInfoType` structure corresponding to each payment:

- The `PaymentRequestID` will match the value you passed in the `DoExpressCheckoutPayment` request. Use this value to locate the response data for each payment.
- `SellerDetailsType.PayPalAccountId` returns one of the following values that was passed in:
 - The merchant's email address
 - The merchant's Payer ID (secure merchant account ID)

If errors occur, check the response data in the `PaymentInfo.PaymentError` field. `PaymentError` returns the `ErrorType` information. It is possible that errors are returned only for a subset of payments, while other payments are successful. For failed payments, you should ask the buyer for an alternate payment method.

The following SOAP example sets up the merchants receiving funds:

```
PaymentDetailsType[] PaymentDetailsArray = new PaymentDetailsType[9];
//*****
//merchant 1
//*****
PaymentDetailsType payment1 = new PaymentDetailsType();
payment1.PaymentAction = PaymentActionCodeType.Order;
payment1.PaymentActionSpecified = true;
payment1.SellerDetails = new SellerDetailsType();
payment1.SellerDetails.PayPalAccountId = "support@1stimagehosting.com";
//set up the line items for the first merchant
PaymentDetailsItemType[] payment1_items_array =
    new PaymentDetailsItemType[2];
PaymentDetailsItemType payment1_item1 = new PaymentDetailsItemType();
payment1_item1.Amount = new BasicAmountType();
payment1_item1.Amount.currencyID = CurrencyCodeType.USD;
payment1_item1.Amount.Value = "1.00";
payment1_item1.Description = "payment1_item1_desc";
payment1_item1.Name = "payment1_item1_name";
payment1_item1.Number = "payment1_item1_number";
payment1_item1.ItemURL = "http://item1.com";
payment1_item1.Quantity = "3";
payment1_item1.Tax = new BasicAmountType();
payment1_item1.Tax.currencyID = CurrencyCodeType.USD;
payment1_item1.Tax.Value = ".50";
PaymentDetailsItemType payment1_item2 = new PaymentDetailsItemType();
payment1_item2.Amount = new BasicAmountType();
payment1_item2.Amount.currencyID = CurrencyCodeType.USD;
payment1_item2.Amount.Value = "1.00";
payment1_item2.Description = "payment1_item2_desc";
payment1_item2.Name = "payment1_item2_name";
```

```

payment1_item2.Number = "payment1_item2_number";
payment1_item2.ItemURL = "http://item2.com";
payment1_item2.Quantity = "2";
payment1_item2.Tax = new BasicAmountType();
payment1_item2.Tax.currencyID = CurrencyCodeType.USD;
payment1_item2.Tax.Value = ".25";
payment1_items_array.SetValue(payment1_item1, 0);
payment1_items_array.SetValue(payment1_item2, 1);
//bind the items
payment1.PaymentDetailsItem = payment1_items_array;
//set the totals
decimal tax_total = 0;
decimal item_total = 0;
foreach (PaymentDetailsItemType Key in payment1_items_array)
{
    if (Key.Tax != null)
    {
        tax_total = decimal.Add(tax_total,
            decimal.Multiply(decimal.Parse(Key.Tax.Value),
                decimal.Parse(Key.Quantity)));
    }
    if (Key.Amount != null)
    {
        item_total = decimal.Add(item_total,
            decimal.Multiply(decimal.Parse(Key.Amount.Value),
                decimal.Parse(Key.Quantity)));
    }
}
payment1.ShippingTotal = new BasicAmountType();
payment1.ShippingTotal.currencyID = CurrencyCodeType.USD;
payment1.ShippingTotal.Value = "3.00";
payment1.ItemTotal = new BasicAmountType();
payment1.ItemTotal.currencyID = CurrencyCodeType.USD;
payment1.ItemTotal.Value = item_total.ToString();
payment1.TaxTotal = new BasicAmountType();
payment1.TaxTotal.currencyID = CurrencyCodeType.USD;
payment1.TaxTotal.Value = tax_total.ToString();
decimal order_total = decimal.Add(decimal.Add(tax_total, item_total),
    decimal.Parse(payment1.ShippingTotal.Value));
payment1.OrderTotal = new BasicAmountType();
payment1.OrderTotal.currencyID = CurrencyCodeType.USD;
payment1.OrderTotal.Value = order_total.ToString();
//mandatory for api call
payment1.PaymentRequestID = System.Guid.NewGuid().ToString();
//add the merchants to the array
PaymentDetailsArray.SetValue(payment1, 0);

//*****
//merchant 2
//*****
PaymentDetailsType payment2 = new PaymentDetailsType();

```

```

payment2.PaymentAction = PaymentActionCodeType.Order;
payment2.PaymentActionSpecified = true;
payment2.SellerDetails = new SellerDetailsType();
payment2.SellerDetails.PayPalAccountID = "airline@grupellc.com";
//items for payment2
PaymentDetailsItemType[] payment2_items_array =
    new PaymentDetailsItemType[2];
PaymentDetailsItemType payment2_item1 = new PaymentDetailsItemType();
payment2_item1.Amount = new BasicAmountType();
payment2_item1.Amount.currencyID = CurrencyCodeType.USD;
payment2_item1.Amount.Value = "1.00";
payment2_item1.Description = "payment2_item1_desc";
payment2_item1.Name = "payment2_item1_name";
payment2_item1.Number = "payment2_item1_number";
payment2_item1.Quantity = "1";
PaymentDetailsItemType payment2_item2 = new PaymentDetailsItemType();
payment2_item2.Amount = new BasicAmountType();
payment2_item2.Amount.currencyID = CurrencyCodeType.USD;
payment2_item2.Amount.Value = "1.00";
payment2_item2.Description = "payment2_item2_desc";
payment2_item2.Name = "payment2_item2_name";
payment2_item2.Number = "payment2_item2_number";
payment2_item2.Quantity = "1";
payment2_items_array.SetValue(payment2_item1, 0);
payment2_items_array.SetValue(payment2_item2, 1);
//bind the items
payment2.PaymentDetailsItem = payment2_items_array;
//mandatory for api call
payment2.PaymentRequestID = System.Guid.NewGuid().ToString();

//set the totals
decimal tax_total2 = 0;
decimal item_total2 = 0;
foreach (PaymentDetailsItemType Key in payment2_items_array)
{
    if (Key.Tax != null)
    {
        tax_total2 = decimal.Add(tax_total2,
            decimal.Multiply(decimal.Parse(Key.Tax.Value),
                decimal.Parse(Key.Quantity)));
    }
    if (Key.Amount != null)
    {
        item_total2 = decimal.Add(item_total2,
            decimal.Multiply(decimal.Parse(Key.Amount.Value),
                decimal.Parse(Key.Quantity)));
    }
}
payment2.ShippingTotal = new BasicAmountType();
payment2.ShippingTotal.currencyID = CurrencyCodeType.USD;
payment2.ShippingTotal.Value = "3.00";

```

```

payment2.ItemTotal = new BasicAmountType();
payment2.ItemTotal.currencyID = CurrencyCodeType.USD;
payment2.ItemTotal.Value = item_total2.ToString();
payment2.TaxTotal = new BasicAmountType();
payment2.TaxTotal.currencyID = CurrencyCodeType.USD;
payment2.TaxTotal.Value = tax_total2.ToString();
decimal order_total2 = decimal.Add(decimal.Add(tax_total2, item_total2),
    decimal.Parse(payment2.ShippingTotal.Value));
payment2.OrderTotal = new BasicAmountType();
payment2.OrderTotal.currencyID = CurrencyCodeType.USD;
payment2.OrderTotal.Value = order_total2.ToString();
//add the merchants to the array
PaymentDetailsArray.SetValue(payment2, 1);

//bind the merchants to the request
SetECReq.SetExpressCheckoutRequest.SetExpressCheckoutRequestDetails.Payment
    Details = PaymentDetailsArray;

```

Best Practices for Online Travel Agencies Implementing Parallel Payments

PayPal provides special recommendations to online travel agencies implementing parallel payments to help the buyer complete the payment flow.

