Standard Checkout

XML Implementation Guide

India Version 2015-01-27

Contents

Overview of Pay with Amazon

Welcome to the Standard Checkout - XML Implementation Guide.

This document gives you the information and steps you need to integrate your website with Pay With Amazon.

Prerequisites

This guide assumes you--or your development staff--are familiar with using HTML and XML in an application development environment.

Before you can use Pay with Amazon, your systems must meet the following prerequisites:

- You must have a shopping cart on your site and make it available for processing using XML. That is, no matter what tool you use to build your shopping cart, you must be able to send it as XML for processing.
- You must be familiar with using JavaScript. There are several required JavaScript code snippets that you must use in your carts to enable Standard Checkout.
- Your cart XML, when submitted for processing, must use the XMLNS named http://payments.amazon.com/checkout/2009-05-15
- You must be using the order.xsd with the file date of 2009-05-15 or later. (Download this version here.)

Your Tasks When Using Pay with Amazon

You have four basic tasks to accomplish when using Pay with Amazon on your site:

1. Set up your Pay with Amazon account.
2. Complete your Seller Central settings.
3. Integrate your site with Pay with Amazon.
4. Test and sign off your integration.

The rest of this document discusses how to accomplish these tasks.

Notice About Pay with Amazon Services Functionality

The Pay with Amazon service has been designed and developed for use within a web browser only. Our service cannot be used within a native application (including, without limitation, iOS, Android, RIM, or Windows operating systems). Pay with Amazon reserves the right to suspend the Payment Account of any user of our services that has implemented our Services within a native application

Chapter 1 - Setting Up Your Pay with Amazon Account

Topics:

You have six required tasks (and three optional, but recommended, tasks) when you initially set up your Pay with Amazon account on Seller Central.

In this section you'll learn about the following required tasks:

1. Sign up for your Pay with Amazon account.
2. Switch between the Production and Sandbox modes.
3. Set up your Info & Policies.
4. Set up your Integration Settings.
5. Set up your basic settings for Shipping Rates, Tax Rates, and Promotions.
6. Set up your Buyer (testing) Account.

You'll also learn about the following optional, but recommended, tasks:

1. Set up your Role-based Permissions.
2. Modify your e-mail client's whitelist to accept e-mail from Pay with Amazon.
3. Get your Security Keys, used for signed carts.

Signing Up for Pay with Amazon

To get started integrating your website with Pay with Amazon, you must first sign up for your Pay with Amazon account.

Logging In to Your Pay with Amazon Account

After you've signed up for your account, you use your account credentials (your e-mail and password) to sign in to Seller Central to manage your account.

Seller Central is the "homepage" for your account. It's where you set up your basic settings for your account, such as your info & policies, or your Integration Settings experience, and so on.

In Seller Central, you can perform the following actions:

- Switch between the Sandbox and Production
- View and fulfill orders
- View your payments
- Edit your account settings

Log In to Seller Central

You log in to Seller Central when you want to manage orders, view reports, or manage your account settings.

Note: You can also manage orders and view reports programmatically. For more information about how to do so, please see Managing Orders.

Follow these steps to log in to Seller Central:

1. Go to https://sellercentral.amazon.in.
   You'll see the Seller Central log-in page.
2. Enter your Seller Central account credentials (the e-mail and password you used to sign up for the Pay with Amazon account).
3. Click **Sign in**.
   You'll see the Seller Central home page.

For more information about using Seller Central, please download and read Using Seller Central.

Your Sandbox and Production Settings

After you have signed in to Seller Central you are ready to edit your account settings.

However, before you begin setting up your account, it's important for you to understand that there are two basic methods of operation, Sandbox and Production.

- You use the **Sandbox** for testing your setup and your code. Nothing you do in the Sandbox generates paid orders.
- You use **Production** to generate orders. Anything you do in Production is treated as a real order--you must either fulfill or cancel all Production orders within 30 days.

The basic difference between the Sandbox and Production is that anything you do in the Sandbox cannot generate an order that is charged to a payment method. That is, you can "buy" an item in the Sandbox mode, but the credit card you use to buy the item can never be charged for the item. If you buy an item in Production, you are placing a real order that must be either fulfilled or canceled--that is, you must either ship it or cancel the order.

Important: You cannot use your merchant account (the account you use to sign in to Seller Central) to place an order in Production. If you want to place orders in Production so you can test the end-to-end experience and make sure your fulfillment systems are working, you must create and use a separate "buyer" account. See Seeting Up Your Buyer (Testing) Account for more information about setting up and using a buyer account.

This table illustrates the functions available in the Sandbox and in Production.

Table 1: Functions Available in the Sandbox and in Production

Function	Sandbox	Production	Notes
Generate orders	Yes	Yes	
Build functional Checkout buttons	Yes	Yes	"Functional" here means that you can click the Checkout button and go through the order pipeline
Create an order using a payment method	Yes	Yes	You can use your payment method in the Sandbox, but your payment method will never be charged
View orders	Yes	Yes	All orders in the Sandbox are marked as "Pending.
Fulfill orders	No	Yes	All orders in the Sandbox are marked as "Pending"; you cannot fulfill "Pending" orders
Give feedback on an order	No	Yes	
Cancel an order as a customer	Yes	Yes	
Cancel an order as a merchant	No	Yes	
Have a payment method charged for an order	No	Yes	
View and download order- related reports	No	Yes	

When you set up your account, **you must set up your Sandbox and Production settings separately**. To set them up, you must switch between Production and Sandbox mode in Seller Central.

Note: When you first set up your Pay with Amazon account, you are in the Production mode. You must explicitly switch to the Sandbox mode.

The mode you pick (Production or Sandbox) is "sticky"; that is, if you select a mode, log off from Seller Central, and then log in again, you return to the same mode.

Switching Between Production and the Sandbox

Follow these steps to switch between Production and the Sandbox:

1. Locate the **Website** drop-down box near the top left.
2. Click the **Website** drop-down box to display the available websites.
3. To select the Production mode, click **Pay With Amazon -IN (Production View).**
4. To select the Sandbox mode, click **Pay With Amazon - IN (Sandbox View)**.

Your Account and Your Info & Policies

Your Pay with Amazon account has two separate account settings that you set up before you accept your first order:

- **Your Account Settings**: Used to set up your account settings we use to transact business with with you, such as your business name or your bank account information.
- **Your Info & Policies Settings**: Used to set up the information your customers see about you, such as your refunds and return policy

Note: Be sure you switch to the correct mode before you change these settings. To learn more about switching between modes, please see Your Sandbox and Production Settings in this document.

Set Up Your Account Info

The Account Information page lets you change the following options:

Selling Plan

- Pay With Amazon.com
- Pay With Amazon Sanbox
- Amazon Product Ads

Listing Status

- Current Status of Listings

Seller Information

- Your business name
- Customer service e-mail
- Customer service reply to e-mail
- Customer service phone

- Contact email address
- Exceptions

Product Status

- Pay with Amazon
- Pay with Amazon Sandbox

Place of Establishment Information

- Legal name
- Place of Establishment address

Exceptions

Deposit Method

- Bank account information

Company logo

- Current logo

Website Domain Address

- Website domain address

Bank Account Information

Important: It is **critical** that you enter your bank account information in your Account Settings page in Production--we cannot pay you unless you set up a valid Indian bank account.