Best practices address ways that you as an online travel agency designer can meet the needs of the merchants hosted on your marketplace. The following are examples of special needs of these merchants:

- Merchants providing travel services must offer different styles of payment.
- Merchants offering a service with reservations would want the buyer to know about the cancellation policy and fees.

Styles of Payment

Merchants hosted by online travel agencies might be using one or more of the following styles of payment:

- **Prepaid** – Prepaid payments are paid in full at the time of checkout. This style is typical of most online purchases.
- **Deposit** – A deposit is paid before a service is rendered. Usually it is a flat rate equal to a small percentage of the total, such as a \$50.00 deposit on a cruise. The balance is paid offline at some point before the cruise or other service takes place.
- **Postpaid** – A postpaid expense is not paid until after the service is rendered. Hotel stays, for example, are typically paid at the time of checkout.

Express Checkout displays to the buyer a total amount for all goods or services. For a good buyer experience, you can pass additional information about a deposit or postpaid expense

using the Express Checkout `PaymentDetailsType` and `PaymentDetailsItemType` parameter fields. To eliminate potential buyer confusion, you can let the buyer making a deposit through Express Checkout know, for example, that they will need another payment instrument to complete their purchase.

Payment Details

The following table provides PayPal's recommendations for passing payment information.

Best practices for passing payment information

Item	Best practices	Example content
Merchant name	You do not need to pass this data. PayPal obtains the merchant's name that displays on the PayPal Review your information page from the merchant's account record.	PayPal presents the business name, for example, <i>Airline Name</i> .
Item description	Use <code>L_PAYMENTREQUEST_n_DESCn</code> to pass item description information as a continuous text string. Pass at least the primary information shown in the table below. PayPal recommends that you also pass the secondary information. If the style of payment is a reservation with deposit, pass the deposit only in the item description.	See the table below.
Order description	Use <code>PAYMENTREQUEST_n_DESC</code> to pass merchant-specific messages such as cancellation charges for post-paid offline transactions. This information does not display on the PayPal Review page. Instead, it appears in the buyer's email and in Transaction Details.	Your account will not be charged now. You'll later make a payment directly to <i>merchant name</i> . If you cancel the order, ...
Note to merchant	Do not use the <code>ALLOWNOTE</code> parameter. It is ignored for parallel payments.	

Pass, at least, the primary information below. PayPal recommends that you also pass the secondary information. If the style of payment is a reservation with deposit, pass the deposit only in the item description.

Recommended item description information

Merchant	Item description information
Flight	<p>Primary information</p> <p>Flight No. Departure Location, Date & Time Arrival Location, Date & Time</p> <p>Secondary information</p> <p>Duration Terminal No. Distance</p>
Hotel	<p>Primary information</p> <p>Location Check-in Date Checkout Date</p> <p>Secondary information</p> <p>Check in Time Room Type Bed Size</p>
Car	<p>Primary information</p> <p>Pickup Location, Date & Time Drop-off Location, Date & Time</p> <p>Secondary information</p> <p>Car Model & Make Telephone No.</p>
Insurance	<p>Primary information</p> <p>Insurance Company Name & Description</p>
Cruise	<p>Primary information</p> <p>Date, Time & Description</p>

Handling Errors

It is possible for some merchant payments to succeed while others fail. Parallel payments creates multiple independent payments, and each payment is subject to its own validation and review.

If part of a payment fails, the ACK value is `PartialSuccess`. To find the error, check the value returned in `PAYMENTINFO_n_ERRORCODE` in the response to `DoExpressCheckoutPayment`.

NOTE: If an error is generated by any of the payments in the call to `SetExpressCheckout`, the transaction fails.

6

Integrating giropay with Express Checkout

You must modify your Express Checkout implementation to use giropay, a common German funding source.

giropay Page Flows

If you accept giropay, you must redirect to the giropay website to collect the funds after completing the Express Checkout transaction.

When your buyer selects giropay as a funding source during the Express Checkout flow, you redirect the buyer to a static PayPal URL after your order review page. PayPal then redirects the buyer to the giropay website to push the funds to the merchant. After the giropay payment is successfully completed, the transaction is confirmed.

If the giropay payment fails or the buyer cancels, PayPal provides the necessary details for an electronic funds transfer (EFT) so that the buyer can complete the transaction by pushing funds from his or her bank account. If your PayPal account profile blocks non-instant payments, the transaction is canceled.

giropay Payment Page Flow

The following diagram illustrates the flow of a successful giropay payment:



Canceled or Unsuccessful giropay Payment Page Flow

If the giropay payment fails for any reason, such as insufficient funds or the buyer cancels, PayPal provides details to the buyer to do a bank transfer from their bank account. This transaction will remain pending until PayPal receives the funds, at which time the transaction is complete.

If you have disabled non-instant funding transactions for your PayPal account, the transaction is canceled and PayPal redirects the buyer to your **Order Cancel** page.

After the bank transfer flow completes, the transaction is pending until the buyer pushes the funds to PayPal.

If the buyer cancels during the PayPal payment in the bank transfer flow, your **Order Cancel** page is displayed.

giropay Integration

If you offer the giropay payment option, you must take additional steps to integrate with the Express Checkout flow.

Initiate the Flow with SetExpressCheckout

To support giropay payments, pass the following three URLs as part of the SetExpressCheckout request:

NVP Field	Description
GIROPAYSUCCESSURL	The URL on the merchant site to redirect to after a successful giropay payment.
GIROPAYCANCELURL	The URL on the merchant site to redirect to after a giropay or bank transfer payment is cancelled or fails.
BANKTXNPENDINGURL	The URL on the merchant site to transfer to after a bank transfer payment.

These URLs tell PayPal where to redirect the buyer based on the success or failure of each type of payment transaction. See the *PayPal Name-Value Pair Developer Guide and Reference* for more information.

Redirect the Buyer to PayPal

After selecting a funding source on PayPal, the buyer is redirected back to your website, as in the regular Express Checkout flow. There is one additional field, REDIRECTREQUIRED, returned in the response from both GetExpressCheckoutDetails and DoExpressCheckoutPayment:

NVP Field	Description
REDIRECTREQUIRED	Flag to indicate whether you need to redirect the buyer to back to PayPal

If the value of this field is true, redirect the buyer from your **Order Review** page to `https://www.paypal.com/webscr?cmd=_complete-express-checkout&token=<value_from_SetExpressCheckoutResponse>`. PayPal then redirects the buyer from this redirect page to the necessary page for the selected funding source.

The `GetExpressCheckoutDetails` response contains the `REDIRECTREQUIRED` field, which lets you know if you need to redirect the buyer after your **Order Review** page. You can use this value to inform the buyer on your **Order Review** page that they will be sent to the giroipay website to complete the order.

Complete the Transaction

Corresponding to the three fields passed to `SetExpressCheckout`, you must add the following three additional pages to your website:

Your website's pages	Description
Order Completion	The page specified in <code>GIROPAYSUCCESSURL</code> to which PayPal redirects the buyer after a successful giroipay payment.
Order Cancelation	The page specified in <code>GIROPAYCANCELURL</code> to which PayPal redirects the buyer after a giroipay or bank transfer payment is canceled or fails.
Order Pending	The page specified in <code>BANKTXNPENDINGURL</code> to which PayPal redirects the buyer after a bank transfer payment.