Note: Changing your Bank Account information will start an automatic 14-day hold on disbursements. See How often am I paid? for more information

Follow these steps to set up your account settings:

1. Verify that you have selected the correct mode, Production or Sandbox.
   Note: Be sure you switch to the correct mode before you change these settings. To learn more about switching between modes, please see Your Sandbox and Production Settings in this document.
2. In Seller Central, click **Settings > Account Info**.
   The Account Info page appears.
3. Verify that your information is correct for your Seller Information, your Place of establishment, your Bank Account, your company logo, and your website address.
   To change the information, click the **Edit** button in the appropriate section, make the necessary changes, and click **Submit**.

Set Up Your Info & Policies Settings

The Your Info & Policies page lets you change the following attributes about your company. This information appears in the Integration Settings:

Page link	Used for
About Seller	Tell your customer about yourself.
Seller Logo	Add your company logo 120 x 30 pixels (GIF, JPG, or PNG).
Returns & Refunds	Describe your returns and refund policies.
Shipping	By default, your shipping costs page in the checkout pipline shows your sipping consts (from Seller Central Manage Shipping). However, you can use this page to describe more about your shipping policies.
Privacy Policy	Describe your privacy policy.
Frequently Asked Questions	List any frequently asked questions and answer you might have.
Custom Help Pages	Add any additional help pages you'd like customers to see if they want more information within the Integration Settings.

Important: Changes you make to the pages on the Your Account & Policies page take about 24 hours to appear.

Follow these steps to set up your Info & Policies settings.

1. Verify that you have selected the correct mode, Production or Sandbox.
   Note: Be sure you switch to the correct mode before you change these settings. To learn more about switching between modes, please see Your Sandbox and Production Settings in this document.
2. In Seller Central, click **Settings > Your Info & Policies**.
   The Your Information & Policies page appears.
3. For each section in the Your Information & Policies page, click the link to that page, enter the information, and click **Save**.
   If you don't see a page you want, you can create Custom Help Page with the content you specify.

Your Integration Settings

With Pay with Amazon, you can set the following options for your Integration Settings:

- Set the URL for successful payments (the page on your site where customers land after completing their purchase).
- Set the URL for canceled payments (the page on your site where customers land if they cancel the order before completing their purchase).
- Set the URL for your customers' account information (the page on your site where you show customers their account information)
- Set the URL for instant order processing notifications (useful if you integrate using the Instant Order Processing Notification API; see the Instant Order Processing Notification API Guide (PDF) for more information).

Set the Return URLs

Follow these steps to set the three Return URLs.

Note: The Your Account Return URL points to a location on your site where you display information to your customers about their accounts. Customers can go to the Pay with Amazon Your Account page (https://paywithamazon.amazon.in) to see their order status for all payments made using Pay with Amazon anywhere on the internet.

1. In Seller Central, click **Settings** > **Integration Settings**.
   The Integration Settings page appears.
2. Click the first **Edit** button on the page (the one immediately on the right side of Pay With Amazon).
3. In the **Successful Payment Return URL** box, type the URL you want to send your customers to after they successfully complete payment.
4. In the **Cancel Payment Return URL** box, type the URL you want to send your customers to if they cancel the order before they successfully complete payment.
5. Click **Update**.

Set the Integration URLs

If you (or your integrator/developer) are using the Instant Order Processing Notifications, you must specify where you want these notifications to go. You can send them to your own website or to your integrator's website.

Note: Choose either your own website or your integrator's website, but not both, or else you will get multiple order processing messages

1. In Seller Central, click **Settings** > **Integration Settings**.
   The Integration Settings page appears.

2. Click the first **Edit** button on the page (the one immediately under the Instant Order Processing Notification Settings heading).
    a. In the **Merchant URL** box, type the URL (on your website) where you want notifications to be sent.
    b. Or, in the **Integrator URL** box, type the URL (on your integrator's or developer's website) where you want notifications to be sent.
3. Click **Update**.

Set the Order Report Type and Frequency

You can set your order report type and frequency in Seller Central to match your needs.

1. In Seller Central, click **Settings** > Integration Settings.
   The Integration Settings page appears.
2. Click the second **Edit** button on the page (the one on the right side of Pay With Amazon heading).
   The Order Reports Setting page appears.
3. Next to **Order Report Type,** click the radio button for the type of report you want.
   If you are working with an integrator, they will give you the correct format to choose.
4. Next to **Order Report Schedule,** pick the frequency of your reports.
5. Click **Submit**.

Basic Seller Central Settings for Shipping and Promotions

Your Pay with Amazon account uses Seller Central to manage basic settings for shipping rates and promotions.

The rates and options you set in Seller Central determine what shipping rates apply to orders processed using Pay With Amazon.

There are a few default settings already applied when you first create your Pay With Amazon account, but it's a good idea to review them to be sure that you know what to expect when you begin accepting orders.

Set Your Seller Central Shipping Rates

Follow these steps to review and set your Seller Central Shipping Rates:

1. In Seller Central, go to **Settings** > **Shipping Settings**.
   The Shipping Settings page appears, showing your current shipping rates.

Tip: For complete information about setting shipping rates, please see Configure Shipping Rates & Restrictions in Seller Central.

Set Your Seller Central Promotions

Follow these steps to review and set your Seller Central Promotions (discounts):

1. In Seller Central, go to **Settings** > **Manage Promotions**.
   The Manage Promotions page appears, showing your current promotions.
2. You can change the following options:

   - Add a fixed-amount-off promotion
   - Add a percent-off promotion
   - Add a free-shipping promotion
   - Add a tiered promotion
   - Add a fixed-price promotion

Tip: For complete information about setting promotions, please see Offer Promotions in Seller Central.

Setting User Permissions for Your Seller Central Account

When you first set up your Seller Central account, only the original registered user can access the features of Seller Central. But as time goes by, you might find that you want others to access your account. For example, you might want a co-worker to manage your orders or to handle your shipping confirmations. To add users or adjust user permissions, you use the User Permissions feature.

Seller Central uses an invitation model to manage user accounts. First, users are invited to create an account on Seller Central to access your storefront; then, the account manager (also called the account administrator) configures the permissions for that user. By inviting users into the system and then configuring their permissions, you are assured that the appropriate user accounts are linked to the right owners, and that the correct permissions apply to the intended users. The invitation model also helps you to fully manage user accounts without assistance from an Pay with Amazon administrator.

To set up users, first the account manager must set up the initial account using the Pay with Amazon registration process. After setting up the initial account, the account manager then compiles a list of people who can access the system and determines the respective rights for those users.

At this point, the account manager is ready to invite new users to Seller Central. This invitation process takes three steps:

1. The account manager sends an e-mail invitation to each user that contains a link to Seller Central.
2. The user clicks the link and navigates to a page giving access to a unique confirmation code. *At this stage, the user is a pending user.*
3. The user enters the confirmation code and establishes the account. *At this stage, the user is now a current user.*

After the invitation process is complete, the user has a username and password to access the system.

The final step is for the account administrator to assign the appropriate rights to that user.

Note: The username and password are combined together to identify a unique user account. For example, *myname@isp.com + password1* and *myname@isp.com + password2* are different user accounts.

Step One: The Account Manager Sends the Invitation

The first step is to invite the user.