Receive Transaction Status Notification

After PayPal redirects the buyer to giroipay, you receive transaction status information in the following ways:

- IPN Notification
- Email (only for successful giroipay or bank transfer transactions)
- PayPal Account Overview
- PayPal reports

7

Implementing the Instant Update API

The Instant Update API is a callback you can use to obtain the buyer's shipping address.

About the Instant Update API

The Instant Update API is a server call to your callback server that instantly updates PayPal pages and enhances the Express Checkout experience on the **Review your information** page.

The Instant Update API enables you to specify a URL with which PayPal can call your callback server with the buyer's shipping address, so you can provide the buyer with more detailed shipping, insurance, and tax information.

Here is how the Instant Update API works:

1. When a buyer logs in to PayPal, the PayPal server calls your callback server with the buyer's default shipping address, which is stored in the PayPal system.
2. Your callback server responds with the shipping options available for that address, along with any insurance options and tax adjustments on the order.
3. PayPal displays this information in the cart review area so buyers can choose from the options.
4. The buyer's final choices are returned in the `GetExpressCheckoutDetails` response.

Integration Steps

Integrating the Instant Update API requires some preparation and modification to the Express Checkout API calls.

To integrate the server API, follow these steps:

1. Set up a secure, fast web service to accept HTTP requests from PayPal. On the live site, it needs to be secured by means of SSL.
2. Enable the callback service to process PayPal requests and return responses.
3. Modify the existing Express Checkout API calls to accommodate new parameters.
 - Send the callback URL, shipping, insurance, and tax information to PayPal in the call to `SetExpressCheckout`.
 - Call `GetExpressCheckoutDetails` to obtain the buyer's final choices for shipping and insurance, if applicable.
 - Call `DoExpressCheckoutPayment` with the buyer's final selections.

4. Eliminate your shipping options page.
5. Test your integration for the callback and flat-rate shipping options.

Post-Integration Checkout Experience

After you integrate the Instant Update API, you can display the shipping options, related insurance options, and the tax amount. You control what to display, which is instantly updated on the page.

The shipping options, related insurance options, and the tax amount appear on the page, as follows:

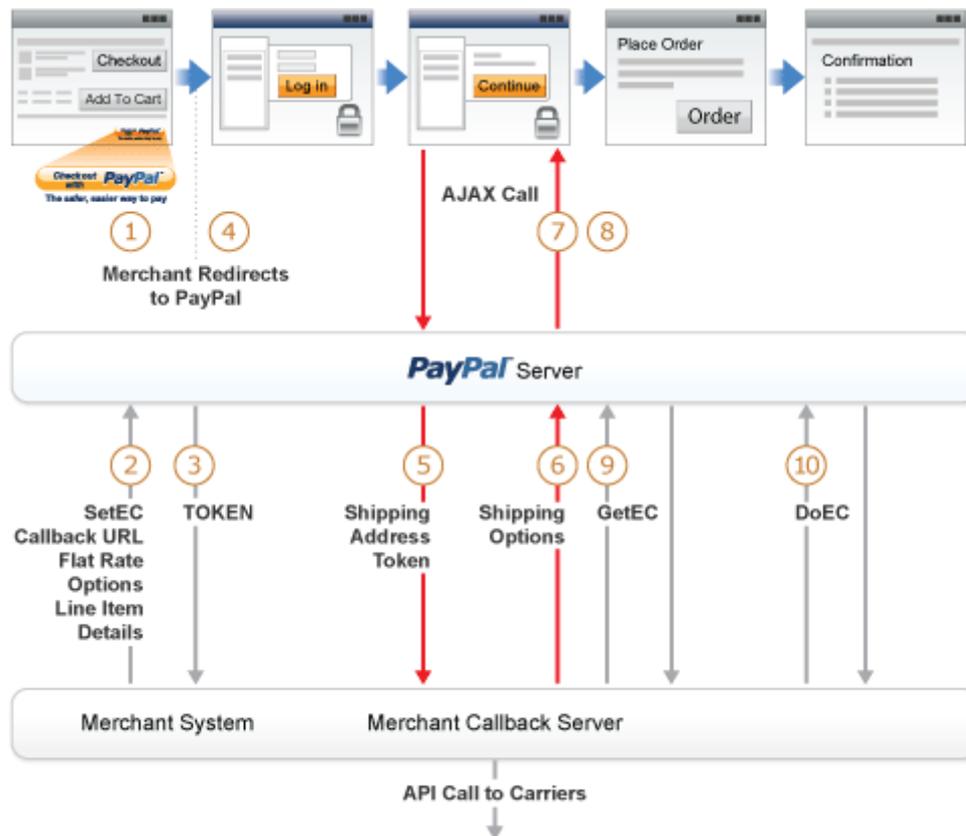
Joe's Generic Business

Your order summary	
Descriptions	Amount
10% Decaf Kona Blend Coffee	\$19.90
Item description: Size: 8.6-oz	
Item number: 623063	
Item price: \$9.95	
Quantity: 2	
<hr/>	
Coffee Filter bags	\$79.40
Item description: Size: Two 24-piece...	
Item number: 6230	
Item price: \$39.70	
Quantity: 2	
<hr/>	
Item total	\$99.30
Tax	\$2.59
Shipping and handling	\$8.00
Shipping discount	\$3.00
Insurance	\$1.00
Total \$107.89 USD	

How the Callback Works in the Express Checkout Flow

The Express Checkout flow is initiated on your shopping cart page when the buyer clicks **Checkout with PayPal**.

Callback integrated into Express Checkout flow



From left to right, the following events are represented:

1. The Express Checkout flow is initiated on your shopping cart page when the buyer clicks **Checkout with PayPal**.
2. In the call to the `SetExpressCheckout` API operation, you provide the URL where PayPal can call your callback server, the flat-rate shipping options, and cart line-item details.
3. You retrieve the token from the response.
4. You redirect the buyer's browser to PayPal.
5. When the buyer first logs in to the PayPal site, PayPal obtains the buyer's shipping address and sends it in the callback request (shown as the red down arrow) to your callback server at the specified URL.

NOTE: If the buyer changes their shipping address on the PayPal pages during checkout, PayPal will make subsequent requests to the callback server.

6. Your callback server responds with the shipping option rates (shown by the red up arrow) based on the buyer's shipping address. You can also adjust the tax amount and send insurance options. Depending on your business processes, you may send an API call to your carrier to calculate the rates and options based on the shipping address.
7. PayPal updates the cart review area and Review your information page content to show the options and rates based on your response.
8. The buyer makes final selections and clicks **Continue**.
9. You must call `GetExpressCheckoutDetails` to obtain the buyer's final shipping option selections.
10. You call `DoExpressCheckoutPayment` to perform the transaction.

Following Instant Update API Best Practices

PayPal recommends its list of best practices as a checklist for completing your implementation of the Instant Update API.

- **Meet the prerequisites** – Provide order line-item details to take advantage of the Instant Update API.
- **Streamline the checkout flow** – Existing partners and merchants with Express Checkout integrations can eliminate the current shipping options page.
- **Use the default callback timeout** – Use the recommended 3-second callback response timeout.
- **Follow PayPal-defined semantics and syntax** – Adhere to well-formed variable names and syntax rules in the callback response to PayPal. If errors occur in the response, PayPal uses the flat-rate shipping options.
- **Call `GetExpressCheckoutDetails`** – You must call `GetExpressCheckoutDetails` to find out what options the buyer selected.
- **Ensure a consistent and good buyer experience** – With flat-rate shipping options, you should honor the rates to ensure a consistent and correct buyer experience.
- **Localize shipping options** – Return localized shipping options, based on the buyer's country and locale, which PayPal sends in the callback request.