1. On the **Settings** tab, click **User Permissions**.
   The User Permissions page appears.
2. Under **Add a New Seller Central User,** type the e-mail address of the user you want to invite to the account.
3. Click **Send Invitation**.
   The e-mail invitation is sent to the e-mail address you specified. A confirmation message appears.
4. Click **Continue**.

Step Two: The User Receives a Confirmation Code

After clicking the link in the invitation e-mail, the invited user navigates to a web page that asks the user to either sign in with an existing Amazon account or to create a new account. After this is done, a page appears containing a confirmation code. The user then shows or sends the confirmation code to the account manager, and the account manager approves the user and grants rights to the user.

Step Three: The User Accesses Seller Central and Creates an Account

After the account manager confirms the user, the user can go to https://sellercentral.amazon.in/ to log in to the account.

Step Four: The Account Manager Sets User Permissions

After creating the user, the account manager can edit user permissions, granting rights to any and all tools.

1. On the **Settings** tab, click **User Permissions**.
   The User Permissions page appears.
2. On the **User Permissions** page, click **Edit** next to the account you want to change.
   The Add or Edit User page appears.
3. On the **Add or Edit User** page, click the button next to each tool you want to allow the user to access.
4. When you are finished, click **Continue**.
   A confirmation message appears confirming the modifications.

Note: The original account holder -- ithe one who first signed up for the account -- is always the default account manager. The account manager can always grant rights to others to be co-account managers;

however, the original owner/account manager cannot be changed or removed.

Getting Your Key

Your keys are used for signing your carts (orders) and are unique to your account. You get (and use) two different keys:

- Your Access Key ID from Seller Central (public)
- Your Secret Access Key from Seller Central (private)

For more information about signing your carts, please see Sign Your Cart.

Get Your Keys

Important: Keep your Secret Access Key in a safe place -- do not share it with unauthorized people, and never include it in an e-mail message or in the order code you develop. Use the Secret Access Key only to sign orders.

Follow these steps to get your keys

1. In Seller Central, go to **Integration > Access Key**.
   The Access Key page appears, showing your Access Key ID and a placeholder for your Secret Access Key.
2. To view your **Secret Access Key**, click **Show**.
   Your Secret Access Key appears.

Setting Up Your Buyer (Testing) Account

Before you begin taking orders on your website, you should test your order and fulfillment systems to be sure that you can reliably accept and fulfill orders. And when you create test orders, you must use a separate account from the one you use when you signed up for Pay with Amazon.

Note: For more information about testing procedures, please see Sandbox Setup.

Be sure you understand that orders placed in the Production environment must be either fulfilled by you as the merchant or canceled using the buyer account. See Cancel Your Test Production Order for more information.

Set Up Your Buyer (Testing) Account

Important: Do not use the same e-mail address for your buyer (testing) account that you are using for your seller or merchant account. Pay with Amazon allows only one account per e-mail account. That is, if you use someone@mail.com for your seller account, don't use the same e-mail address when you create a new buyer (testing) account.

Note: If you already have a separate buyer account (different than the one you used to sign up for Pay with Amazon) that you use for buying on Amazon.com, you can skip these steps.

Follow these steps to set up your buyer (testing) account:

1. Go to the Pay with Amazon site at https://paywithamazon.amzon.in/
2. Click the **Personal** tab.
   The Personal tab page appears.
3. Enter your e-mail address, click **I am a new user**, and click **Sign in using our secure server**.
   You should see the New to Amazon.com page. If you do not, press the Back button on your browser and be sure you enter an e-mail and password that you have not used before to shop on Amazon.in.
4. Fill in the requested information and click **Continue**.
   The Sign Up for Pay with Amazon page appears.
5. Fill in the requested information and click **Continue**.

After you have signed up for an account, we'll send an e-mail to the address you supplied with instructions for you to confirm your new account.

Chapter 2 - Integrating with Pay with Amazon

Topics:

- *Steps to Integrating Pay with Amazon with Your Website*
- *Integration Touch Points*
- *Sandbox Setup*
- *Production Setup*
- *Set Up for Your Mobile- Optimized Website*

Integrating your website with Pay with Amazon requires that you build and test your buttons (and checkout process) in the Sandbox as well as in Production.

The following section is a step-by-step procedure to create working checkout buttons, from your test buttons that you build and use in the Sandbox to working buttons you build and use on in Production.

Steps to integrating Pay with Amazon with your website

Here are the steps you need to follow to integrate with Pay With Amazon:

1. Download the XML schema (order.xsd).
2. Create the cart using Sandbox URLs.
3. Sign the cart.
4. Base64-encode the cart.
5. Place the resulting Sandbox Standard Checkout Widget on your test web page.
6. Test the Sandbox order using your buyer account (complete the checkout in the Sandbox).
7. Create the cart using Production URLs.
8. Sign the cart.
9. Base64-encode the cart.
10. Add the optional signature to the Production Standard Checkout Widget.
11. Place the Production Standard Checkout Widget on your test web page.
12. Test the Production order using your buyer account (complete the purchase in Production).
13. Cancel the Production order on Pay with Amazon site using your buyer account login.
14. Place the Production-mode Standard Checkout Widget on your website.

Integration Touch Points

There are three main steps in the checkout workflow:

1. The customer places the order with the merchant.
2. The merchant processes the order with Pay with Amazon using Pay with Amazon. This step includes fulfilling (shipping) the order. When an order is marked as "shipped," the customer's account is charged.
3. The merchant manages the payments made to their merchant account.

During this process, there are several steps taking place:

1. When the customer places an order, a checkout request is sent to Pay with Amazon.
2. This generates a callback request to the merchant (if callbacks are enabled).
3. The merchant sends back a callback response with the requested data.
4. Pay with Amazon sends the customer to the specified "Thank You" page.
5. Pay with Amazon sends a New Order Notification.
6. When the payment method from customer is processed, Pay with Amazon sends an Order Ready for Fulfillment Notification.
7. The merchant pulls (views or downloads) the Order Report
8. The merchant notifies Pay with Amazon that the order is fulfilled. At this point, Pay with Amazon charges the customer's payment method.
9. Pay with Amazon settles payment to the merchant's bank account.
10. The merchant pulls (views or downloads) the Settlement (Payment) Report.

Sandbox setup

Download the Current XML Schema

Building an XML-based cart requires you to download and use the current order.xsd schema. The current schema version is 2009-05-15

Follow these steps to download the current schema

Download the following file: http://amazonpayments.s3.amazonaws.com/documents/order.xsd

Create a Sandbox Cart Using XML

Building a Sandbox cart lets you create test orders.

A simple XML-based cart contains the elements as listed in the example below.

Element Name	Description	Required	XPath
Order	Describes order schema	Yes	Root
Cart	Container for entries in cart	Yes	\Order\Cart
Items	Container for item entries	Yes	\Order\Cart\Items
Item	Container for elements of the order 1-999 item entries per cart	Yes	\Order\Cart\Items\Item
SKU	Stock keeping ID for the item 1 to 250 characters.	Yes. If using Fulfillment by Amazon, the SKUs must match	\Order\Cart\Items\Item\SKU
MerchantId	14-character alphanumeric string Get from **Settings** > **Account Info** > **Integration Settings**	Yes	\Order\Cart\Items\Item\MerchantId
Title	Name of Item 1 - 80 alphanumeric characters only	Yes	\Order\Cart\Items\Item\Title
Price	Container for Amount, CurrencyCode	Yes	\Order\Cart\Items\Item\Price
Amount	Cost of item	Yes	\Order\Cart\Items\Item\Price\Amount
CurrencyCode	Type of currency	Yes	\Order\Cart\Items\Item\Price \CurrencyCode

The following is an example of a simple Sandbox cart using XML.

```
<?xml version="1.0" encoding="UTF-8"?>
 <Order xmlns="http://payments.amazon.com/checkout/2009-05-15/">
   <Cart>
     <Items>
       <Item>
         <SKU>ABC123</SKU>
         <MerchantId>AEIOU1234AEIOU</MerchantId>
         <Title>Red Fish</Title>
         <Price>
           <Amount>19.99</Amount>
           <CurrencyCode>INR</CurrencyCode>
         </Price>
         <Quantity>1</Quantity>
       </Item>
     </Items>
   </Cart>
   </Order>
```

Note: Do not include / (forward slash) before Item Title value.

Sign Your Cart

To generate a signature, you need a set of signing-key pairs. By default, your Pay with Amazon account supplies you with a set of Access Keys from Seller Central, the Access Key ID (the public key) and the Secret Access Key (the private key). See *Getting Your Key* on page 24 for steps to take to get your Access Keys.

Once you have your signing-key pairs, you compute an RFC2104-compliant HMAC signature of the concatenated cart string. No canonicalization is required.

The following table lists the types of signature and their HTML representation in the cart.

Name	Description	Type of Cart
type	One of the following: • merchant-signed-order/aws-accesskey/1 (for Access Key signatures) • unsigned order	All types
order	base64-encoded order	All types
signature	Signature calculated according to type	merchant-signed-order/aws-accesskey/1
aws-access-key-id	Access Key ID retrieved in Seller Central	merchant-signed-order/aws-accesskey/1

Follow these steps to sign your XML-based carts:

1. Get your Access Keys.

2. Get the XML code used for your cart.

Note that the XML code you sign must be identical in content and layout (well-formatted or not) you use later in the step "Base64-Encode Your Cart."

3. Using the programming language of your choice, sign the XML code with your key.

4. Using the programming language of your choice, base64-encode the resulting signature.

The resulting encoded signature will look something like this: `0nppbXTktfoV80Kh41GK7ruUhq8=`

Base64-Encode Your Cart

After you get the XML code for your cart, you base64-encode the XML code using the programming language of your choice.

If you permit only signed carts, you also must base64-encode the signature before you add it to your cart form. Note that you add the signature to the form after you have signed and base64-encoded the cart.

Here is an example of a signed cart, signed using the Access Key pair (line breaks added for clarity):

```
<html>
...
<head>
<script language=javascript
    src=https://payments-sandbox.amazon.com/cba/js/PaymentWidgets.js>
</script>
...
</head>
<body>
...
<div id="cbaButton1">
  // Replace "AEIOU1234AEIOU" with your own merchantId value
  <img src="https://payments.amazon.com/gp/cba/button?
cartOwnerId=AEIOU1234AEIO
U&color=orange&size=large&background=white"/>
</div>

<script>
  new CBA.Widgets.StandardCheckoutWidget({
    merchantId:'AEIOU1234AEIOU',
    orderInput: {format: "XML",
    value: "type:merchant-signed-order/aws-
     accesskey/1;order:PD94bWwgdmVyc2lvbj0nMS4wJyBlbmNvZGlu
    Zz0nVVRGLTgnPz48T3JkZXIgeG1sbnM9J2h0dHA6Ly9wYXltZW50cy5
    hbWF6b24uY292NoZWNrb3V0LzIwMDgtMTEtMzAvJz48Q2FydD48SXRl
    bXM+PEl0ZW0+PFNLVT5BQkMxMjM8L1NLVT48TWVyY2hhbnRJZD5BMUV
    KTjJKSUcxMEs2NzwvTWVyY2hhbnRJZD48VGl0bGU+UmVkIEZpc2g8L1
    RpdGxlPjxQcmljZT48QW1vdW50PjE5Ljk5PC9BbW91bnQ+PEN1cnJlb
    mN5Q29kZT5VU0Q8L0N1cnJlbmN5Q29kZT48L1ByaWNlPjxRdWFudGl0e
    T4xPC9RdWFudGl0eT48RnVsZmlsbG1lbnROZXR3b3JrPk1FUkNIQU5UP
    C9GdWxmaWxsbWVudE5ldHdvcms
    +PC9JdGVtPjwvSXRlbXM+PC9DYXJ0PjwvT3JkZXI
    +;signature:0nppbXTktfoV80Kh41GK7ruUhq8=
    ;aws-access-key-id:19G7E7X2QE2V45LET"},
    }).render("cbaButton1");
</script>
...
</body>
</html>
```

And here is an example of an unsigned cart:

```
<html>
...
<head>
<script language=javascript
    src=https://payments-sandbox.amazon.com/cba/js/PaymentWidgets.js>
</script>
...
</head>
<body>
...
<div id="cbaButton1">
  // Replace "AEIOU1234AEIOU" with your own merchantId value
  <img src="https://payments.amazon.com/gp/cba/button?
cartOwnerId=AEIOU1234AEIOU&color=orange&size=large&background=white"/
>
</div>

<script>
  new CBA.Widgets.StandardCheckoutWidget({
    merchantId:'AEIOU1234AEIOU',
    orderInput: {format: "XML",
    value:"type:unsigned-
    order;order:PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4KPE9yZGVyI
    HhtbG5zPSJodHRwOi8vcGF5bWVudHMuYW1hem9uLmNvbS9jaGVja291dC8yMDA4LTA
    2LTE1LyI+CiAgPENsaWVudFJlcXVlc3RJZD5PcmRlci0yNDwvQ2xpZW50UmVxdWVzdElkPg
    ogIDxFeHBpcmF0aW9uRGF0ZT4yMDA4LTEyLTEeHBpcmF0aW9uRGF0ZT4KICA8QFydD4
    KICAgIDxJdGVtcz4KICAgICAgPEl0ZW0+CiAgICAgICAgPFNLVT5UU0hMDAxPC9TS1U+CiAgI
    CAgICAgPE1lcmNoYW50SWQ+PC9NZXJjaGFudElkPgogICAgICAgIDxUaXRsZT5BbWF6b24g
    UGF5bWVudHMgVC1TaGlydDwvVGl0bGU+CiAgICAgICAgPFByaWNlPgogICAgICAgICAgPE
    Ftb3VudD45Ljk1PC9BbW91bnQ+CiAgICAgICAgICA8Q3VycmVuY3lDb2RlPlVTRDwvQ3Vyc
    mVuY3lDb2RlPgogICAgICAgIDwvUHJpY2U+CiAgICAgICAgPFF1YW50aXR5PjE8L1F1YW50a
    XR5PgogICAgICA8L0l0ZW0+CiAgICA8L0l0ZW1zPgogIDwvQ2FydD4KPC9 cj4K"},
    }).render("cbaButton1");
</script>

...
```

```
  </body>
  </html>
```

Follow these steps to base64-encode your cart.

1. Get your Access Keys.
2. Get XML code used for your cart.
   Note that the XML code you sign **must be identical** in content and layout (well-formatted or not) to the code you used earlier in the Step "Sign Your Cart."
3. Using the programming language of your choice, sign the XML code with your key.
4. Using the programming language of your choice, base64-encode the resulting signature. The resulting encoded signature will look something like this:
   `0nppbXTktfoV80Kh41GK7ruUhq8=`

Add the Signature to the Cart

You can add this signature to your cart.