Related information:

"PayPal Review Page Order Details" on page 9

Setting Up the Callback

To set up the callback, establish a connection with PayPal by providing the location where PayPal calls your callback server, along with your shipping options.

To start, you must build and operate a secure, reliable, and fast callback server that computes shipping options, corresponding insurance options, and tax, based on your business rules. To verify that callback requests originate from PayPal.

The HTTP protocol to specify in your callback URL depends on the integration environment you are using:

- The callback URL must start with HTTPS for production integration.
- The callback URL must start with HTTP or HTTPS for PayPal Sandbox integration.

In the call to `SetExpressCheckout`, you must complete the steps 1 through 3 below. Steps 4 and 5 are optional:

1. Provide line-item details for the merchandise the buyer selected.
2. Provide the URL to your callback server, which PayPal validates.
3. Provide values for the flat-rate shipping options. For each option, specify:
 - Option name (`L_SHIPPINGOPTIONNAMEn`)
 - Option amount (`L_SHIPPINGOPTIONAMOUNTn`)
 - The shipping option to appear as the default (`true`) (`L_SHIPPINGOPTIONISDEFAULTn`).

NOTE: Set `L_SHIPPINGOPTIONISDEFAULTn` to `true` (default) for one and only one shipping option. Set `L_SHIPPINGOPTIONISDEFAULTn` to `false` for each of the remaining options.

 - If required, an adjusted value for `PAYMENTREQUEST_ n _TAXAMT`
 - If required, an adjusted value `PAYMENTREQUEST_ n _INSURANCEAMT`
4. If necessary to adjust the callback timeout (default: 3 seconds), provide a value from 1 to 6 for the `CALLBACKTIMEOUT` parameter.
5. Optionally, provide values for any of the shipping option description details fields listed below:
 - Option weight (`L_PAYMENTREQUEST_ n _ITEMWEIGHTVALUEm`, `L_PAYMENTREQUEST_ n _ITEMWEIGHTUNITm`)
 - Option height (`L_PAYMENTREQUEST_ n _ITEMHEIGHTVALUEm`, `L_PAYMENTREQUEST_ n _ITEMHEIGHTUNITm`)
 - Option length (`L_PAYMENTREQUEST_ n _ITEMLENGTHVALUEm`, `L_PAYMENTREQUEST_ n _ITEMLENGTHUNITm`)
 - Option width (`L_PAYMENTREQUEST_ n _ITEMWIDTHVALUEm`, `L_PAYMENTREQUEST_ n _ITEMWIDTHUNITm`)

Related information:

"PayPal Review Page Order Details" on page 9

GetExpressCheckoutDetails and DoExpressCheckoutPayment Changes

When you implement the callback, you need to call `GetExpressCheckoutDetails` and `DoExpressCheckoutPayment`.

`GetExpressCheckoutDetails` and `DoExpressCheckoutPayment` include new parameter fields in support of the Instant Update API.

You must call the `GetExpressCheckoutDetails` API operation to obtain the buyer's final shipping option selections. `GetExpressCheckoutDetails` has been updated to return the buyer's selections.

Because the cart information passed in the call to `SetExpressCheckout` is relevant only for display on the PayPal pages, you must call the `DoExpressCheckoutPayment` API operation with the updated shipping, insurance, and tax data to ensure the buyer sees it upon redirect to your website.

Other Callback Considerations

When you implement the callback, you must consider callback response errors, .timeouts, and shipping options.

Callback Response Errors

If there are any callback response errors, PayPal responds by displaying the flat-rate shipping options on the PayPal pages. To obtain the richer set of options available through the callback, exercise care in the syntax and values you specify, and test the callback integration.

Callback Timeouts

If the callback does not return within the timeout period, PayPal displays the flat-rate shipping options you specified in the call to `SetExpressCheckout`.

Minimum and Maximum Shipping Options

You can specify up to 10 shipping options for the flat-rate options in the call to `SetExpressCheckout` and for the detailed options based on shipping address in the callback response. You must specify at least 1 shipping option.

You Do Not Ship to the Buyer's Shipping Address

If you do not ship to the buyer's shipping address that PayPal sends in the callback request, set `NO_SHIPPING_OPTION_DETAILS` to 1 in the callback response.

NOTE: The `CALLBACKVERSION` must have been set to 61.0 or greater in the `SetExpressCheckout` request.

The sample code below illustrates the callback response when you do not ship to the buyer's address.

```
METHOD=CallbackResponse
NO_SHIPPING_OPTION_DETAILS =1
```

When your callback server sends the previous response, the Review your information page has these features:

- A message at the top of the page indicates that your business does not ship to this location.
- The shipping and handling section and the insurance section are dimmed.
- The buyer can change the shipping address.
- A new callback request is sent if the buyer changes the shipping address.

Using the Callback

To use the callback, you add parameter fields to `SetExpressCheckout`, provide PayPal a URL for sending a callback request, and send PayPal the callback response in Name-Value pair (NVP) format.

SetExpressCheckout

In the call to `SetExpressCheckout`, set the following parameters:

- Set the `CALLBACK` field to the URL where PayPal can call your callback server. PayPal makes the HTTPS callback request each time either of the following events occur:
 - The buyer changes their shipping address
 - The buyer enters a new shipping address
- Provide values for the following required parameters:
 - Provide values for the line-item details parameters, such as `L_PAYMENTREQUEST_n_NAME m` , `L_PAYMENTREQUEST_n_NUMBER m` , `L_PAYMENTREQUEST_n_DESC m` , `L_PAYMENTREQUEST_n_AMT m` , and `L_PAYMENTREQUEST_n_QTY m` .
 - Provide values for the flat-rate shipping options: `n`, `L_SHIPPINGOPTIONISDEFAULT n` , `L_SHIPPINGOPTIONNAME n` , and `L_SHIPPINGOPTIONAMOUNT n` .
 - Set `PAYMENTREQUEST_n_SHIPPINGAMT` to the amount set for the default flat-rate shipping option.

If, for example, `L_SHIPPINGOPTIONISDEFAULT0=true` and `L_SHIPPINGOPTIONAMOUNT0=8.00`, then `PAYMENTREQUEST_0_SHIPPINGAMT=8.00`
 - Set `MAXAMT` to the expected maximum total amount of the complete order.

NOTE: PayPal recommends that the maximum total amount be slightly greater than the sum of the line-item order details, tax, and the shipping option of greatest value.

- Optionally, provide values for the following parameters:
 - Set `PAYMENTREQUEST_n_INSURANCEOPTIONOFFERED` to `true` to inform PayPal that you are offering insurance options. Otherwise, set `PAYMENTREQUEST_n_INSURANCEOPTIONSOFFERED` to `false`.
 - Set line-item description details such as `L_PAYMENTREQUEST_n_ITEMWEIGHTUNIT0` and `L_PAYMENTREQUEST_n_ITEMWEIGHTVALUE0` shown in the example below.
 - Set `CALLBACKTIMEOUT` to the amount of time in seconds to process the callback. By default, `CALLBACKTIMEOUT` is 3. You can specify a value in the range of 1 to 6 inclusive.

The following is an example `SetExpressCheckout` request:

Request Parameters

[requiredSecurityParameters]

```
&METHOD=SetExpressCheckout
&RETURNURL=http://...
&CANCELURL=http://...
&PAYMENTREQUEST_0_PAYMENTACTION=Sale
&PAYMENTREQUEST_0_SHIPTONAME=J Smith
&PAYMENTREQUEST_0_SHIPTOSTREET=1 Main St
&PAYMENTREQUEST_0_SHIPTOCITY=San Jose
&PAYMENTREQUEST_0_SHIPTOSTATE=CA
&PAYMENTREQUEST_0_SHIPTOCOUNTRYCODE=US
&PAYMENTREQUEST_0_SHIPTOZIP=95131
&L_PAYMENTREQUEST_0_NAME0=10% Decaf Kona Blend Coffee
&L_PAYMENTREQUEST_0_NUMBER0=623083
&L_PAYMENTREQUEST_0_DESC0=Size: 8.8-oz
&L_PAYMENTREQUEST_0_AMT0=9.95
&L_PAYMENTREQUEST_0_QTY0=2
&L_PAYMENTREQUEST_0_NAME1=Coffee Filter bags
&L_PAYMENTREQUEST_0_NUMBER1=6230
&L_PAYMENTREQUEST_0_DESC1=Size: Two 24-piece boxes
&L_PAYMENTREQUEST_0_AMT1=39.70
&L_PAYMENTREQUEST_0_QTY1=2
&L_PAYMENTREQUEST_0_ITEMWEIGHTVALUE0=0.5
&L_PAYMENTREQUEST_0_ITEMWEIGHTUNIT0=lbs
&PAYMENTREQUEST_0_ITEMAMT=99.30
&PAYMENTREQUEST_0_TAXAMT=2.59
&PAYMENTREQUEST_0_MAXAMT=150.00
&PAYMENTREQUEST_0_SHIPPINGAMT=8.00
&PAYMENTREQUEST_0_SHIPDISCAMT=-3.00
&PAYMENTREQUEST_0_AMT=107.89
&PAYMENTREQUEST_0_CURRENCYCODE=USD
&ALLOWNOTE=1
&CALLBACK=https://...
&CALLBACKTIMEOUT=4
&PAYMENTREQUEST_0_INSURANCEOPTIONOFFERED=true
&PAYMENTREQUEST_0_INSURANCEAMT=1.00
```

```

&L_SHIPPINGOPTIONISDEFAULT0=false
&L_SHIPPINGOPTIONNAME0=UPS Ground 7 Days
&L_SHIPPINGOPTIONAMOUNT0=3.50
&L_SHIPPINGOPTIONISDEFAULT1=true
&L_SHIPPINGOPTIONNAME1=UPS Next Day Air
&L_SHIPPINGOPTIONAMOUNT1=8.00

```

Response Parameters

[successResponseFields]

```
&TOKEN=EC-17C76533PL706494P
```

Callback Request

The PayPal sends the parameters in the callback request to the location you specified for `CALLBACK`. The callback request parameters include:

- Line-item details you sent in the call to `SetExpressCheckout`. PayPal also sends back any line-item description details you may have specified, such as the `L_ITEMWEIGHTUNIT1` and `L_ITEMWEIGHTVALUE1` values shown in the example below. By passing this data back to you, PayPal expedites your callback response by eliminating the need for you to perform a database query to obtain this information.
- Shipping address of the buyer.

Using the information in the callback request, calculate the rates and options yourself or send the information in an API call to your carrier to perform the calculations for you. Then send the shipping options, insurance amounts, and taxes to PayPal in the callback response.

This is an example callback request:

```

METHOD=CallbackRequest
&CALLBACKVERSION=XX.0
&TOKEN=EC-0EE85728D547104V
&CURRENCYCODE=USD
&LOCALECODE=en_US
&L_NAME0=10% Decaf Kona Blend Coffee
&L_NUMBER0=623083
&L_DESC0=Size: 8-oz
&L_AMT0=9.95
&L_QTY0=2
&L_NAME1=Coffee Filter bags
&L_NUMBER1=6230
&L_DESC1=Size: Two 24-piece boxes
&L_AMT1=39.70
&L_QTY1=2
&L_ITEMWEIGHTUNIT1=lbs
&L_ITEMWEIGHTVALUE1=0.5
&SHIPTOSTREET=1 Main St

```

```
&SHIPTOCITY=San Jose
&SHIPTOSTATE=CA
&SHIPTOCOUNTRY=US
&SHIPTOZIP=95131
&SHIPTOSTREET2
```

Callback Response

Each time your callback server receives a request from PayPal, it must process the request and respond with the appropriate details.

This is an example callback response:

```
METHOD=CallbackResponse
&OFFERINSURANCEOPTION=true
&L_SHIPPINGOPTIONNAME0=UPS Next Day Air
&L_SHIPPINGOPTIONAMOUNT0=20.00
&L_TAXAMT0=2.20
&L_INSURANCEAMOUNT0=1.51
&L_SHIPPINGOPTIONISDEFAULT0=false
&L_SHIPPINGOPTIONNAME1=UPS Express 2 Days
&L_SHIPPINGOPTIONAMOUNT1=10.00
&L_TAXAMT1=2.00
&L_INSURANCEAMOUNT1=1.35
&L_SHIPPINGOPTIONISDEFAULT1=true
&L_SHIPPINGOPTIONNAME2=UPS Ground2 to 7 Days
&L_SHIPPINGOPTIONAMOUNT2=9.99
&L_TAXAMT2=1.99
&L_INSURANCEAMOUNT2=1.28
&L_SHIPPINGOPTIONISDEFAULT2=false
```


8

Payment Review

Payment Review is an Express Checkout feature that identifies high-risk transactions and notifies you so that you can hold shipments until the risk has been evaluated by PayPal.

Handling Payment Review

You are immediately notified that a payment is under review and you should not ship merchandise or, in the case of electronic media, you should not allow download access while the payment is under review. You are notified of the resolution within 24 hours.

NOTE: Payment Review is not applicable to Direct Payment.

You can determine the status of a payment in the following ways:

- By logging in to <https://www.paypal.com/> and viewing the status information in the Transaction History
- By email sent by PayPal
- By Instant Payment Notification (IPN) message
- By Payment Data Transfer (PDT) message
- By checking the status of a transaction programmatically

Programmatically, the merchant can determine the status of a payment by checking the initial status of a transaction using any of the following the API operations:

- `DoExpressCheckoutPayment`
- `DoReferenceTransaction`
- `DoAuthorization`
- `DoReauthorization`

You can check the subsequent status of a transaction programmatically by calling the `GetTransactionDetails` API operation.

NOTE: You must use version 58.0 or higher to obtain the initial status information provided by `DoExpressCheckoutPayment`, `DoReferenceTransaction`, `DoAuthorization`, or `DoReauthorization`.