Signature type	Value entry	Keyword entry
MWS Access Keys	value="type:merchant-signed-order/aws-accesskey/1	aws-access-key-id

Follow these steps to add the signature to the HTML form.

Add to the orderinput tag the following values:

```
value="type:merchant-signed-order/aws-accesskey/1; ...
signature:[Signature value]; aws-access-key-id:[Access Key ID value]
```

Your form should look like the following (line breaks added for clarity):

```
      <script>
        new CBA.Widgets.StandardCheckoutWidget({
          merchantId:'AEIOU1234AEIOU',
          orderInput: {format: "XML",
          value: "type:merchant-signed-order/aws-accesskey/1;
          order:PD94bWwgdmVyc2lvbj0nMS4wJyBlbmNvZGluZz0nVVRGL
          TgnPz48T3JkZXIgeG1sbnM9J2h0dHA6Ly9wYXltZW50cy5hbWF6
          b24uY29tL2NoZWNrb3V0LzIwMDgtMTEtMzAvJz48Q2FydD48SXR
          lbXM+PEl0ZW0+PFNLVT5BQkMxMjM8L1NLVT48TWVyY2hhbnRJZD
          5BMUVKTjJKSUcxMEs2NzwvTWVyY2hhbnRJZD48VGl0bGU+UmVkI
          EZpc2g8L1RpdGxlPjxQcmljZT48QW1vdW50PjE5Ljk5PC9BbW91
          bnQ+PEN1cnJlbmN5Q29kZT5VU0Q8L0N1cnJlbmN5Q29kZT48L1B
          yaWNlPjxRdWFudGl0eT4xPC9RdWFudGl0eT48RnVsZmlsbG1lbn
          ROZXR3b3JrPk1FUkNIQU5UPC9GdWxmaWxsbWVudE5ldHdvcms+P
        C9JdGVtPjwvSXRlbXM+PC9DYXJ0PjwvT3JkZXI+;signature:
        0nppbXTktfoV80Kh41GK7ruUhq8=
        ;aws-access-key-id:19G7E7X2QE2V45LET"},
        buttonSettings:
        {size:'large',color:'orange',background:'white'}
        }).render("cbaButton1");
    </script>
```

Place the Sandbox Checkout Widget Code on Your Test Webpage

When you have built the XML code for the cart, signed it (if you accept only signed carts), and base64-encoded the cart value (and the signature, if you accept only signed carts), you add the resulting cart value and signature to an HTML form that displays the Checkout button.

For example, given the example earlier, the resulting code will look something like this (line breaks added for clarity):

```
    ...
    <script>
      new CBA.Widgets.StandardCheckoutWidget({
        merchantId:'AEIOU1234AEIOU',
        orderInput: {format: "XML",
        value: "type:merchant-signed-order/aws-accesskey/1;
        order:PD94bWwgdmVyc2lvbj0nMS4wJyBlbmNvZGluZz0nVVRGLT
        gnPz48T3JkZXIgeG1sbnM9J2h0dHA6Ly9wYXltZW50cy5hbWF6b2
        4uY29tL2NoZWNrb3V0LzIwMDgtMTEtMzAvJz48Q2FydD48SXRlbX
        M+PEl0ZW0+PFNLVT5BQkMxMjM8L1NLVT48TWVyY2hhbnRJZD5BMU
        VKTjJKSUcxMEs2NzwvTWVyY2hhbnRJZD48VGl0bGU+UmVkIEZpc2
        g8L1RpdGxlPjxQcmljZT48QW1vdW50PjE5Ljk5PC9BbW91bnQ+PE
        N1cnJlbmN5Q29kZT5VU0Q8L0N1cnJlbmN5
        Q29kZT48L1ByaWNlPj
        xRdWFudGl0eT4xPC9RdWFudGl0eT48RnVs
        ZmlsbG1lbnROZXR3b3
        JrPk1FUkNIQU5UPC9GdWxmaWxsbWVudE5ldHd
        vcms+PC9JdGVtPj
        wvSXRlbXM+PC9DYXJ0PjwvT3JkZXI+;signat
        ure:0nppbXTktfo
        V80Kh41GK7ruUhq8=;aws-access-key-
        id:19G7E7X2QE2V45LET"},
        }).render("cbaButton1");
    </script>
    ...
```

The resulting page will look something like this:

```
<html>
...
<head>
<script language=javascript
   src=https://payments-
   sandbox.amazon.com/cba/js/PaymentWidgets.js>
</script>
...
</head>
<body>
...
<<div id="cbaButton1">
  // Replace "AEIOU1234AEIOU" with your own merchantId value
  <img src="https://payments.amazon.com/gp/cba/button?
cartOwnerId=AEIOU1234AEIOU&color=orange&size=large&background=whi
te"/>
</div>

<script>
  new CBA.Widgets.StandardCheckoutWidget({
    merchantId:'AEIOU1234AEIOU',
    orderInput: {format: "XML",
    value: "type:merchant-signed-order/aws-accesskey/1;
    order:PD94bWwgdmVyc2lvbj0nMS4wJyBlbmNvZGluZz0nVVRGLT
    gnPz48T3JkZXIgeG1sbnM9J2h0dHA6Ly9wYXltZW50cy5hbWF6b2
    4uY29tL2NoZWNrb3V0LzIwMDgtMTEtMzAvJz48Q2FydD48SXRlbX
    M+PEl0ZW0+PFNLVT5BQkMxMjM8L1NLVT48TWVyY2hhbnRJZD5BMU
    VKTjJKSUcxMEs2NzwvTWVyY2hhbnRJZD48VGl0bGU+UmVkIEZpc2
    g8L1RpdGxlPjxQcmljZT48QW1vdW50PjE5Ljk5PC9BbW91bnQ+PE
    N1cnJlbmN5Q29kZT5VU0Q8L0N1cnJlbmN5Q29kZT48L1ByaWNlPj
    xRdWFudGl0eT4xPC9RdWFudGl0eT48RnVsZmlsbG1lbnROZXR3b3
    JrPk1FUkNIQU5UPC9GdWxmaWxsbWVudE5ldHdvcms+PC9JdGVtPj
    wvSXRlbXM+PC9DYXJ0PjwvT3JkZXI+;signature:0nppbXTktfo
    V80Kh41GK7ruUhq8=;aws-access-key-id:19G7E7X2QE2V45LET"},
    }).render("cbaButton1");
</script>
...
</body>
</html>
```

Now place this code on the test web page where you want the Sandbox Checkout button to appear.

Note: Your checkout button and cart code must contain the lines of JavaScript as shown in the example above. For a complete list of button styles, please see Button Styles.

Do not mix Sandbox and Production buttons on the same page.

Test Your Sandbox Button by Placing a Test Order

Now that you have created a Sandbox Checkout button and placed it on a test web page, you're ready to test it.

Note: Remember, you must use your Buyer (Test) Account to test buying items, even if you buy items in the Sandbox.

Note: Orders you place using Sandbox buttons are never charged. This means your Buyer (Test) Account will not receive a confirmation e-mail, and your Seller Account will not receive any notification on the sale. You can, however, track Sandbox orders in Seller Central by searching for "Pending Orders." For more information about how to do so, please see Managing Orders.