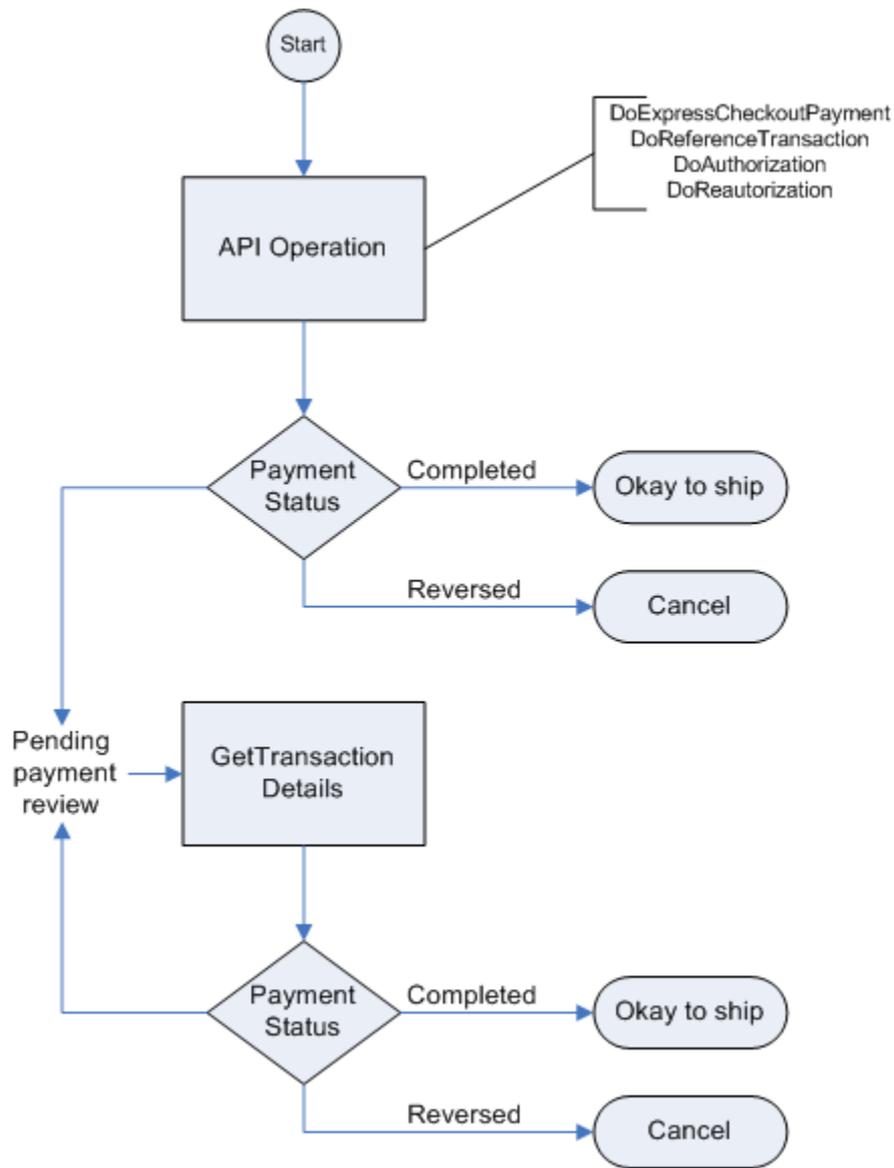
To use payment review with the `DoExpressCheckoutPayment`, `DoReferenceTransaction`, `DoAuthorization`, and `DoReauthorization` PayPal API operations, you must:

1. Check the payment status in the response to the API operation; specifically, check whether `PaymentStatus` is set to `Pending`.
2. If the `PaymentStatus` is set to `Pending`, check whether the `PendingReason` is set to `PaymentReview`, because there are other reasons that a transaction may become pending. For example, an unsettled authorization's `PaymentStatus` is set to `Pending`; however, its `PendingReason` is set to `authorization`, which is not related to payment review.

If `PaymentStatus` is set to `Pending` and the `PendingReason` is set to `PaymentReview`, you should not ship merchandise or, in the case of electronic media, you should not allow download access.

Because the payment status changes after review, you must periodically check the payment status using the `GetTransactionDetails` API operation.

The following diagram shows how to use the payment status to detect payments under review by PayPal.



IMPORTANT: For best results, call the `GetTransactionDetails` API operation every six hours. PayPal recommends not calling `GetTransactionDetails` more frequently than once per hour.

9

Express Checkout Dynamic Image Integration

PayPal hosts the PayPal buttons and logo images that you use on your website. Using only hosted buttons and images standardizes their appearance on websites that use PayPal as a payment option, which is convenient for you and gives your buyers confidence that their transaction will be safe and secure.

Dynamic Images

To use dynamic images, you must pass information to PayPal as parameters appended to the image URL. Your unique ID tells PayPal whether or not you are participating in events that require image changes. Other information you pass instructs PayPal on the types of images to return.

If, for example, you are participating in a PayPal campaign that you have signed up for with PayPal and you have passed the appropriate parameter information to PayPal, PayPal automatically updates the image to reflect the campaign information. When the campaign is over, PayPal restores the default image. You are not responsible for scheduling or making changes to your website application code before, during, or after the campaign. Because you set up the dynamic image, PayPal handles these activities for you.

If you require localized campaign images, you can have the localized button image display for each country in which you participate. Simply assign the correct code for the country to the locale parameter you append to the dynamic image URL. PayPal returns to the default button image associated with each locale when the campaign is not available.

Configuring the Dynamic Image

To set up the dynamic image, provide the name-value pair parameter information in the image URL. You can pass information in the image URL for each option.

- [Set Up the Default Image](#)
- [Set Up Image for Dynamic Use](#)
- [Change the Locale](#)
- [Provide Incentive Eligibility Feedback to Buyer](#)
- [Choose the Image](#)

Set Up the Default Image

The following URL points to the default **Check out with PayPal** image:

https://fpdbs.paypal.com/dynamicimageweb?cmd=_dynamic-image

To make the image dynamic, you need only add parameters to this URL to specify the changes you want displayed.

To test in the Sandbox environment, send the image to the following Sandbox URL:

https://fpdbs.sandbox.paypal.com/dynamicimageweb?cmd=_dynamic-image

Set Up Image for Dynamic Use

To set up the image URL for dynamic use, associate the URL with your PayPal Secure Merchant Account ID, or `pal`. Use the Secure Merchant Account ID in your Profile or call `GetPalDetails`.

This is an example call to `GetPalDetails` request.

Request Parameters

[requiredSecurityParameters]

`&METHOD=GetPalDetails`

Response Parameters

This `GetPalDetails` response returns the value of `PAL` and your country code (`LOCALE`).

[successResponseFields]

`&PAL=SFJXCXFDLNR5U`

`&LOCALE=en_US`

1. Append the `pal` parameter to the image URL, and set the parameter to the value of your encrypted PayPal merchant account number.

```
https://fpdbs.paypal.com/dynamicimageweb?cmd=_dynamic-image&pal=SFJXCXFDLNR5U
```

2. You can optionally change the value of `LOCALE`. See [Change the Locale](#) for details.
3. Place the URL with parameter information at the appropriate image locations in your web application.

The `pal` alerts PayPal to campaigns in which you are participating. PayPal obtains this information from your account and replaces the default image with the appropriate campaign image during the relevant campaign.

NOTE: If you pass in a `pal` value for a merchant account that is not yours, PayPal displays the image for that account. Be sure to pass the `pal` value for your account.

Change the Locale

To specify the locale of the image, append the `locale` parameter set to the code for the appropriate country to the image URL. If a country does not have a localized image or if you do not pass a `locale` value, the default US image displays. This example displays the image for the Spanish locale:

```
https://fpdbs.paypal.com/dynamicimageweb?cmd=_dynamic-image&pal=SFJCXFDLNFR5U&locale=es_ES
```

If you are participating in a campaign across multiple countries, you can set the image locale for each country in which you participate. PayPal returns the default image associated with the locale when the campaign is over.

Provide Incentive Eligibility Feedback to Buyer

Pass the order total amount in the `ordertotal` parameter so PayPal can determine if the buyer is eligible for an incentive. Say, for example, that you are participating in a campaign in which the buyer is eligible for a 20% discount when their order meets a minimum of \$50.00. You can pass that value to PayPal in the `ordertotal` parameter, as shown here:

```
https://fpdbs.paypal.com/dynamicimageweb?cmd=_dynamic-image&pal=SFJCXFDLNFR5U&ordertotal=50.00
```

When a buyer's order meets or exceeds \$50.00, PayPal displays the incentive image informing the buyer of their eligibility for the discount. When a buyer's order is less than \$50.00, PayPal displays the default image.

NOTE: If `ordertotal` is not passed, PayPal does not display the incentive image even if the buyer is eligible for the incentive.

Choose the Image

To specify the image that you want to display, set the value of `buttontype`. This example sets `buttontype` to the **PayPal** mark image:

```
https://fpdbs.paypal.com/dynamicimageweb?cmd=_dynamic-image&pal=SFJCXFDLNFR5U&buttontype=ecmark
```

The default value for `buttontype` is `ecshortcut`.

Dynamic Image Command Reference

To set up the information that enables dynamic images, add parameters to the dynamic image URL.

Dynamic Image Parameters

pal	<i>(Optional)</i> The encrypted PayPal account number. When merchants sign up for a PayPal business account, PayPal assigns them an account number. The pal value represents the pay-to-merchant account, not a third party making the API request on behalf of this merchant. By default, PayPal displays the default Check out with PayPal button.
ordertotal	<i>(Optional)</i> The total cost of the order to the buyer. If shipping and sales tax are known, include them in this value. If not, this value should be the current subtotal of the order. By default, PayPal does not display the incentive image even if the buyer is eligible for the incentive. Character length and limitations: Must not exceed \$10,000.00 USD in any currency. Do not specify the currency symbol. You must include two decimal places. The decimal separator must be a period (.), and the optional thousands separator must be a comma(,).
locale	<i>(Optional)</i> The five-character locale code. See Locale Codes and Priorities . Any other values default to US. NOTE: The merchant can participate in one campaign per country.
buttontype	<i>(Optional)</i> Indicates a dynamic image. The values are: <ul style="list-style-type: none"> • Check out with PayPal button image: ecshortcut (default) • PayPal mark image: ecmak

Locale Codes and Priorities

A country code is the two-letter code for the country. Language priority is the language associated with the country code, where language_0 is the default priority.

Country code	Language priority	Locale
AT	language_0	de_DE
AT	language_1	en_US
AU	language_0	en_AU
BE	language_0	en_US
BE	language_1	nl_NL
BE	language_2	fr_FR

Country code	Language priority	Locale
C2	language_0	en_US
C2	language_1	zh_XC
C2	language_2	fr_XC
C2	language_3	es_XC
CH	language_0	de_DE
CH	language_1	fr_FR
CH	language_2	en_US
CN	language_0	zh_CN
default	language_0	en_US
default	language_1	fr_XC
default	language_2	es_XC
default	language_3	zh_XC
DE	language_0	de_DE
DE	language_1	en_US
ES	language_0	es_ES
ES	language_1	en_US
FR	language_0	fr_FR
FR	language_1	en_US
GB	language_0	en_GB
GF	language_0	fr_FR
GF	language_1	en_US
GI	language_0	en_US
GP	language_0	fr_FR
GP	language_1	en_US
IE	language_0	en_US
IT	language_0	it_IT
IT	language_1	en_US
JP	language_0	ja_JP
JP	language_1	en_US
MQ	language_0	fr_FR
MQ	language_1	en_US

Country code	Language priority	Locale
NL	language_0	nl_NL
NL	language_1	en_US
PL	language_0	pl_PL
PL	language_1	en_US
RE	language_0	fr_FR
RE	language_1	en_US
US	language_0	en_US
US	language_1	fr_XC
US	language_2	es_XC
US	language_3	zh_XC

10

Immediate Payment

Immediate Payment ensures a buyer pays for a purchase immediately after committing to it.

Overview of Immediate Payment

Immediate Payment supports instant funding sources only, ensuring that you receive payment at the time the buyer commits to a purchase.

PayPal offers two applications of Immediate Payment. Use the one that is appropriate for your integration:

- **Immediate Payment for third-party checkout** – For third parties who sell items on eBay and host Express Checkout on their website.
- **Immediate Payment for Express Checkout** – For any merchant who integrates Express Checkout.

About Immediate Payment For Third-Party Checkout

Immediate Payment for third-party checkout ensures that the buyer pays for an item you are selling on eBay at the time the buyer commits to it. You add a few Express Checkout parameters to your integration and use your own off-eBay checkout flow.

NOTE: This feature is available with API version 60.0 and higher.

The `SetExpressCheckout` request parameters ensure the eBay item is available to ship to the buyer when you call `DoExpressCheckoutPayment`. The successful transaction returns the eBay transaction ID as an additional parameter value in the `DoExpressCheckoutPayment` response.

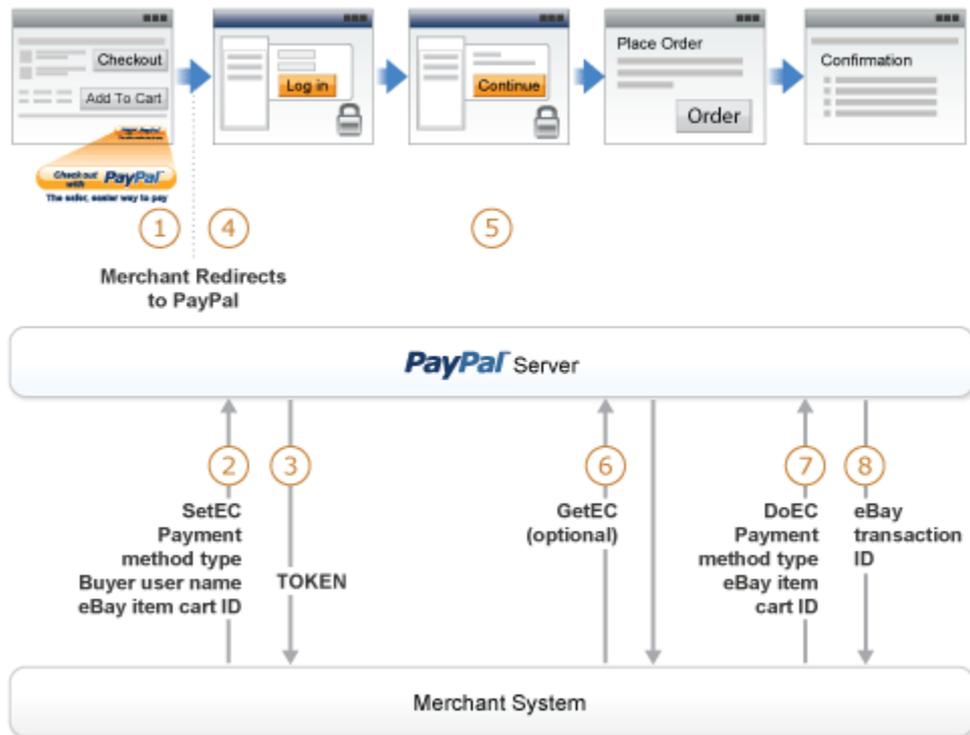
NOTE: PayPal recommends that you do not mix eBay and non-eBay items in an immediate payment transaction.

Immediate Payment for third-party checkout has the following caveats:

- By design, it does not support non-instant funds, such as Electronic Funds Transfer, eCheck, or ELV.
- It is limited to Express Checkout for eBay auctions only.
- It is limited to processing a single payment per transaction; the buyer can check out with only one item at a time.
- Sale is the only payment action supported.