When you test a Sandbox button, you test for the following exit criteria:

- Clicking the Sandbox Checkout button starts the checkout process.
- When you come to a page that displays a Place order button, clicking the button delivers a message that the order was placed.
- Using your Seller Account in Seller Central, you can find the pending order on the Manage Orders page.

Follow these steps to place test orders in the Sandbox.

1. Open your web browser and go the URL for your test web page.
2. Click the **Sandbox Checkout** button you just created.
3. At the confirmation page, review the information to be sure it's accurate.
   Make sure the shipping and promotional discount information is correct.
4. Click the **Place this order** button to complete the Sandbox purchase.

Remember, although you will see a confirmation message, your Buyer (Test) Account will not get a confirmation e-mail, because Sandbox transactions do not become orders.

Production Setup

Create a Production Cart Using XML

Building a Production cart lets you create real orders; that is, orders that are processed, charged, and fulfilled. A simple XML-based cart contains following elements:

Element Name	Description	Required	XPath
Order	Describes order schema	Yes	Root
Cart	Container for entries in cart	Yes	\Order\Cart
Items	Container for item entries	Yes	\Order\Cart\Items
Item	Container for elements of the order 1-999 item entries per cart	Yes	\Order\Cart\Items\Item
SKU	Stock keeping ID for the item 1 to 250 characters.	Yes. If using Fulfillment by Amazon, the SKUs must match	\Order\Cart\Items\Item\SKU
MerchantId	14-character alphanumeric string Get from **Settings** > **Account Info** > Integration Settings Settings	Yes	\Order\Cart\Items\Item\MerchantId
Title	Name of Item 1 - 80 alphanumeric characters only	Yes	\Order\Cart\Items\Item\Title
Price	Container for Amount, CurrencyCode	Yes	\Order\Cart\Items\Item\Price
Amount	Cost of item	Yes	\Order\Cart\Items\Item\Price\Amount
CurrencyCode	Type of currency	Yes	\Order\Cart\Items\Item\Price

	Only "INR" accepted		\CurrencyCode

The following is an example of a simple Production cart using XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<Order xmlns="http://payments.amazon.com/checkout/2009-05-15/">
  <Cart>
    <Items>
      <Item>
        <SKU>ABC123</SKU>
        <MerchantId>AEIOU1234AEIOU</MerchantId>
        <Title>Red Fish</Title>
        <Price>
          <Amount>19.99</Amount>
          <CurrencyCode>INR</CurrencyCode>
        </Price>
        <Quantity>1</Quantity>
      </Item>
    </Items>
  </Cart>
</Order>
```

Note: Do not include / (forward slash) before Item `Title` value.

Sign Your Cart

To generate a signature, you need a set of signing-key pairs. By default, your Pay with Amazon account supplies you with a set of MWS Access Keys from Seller Central, the Access Key ID (the public key) and the Secret Access Key (the private key). See Getting Your Key for steps to take to get your Access Keys.

Once you have your signing-key pairs, you compute an RFC2104-compliant HMAC signature of the concatenated cart string. No canonicalization is required.

The following table lists the types of signature and their HTML representation in the cart.

Name	Description	Type of Cart
type	One of the following: • merchant-signed-order/aws- accesskey/1 (for Access Key signatures) • unsigned order	All types

order	base64-encoded order	All types
signature	Signature calculated according to type	merchant-signed-order/aws-accesskey/1
aws-access-key-id	Access Key ID retrieved in Seller Central	merchant-signed-order/aws-accesskey/1

Follow these steps to sign your XML-based carts:

1. Get your MWS Access Keys.
2. Get the XML code used for your cart.
   Note that the XML code you sign must be identical in content and layout (well-formatted or not) you use later in the step "Base64-Encode Your Cart."
3. Using the programming language of your choice, sign the XML code with your key.
4. Using the programming language of your choice, base64-encode the resulting signature.

The resulting encoded signature will look something like this: `0nppbXTktfoV80Kh41GK7ruUhq8=`

Base64-Encode Your Cart

After you get the XML code for your cart, you base64-encode the XML code using the programming language of your choice.

If you permit only signed carts, you also must base64-encode the signature before you add it to your cart form. Note that you add the signature to the form after you have signed and base64-encoded the cart.

Here is an example of a signed cart, signed using the Access Key pair (line breaks added for clarity):

```
<html>
...
<head>
<script language=javascript
    src=https://static-na.payments-
    amazon.com/cba/js/us/PaymentWidgets.js >
</script>
...
</head>
<body>
...
<div id="cbaButton1">
  // Replace "AEIOU1234AEIOU" with your own merchantId value
  <img src="https://payments.amazon.com/gp/cba/button?
cartOwnerId=AEIOU1234AEIOU&color=orange&size=large&background=whi
te"/>
</div>

<script>
```

```
      new CBA.Widgets.StandardCheckoutWidget({
        merchantId:'AEIOU1234AEIOU',
        orderInput: {format: "XML",
        value: "type:merchant-signed-order/aws-accesskey/1;
    order:PD94bWwgdmVyc2lvbj0nMS4wJyBlbmNvZGluZz0nVVRGLT
    gnPz48T3JkZXIgeG1sbnM9J2h0dHA6Ly9wYXltZW50cy5hbWF6b2
    4uY29tL2NoZWNrb3V0LzIwMDgtMTEtMzAvJz48Q2FydD48SXRlbX
    M+PEl0ZW0+PFNLVT5BQkMxMjM8L1NLVT48TWVyY2hhbnRJZD5BMU
    VKTjJKSUcxMEs2NzwvTWVyY2hhbnRJZD48VGl0bGU+UmVkIEZpc2
    g8L1RpdGxlPjxQcmljZT48QW1vdW50PjE5Ljk5PC9BbW91bnQ+PE
    N1cnJlbmN5Q29kZT5VU0Q8L0N1cnJlbmN5Q29kZT48L1ByaWNlPj
    xRdWFudGl0eT4xPC9RdWFudGl0eT48RnVsZmlsbG1lbnROZXR3b3
    JrPk1FUkNIQU5UPC9GdWxmaWxsbWVudE5ldHdvcms+PC9JdGVtPj
    wvSXRlbXM+PC9DYXJ0PjwvT3JkZXI+;signature:0nppbXTktfo
    V80Kh41GK7ruUhq8=;aws-access-key-id:19G7E7X2QE2V45LET"},
    }).render("cbaButton1");
</script>
...
</body>
</html>
```

And here is an example of an unsigned cart:

```
<html>
...
<head>
<script language=javascript
    src=https://static-na.payments-
    amazon.com/cba/js/us/PaymentWidgets.js >
</script>
...
</head>
<body>
...
<div id="cbaButton1">
  // Replace "AEIOU1234AEIOU" with your own merchantId value
  <img src="https://payments.amazon.com/gp/cba/button?
cartOwnerId=AEIOU1234AEIOU&color=orange&size=large&background=white"/
>
</div>

<script>
  new CBA.Widgets.StandardCheckoutWidget({
    merchantId:'AEIOU1234AEIOU',
    orderInput: {format: "XML",
    value:"type:unsigned-order;order:PD94bWwgdmV
    yc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4KPE9
    yZGVyIHhtbG5zPSJodHRwOi8vcGF5bWVudHMuYW1hem9
    uLmNvbS9jaGVja291dC8yMDA4LTA2LTE1LyI+CiAgPEN
    saWVudFJlcXVlc3RJZD5PcmRlci0yNDwvQ2xpZW50UmV
    xdWVzdElkPgogIDxFeHBpcmF0aW9uRGF0ZT4yMDA4LTE
    yLTEyPC9FeHBpcmF0aW9uRGF0ZT4KICA8QFydD4KICAg
    IDxJdGVtcz4KICAgICAgPEl0ZW0+CiAgICAgICAgPFNL
    VT5UU0hJUlQwMDAxPC9TS1U+CiAgICAgICAgPE1lcmNo
    YW50SWQ+PC9NZXJjaGFudElkPgogICAgICAgIDxUaXRs
    ZT5BbWF6b24gUGF5bWVudHMgVC1TaGlydDwvVGl0bGU+
    CiAgICAgICAgPFByaWNlPgogICAgICAgICAgPEFtb3Vu
```

```
      dD45Ljk1PC9BbW91bnQ+CiAgICAgICAgICA8Q3VycmVu
      Y3lDb2RlPlVTRDwvQ3VycmVuY3lDb2RlPgogICAgICAg
      IDwvUHJpY2U+CiAgICAgICAgPFF1YW50aXR5PjE8L1F1
      YW50aXR5PgogICAgICA8L0l0ZW0+CiAgICA8L0l0ZW1z
      PgogIDwvQ2FydD4KPC9PcmRlcj4K"},
  </script>

  ...
  </body>
  </html>
```

Follow these steps to base64-encode your cart.

1. Get your Keys.
2. Get XML code used for your cart.
   Note that the XML code you sign **must be identical** in content and layout (well-formatted or not) to the code you used earlier in the Step "Sign Your Cart."
3. Using the programming language of your choice, sign the XML code with your key.
4. Using the programming language of your choice, base64-encode the resulting signature. The resulting encoded signature will look something like this:

```
0nppbXTktfoV80Kh41GK7ruUhq8=
```

Add the Signature to the Cart

You can add this signature to your cart.

Signature type	Value entry	Keyword entry
MWS Access Keys	value="type:merchant-signed-order/aws-accesskey/1	aws-access-key-id

Follow these steps to add the signature to the HTML form.

Add to the orderinput tag the following values:

```
value="type:merchant-signed-order/aws-accesskey/1; ...
signature:[Signature value];
aws-access-key-id:[Access Key ID value]
```

Your form should look like the following (line breaks added for clarity):

```
<script>
  new CBA.Widgets.StandardCheckoutWidget({
    merchantId:'AEIOU1234AEIOU',
    orderInput: {format: "XML",
    value: "type:merchant-signed-order/aws-accesskey/1;
    order:PD94bWwgdmVyc2lvbj0nMS4wJyBlbmNvZGluZz0nVVRGL
    TgnPz48T3JkZXIgeG1sbnM9J2h0dHA6Ly9wYXltZW50cy5hbWF6
    b24uY29tL2NoZWNrb3V0LzIwMDgtMTEtMzAvJz48Q2FydD48SXR
    lbXM+PEl0ZW0+PFNLVT5BQkMxMjM8L1NLVT48TWVyY2hhbnRJZD
    5BMUVKTjJKSUcxMEs2NzwvTWVyY2hhbnRJZD48VGl0bGU+UmVkI
    EZpc2g8L1RpdGxlPjxQcmljZT48QW1vdW50PjE5Ljk5PC9BbW91
    bnQ+PEN1cnJlbmN5Q29kZT5VU0Q8L0N1cnJlbmN5Q29kZT48L1B
    yaWNlPjxRdWFudGl0eT4xPC9RdWFudGl0eT48RnVsZmlsbG1lbn
    ROZXR3b3JrPk1FUkNIQU5UPC9GdWxmaWxsbWVudE5ldHdvcms+P
    C9JdGVtPjwvSXRlbXM+PC9DYXJ0PjwvT3JkZXI+;signature:
    0nppbXTktfoV80Kh41GK7ruUhq8=
    ;aws-access-key-id:19G7E7X2QE2V45LET"},
    buttonSettings:
    {size:'large',color:'orange',background:'white'}
    }).render("cbaButton1");
</script>
```

Place the Production Checkout Button Code on Your Test Webpage

When you have built the XML code for the cart, signed it (if you accept only signed carts), and base64-encoded the cart value and the optional signature, you add the resulting cart value and signature to an HTML form that displays the checkout button.

For example, given the example earlier, the resulting code will look something like this (note that line breaks are for ease of reading):

```
...
<script>
  new CBA.Widgets.StandardCheckoutWidget({
    merchantId:'AEIOU1234AEIOU',
    orderInput: {format: "XML",
    value: "type:merchant-signed-order/aws-accesskey/1;
    order:PD94bWwgdmVyc2lvbj0nMS4wJyBlbmNvZGluZz0nVVRGLT
    gnPz48T3JkZXIgeG1sbnM9J2h0dHA6Ly9wYXltZW50cy5hbWF6b2
    4uY29tL2NoZWNrb3V0LzIwMDgtMTEtMzAvJz48Q2FydD48SXRlbX
    M+PEl0ZW0+PFNLVT5BQkMxMjM8L1NLVT48TWVyY2hhbnRJZD5BMU
    VKTjJKSUcxMEs2NzwvTWVyY2hhbnRJZD48VGl0bGU+UmVkIEZpc2
    g8L1RpdGxlPjxQcmljZT48QW1vdW50PjE5Ljk5PC9BbW91bnQ+PE
    N1cnJlbmN5Q29kZT5VU0Q8L0N1cnJlbmN5Q29kZT48L1ByaWNlPj
    xRdWFudGl0eT4xPC9RdWFudGl0eT48RnVsZmlsbG1lbnROZXR3b3
    JrPk1FUkNIQU5UPC9GdWxmaWxsbWVudE5ldHdvcms+PC9JdGVtPj
    wvSXRlbXM+PC9DYXJ0PjwvT3JkZXI+;signature:0nppbXTktfo
    V80Kh41GK7ruUhq8=;aws-access-key-id:19G7E7X2QE2V45LET"},
    }).render("cbaButton1");
</script>
...
```

The resulting page will look something like this:

```
<html>
...
<head>
<script language=javascript
    src=https://payments.amazon.com/cba/jw/PaymentWidgets.js>
</script>
...
</head>
<body>
...
<div id="cbaButton1">
  // Replace "AEIOU1234AEIOU" with your own merchantId value
  <img src="https://payments.amazon.com/gp/cba/button?
cartOwnerId=AEIOU1234AEIOU&color=orange&size=large&background=white"/
>
</div>

<script>
  new CBA.Widgets.StandardCheckoutWidget({
    merchantId:'AEIOU1234AEIOU',
    orderInput: {format: "XML",
    value: "type:merchant-signed-order/aws-accesskey/1;
```