- It does inventory checking at eBay and attempts to purchase the item.

After the buyer selects the item on eBay, the buyer is redirected to your website Express Checkout flow. The following figure shows how Immediate Payment for third-party checkout integrates into your checkout flow:



From left to right, the following events are represented. The numbered callouts in the figure directly correspond to the numbered comments below:

1. The eBay flow for third-party checkout redirects the buyer from an eBay payment review page to your shopping cart page. On your shopping cart page, the buyer initiates Express Checkout by clicking the **Checkout with PayPal** button.
2. In the call to the `SetExpressCheckout` API operation, you must pass the following Immediate Payment information:
 - `PAYMENTREQUEST_n_ALLOWEDPAYMENTMETHOD`: This is the payment method type. For immediate payment, the value is `InstantPaymentOnly`.
 - `BUYERUSERNAME`: eBay provides you with this value.
 - `L_PAYMENTREQUEST_n_EBAYITEMCARTIDm`: eBay provides you with this value.

Additionally, in the call to the `SetExpressCheckout` API operation, you must pass the following Express Checkout API parameter data:

- `CHANNELTYPE`: The value of the channel type, which is `eBayItem`.
- `L_PAYMENTREQUEST_n_EBAYITEMNUMBERm`: This is the eBay item number.
- Buyer's shipping address.

3. The `SetExpressCheckout` response returns a `TOKEN`.
4. The buyer is redirected to PayPal.
5. The buyer reviews their payment and clicks **Continue**.
6. (Optional) Call `GetExpressCheckoutDetails`. The `GetExpressCheckoutDetails` response returns information about the buyer.
7. In the call to `DoExpressCheckoutPayment`, you must pass:
 - `PAYMENTREQUEST_n_ALLOWEDPAYMENTMETHOD`: This is the payment method type. For immediate payment, the value is `InstantPaymentOnly`.
 - `L_PAYMENTREQUEST_n_EBAYITEMCARTIDm`: eBay provides you with this value.At this time, PayPal checks the availability of the eBay item.
8. If the item is available and payment is successful, the `DoExpressCheckoutPayment` response returns the eBay transaction ID (`PAYMENTREQUEST_n_EBAYITEMAUCTIONTXNID`).

Integrating Immediate Payment for Third-Party Checkout

To integrate Immediate Payment for third-party checkout into your Express Checkout implementation, you add a few new fields to `SetExpressCheckout` and `DoExpressCheckoutPayment`.

The Call to `SetExpressCheckout`

You must have Express Checkout integrated into your payment solution. PayPal recommends that you sell one eBay item in a transaction or a quantity of the same item (considered a *single eBay listing*).

To integrate Immediate Payment into the `SetExpressCheckout` call:

1. Set `PAYMENTREQUEST_n_ALLOWEDPAYMENTMETHOD` to `InstantPaymentOnly`.
This blocks all pending funding sources and transactions in a pending state.
2. Pass `BUYERUSERNAME` and `L_PAYMENTREQUEST_n_EBAYITEMCARTIDm`.
3. Set `CHANNELTYPE` to `eBayItem`.
4. Set `L_PAYMENTREQUEST_n_EBAYITEMNUMBERm` to the number of the eBay item.
5. Specify the buyer's shipping address.
6. Be prepared to handle Immediate Payment errors.

For errors you choose to handle, you need to send the buyer error messages appropriate to the situations generating the errors. For example, a buyer may have a mixed cart of eBay and non-eBay items. To allow the buyer to check out *only* the non-Immediate Payment items, you must provide an appropriate message to the buyer, remove the Immediate Payment item, and retry the transaction.

The Call to DoExpressCheckoutPayment

To integrate Immediate Payment into the DoExpressCheckoutPayment call:

1. Set PAYMENTREQUEST_n_ALLOWEDPAYMENTMETHOD to InstantPaymentOnly.

This blocks all pending funding sources and transactions in a pending state.

2. Pass the L_PAYMENTREQUEST_n_EBAYITEMCARTIDm.
3. Be prepared to handle Immediate Payment errors.

For errors you choose to handle, you need to send the buyer error messages appropriate to the situations generating the errors. For example, you determine that the buyer has a mixed cart of eBay and non-eBay items. To allow the buyer to check out *only* the non-Immediate Payment items, you must provide an appropriate message to the buyer, remove the Immediate Payment item, and retry the transaction.

If payment is successful, the DoExpressCheckoutPayment response returns the eBay transaction ID (PAYMENTREQUEST_n_EBAYITEMAUCTIONTXNID).

About Immediate Payment For Express Checkout

Immediate Payment for Express Checkout ensures the buyer pays for purchases at the time the buyer commits. This application is available to any merchant integrating Express Checkout.

NOTE: This feature is available with API version 63.0 and higher.

Immediate Payment for Express Checkout has the following caveats:

- By design, it does not support non-instant funds, such as Electronic Funds Transfer, eCheck, or ELV.
- It is open for use by all merchants, whether or not they are selling on eBay.
- Sale is the only payment action supported.

Integrating Immediate Payment for Express Checkout

Immediate Payment for Express Checkout requires that you specify one value for `ALLOWEDPAYMENTMETHOD` in the Express Checkout API.

1. In the call to the `SetExpressCheckout` API operation, you set `PAYMENTREQUEST_n_ALLOWEDPAYMENTMETHOD` to `InstantPaymentOnly`.
2. (Optional) You can call `GetExpressCheckoutDetails` to obtain information about the buyer.
3. In the call to `DoExpressCheckoutPayment`, you set `PAYMENTREQUEST_n_ALLOWEDPAYMENTMETHOD` to `InstantPaymentOnly`.

Revision History

Revision history for *PayPal Express Checkout Advanced Features Guide*.

Date	Description
03/22/2012	Updated graphics and information related to account optional checkout on mobile devices.
02/13/2012	Updated user experience graphics.
06/21/2011	Added chapters on recurring payments and reference transactions and changed the order of chapters in this guide.
05/02/2011	Revised Express Checkout on Mobile Devices chapter and added support information for iPad.
08/11/2010	Divided the <i>Express Checkout Integration Guide</i> into 2 guides: the <i>Express Checkout Integration Guide</i> and the <i>Express Checkout Advanced Features Guide</i> .
05/11/2010	Added details for integrating parallel payments using the NVP and SOAP API, including use with airlines. Added new Immediate Payment functionality. Updated billing agreements to obtain the latest billing address, skip billing agreement creation, and clarified the use of the BAUpdate API.
03/10/2010	Added support for parallel payments.
01/21/2010	Added new Express Checkout fields to provide the buyer contact information, gift options, promotions, and a survey question on the PayPal pages. Added a new callback response API field providing no-shipping details.
03/10/2010	Added support for parallel payments.
10/05/2009	Added Immediate Payment. Removed PayPal placement guidelines.
06/30/2009	Added a section on payment review.
06/04/2009	Added a chapter on filling out the PayPal review page automatically. Updated PayPal Review pages. Moved some customization topics out of this guide. They are now in the <i>Merchant Setup and Administration Guide</i> .
04/30/2009	Created first edition for Enterprise-level merchants and added chapter on reference transactions.
04/08/2009	Added a chapter describing the Instant Update Callback API.
03/03/2009	Updated to allow useraction=continue for eBay incentives.
11/13/2008	Added information about integrating dynamic images and added information about order details that can be displayed on the PayPal Review page.



Date	Description
06/30/2008	Complete revision.