```
        order:PD94bWwgdmVyc2lvbj0nMS4wJyBlbmNvZGluZz0nVVRGLT
        gnPz48T3JkZXIgeG1sbnM9J2h0dHA6Ly9wYXltZW50cy5hbWF6b2
        4uY29tL2NoZWNrb3V0LzIwMDgtMTEtMzAvJz48Q2FydD48SXRlbX
        M+PEl0ZW0+PFNLVT5BQkMxMjM8L1NLVT48TWVyY2hhbnRJZD5BMU
        VKTjJKSUcxMEs2NzwvTWVyY2hhbnRJZD48VGl0bGU+UmVkIEZpc2
        g8L1RpdGxlPjxQcmljZT48QW1vdW50PjE5Ljk5PC9BbW91bnQ+PE
        N1cnJlbmN5Q29kZT5VU0Q8L0N1cnJlbmN5Q29kZT48L1ByaWNlPj
        xRdWFudGl0eT4xPC9RdWFudGl0eT48RnVsZmlsbG1lbnROZXR3b3
        JrPk1FUkNIQU5UPC9GdWxmaWxsbWVudE5ldHdvcms+PC9JdGVtPj
        wvSXRlbXM+PC9DYXJ0PjwvT3JkZXI+;signature:0nppbXTktfo
        V80Kh41GK7ruUhq8=;aws-access-key-id:19G7E7X2QE2V45LET"},
        }).render("cbaButton1");
    </script>
    ...
    </body>
```

Now place this code on the test webpage where you want the Production Checkout button to appear.

Note: Your checkout button and cart code must contain the lines of JavaScript as shown in the example above. For a complete list of button styles, please see Button Styles.

Do not mix Sandbox and Production buttons on the same page.

Test Your Production Button by Placing a Test Order

Now that you have created a Production Checkout button and placed it on a test web page, you're ready to test it.

Note: Remember, you must use your Buyer (Test) Account to test buying items.

Note: Orders you place using Production buttons are always charged. This means your Buyer (Test) Account **will** receive a confirmation e-mail, and your Seller Account **will** receive notification on the sale. You track Production orders in Seller Central, with Orders Reports, or IOPN. See Managing Orders for more information.

Important: Remember to cancel your test Production order after you place it; otherwise, your Buyer (Test) Account credit card will be charged. Also note that you are required as a seller to fulfill or cancel all orders within 30 days, even if you placed the order using your Buyer (Test) Account.

When you test a Production button, you test for the following exit criteria:

- Clicking the Production Checkout button starts the checkout process.
- After you place the order, your Buyer (Test) Account receives a confirmation e-mail from Pay with Amazon.
- Using your Buyer (Test) Account on the Pay with Amazon site (https://paywithamazon.in), you can find the order on the Your Account page.
- Using your Seller Account in Seller Central, you can find the order on the Manage Orders page.

Follow these steps to place test orders in Production:

1. Open your web browser and go the URL for your test web page.
2. Click the Production Checkout button you just created.
3. At the confirmation page, review the information to be sure it's accurate.
   Make sure the shipping, tax, and promotional discount information is correct.
4. Click the **Place this order** button to complete the Production purchase.
   You will see a confirmation message, and your Buyer (Test) Account will get a confirmation e-mail.

Cancel Your Test Production Order

After you place a test order in Production, you should cancel it within 15 minutes.

Note: You do not need to cancel a test order if you intend to use it to test your fulfillment systems.
All orders that are not canceled by the buyer must be either fulfilled or canceled by the merchant within 30 days. Orders that are not fulfilled or canceled within 30 days by the merchant will be canceled by Pay with Amazon. Your cancellation rate affects merchant performance; a high rate of merchant-caused cancellations can lead to review by Pay with Amazon.

Follow these steps to cancel an order using your Buyer (Test) Account.

1. Go to the Pay with Amazon website (https://payments.amazon.com/).
2. Click the **Your Account** tab.
3. When prompted, sign in using your Buyer (Test) Account e-mail and password.
   You might not be prompted to sign in if you have already signed in using your Buyer (Test) Account on Pay with Amazon, Amazon.in, or other Amazon-related sites.

   Tip: Check the sign-in name in the upper-right corner. If it's not the name you expect, click "Sign Out" and then click the Your Account tab again.
4. Find the test order you placed.
5. Click the **Details** link next to the test order.
   The order details page appears.
6. Click the **Need to cancel this order** button.
7. Click **Cancel** order.
   The order is canceled.

Place the Production Checkout Button Code on Your Live Webpage

After you have built and tested your Production Checkout button, you place the button (the HTML form with the base64-encoded cart and optional signature) on your live web page so that your customers can begin using the button to place real orders.

For example, your complete HTML page might look something like this (line breaks added for clarity):

```
<html>
...
<head>
<script language=javascript
    src=https://payments.amazon.com/cba/jw/PaymentWidgets.js>
</script>
...
</head>
<body>
...
<div id="cbaButton1">
  // Replace "AEIOU1234AEIOU" with your own merchantId value
  <img src="https://payments.amazon.com/gp/cba/button?
cartOwnerId=AEIOU1234AEIOU&color=orange&size=large&background=wh
ite"/>
</div>

<script>
  new CBA.Widgets.StandardCheckoutWidget({
    merchantId:'AEIOU1234AEIOU',
    orderInput: {format: "XML",
    value: "type:merchant-signed-order/aws-
accesskey/1;order:PD94bWwgdmVyc2lvbj0nMS4wJyBlbmNvZGluZz0nVVRGLT
gnPz48T3JkZXIgeG1sbnM9J2h0dHA6Ly9wYXltZW50cy5hbWF6b24uY29tL2NoZW
Nrb3V0LzIwMDgtMTEtMzAvM+PEl0ZW0+PFNLVT5BQkMxMjM8L1NLVT48TWVyY2hh
bnRJZD5BMUVKTjJKSUcxMEs2NzwvTWVyY2hhbnRJZD48VG+UmVkIEZpc2g8L1Rpd
GxlPjxQcmljZT48QW1vdW50PjE5Ljk5PC9BbW91bnQ+PEN1cnJlbmN5Q29kZT5VU
0Q8L0N1cnJlbmN5Q29kZT48L1ByaWNlPjxRdWFudGl0eT4xPC9RdWFudGl0eT48R
nVsZmlsbG1lbnROZXR3b3JrPk1FUkNIQU5UPC9GdWxmaWxsbWVudE5l+PC9JdGVt
PjwvSXRlbXM+PC9DYXJ0PjwvT3JkZXI+;signature:0nppbXTktfoV80Kh41GK7
ruUhq8=;aws-access-key-
id:19G7E7X2QE2V45LET"},}).render("cbaButton1");
</script>
...
</body>
```

Now place this code on the Production web page where you want the Checkout button to appear.

Note: Your checkout button and cart code must contain the lines of JavaScript as shown in the example above.
For a complete list of button styles, please see Button Styles.
Do not mix Sandbox and Production buttons on the same page.

Set Up for Your Mobile-Optimized Website

Enabling Pay with Amazon on Your Mobile-Optimized Website

By default, Pay with Amazon will work on your mobile-optimized website - you do not need to do additional work. To enable Pay with Amazon on your mobile-optimized website, simply follow the same steps as listed in Steps to Integrating Pay with Amazon with Your Website .

When the buyer clicks the Checkout with Amazon button on your mobile-optimized website, Pay with Amazon detects that the request is coming from a mobile device and will redirect the customer to the mobile-optimized Integration Settings hosted by Pay with Amazon. This Integration Settings is optimized for the latest browser-enabled smartphones which support JavaScript, including iPhone, Android, and Webkit-based smartphones.

The Customer Experience on a Mobile-Optimized Website

Here is an example of the customer experience when using a smartphone to complete a purchase on a mobile- optimized website.

1. Customer clicks the Checkout with Amazon button on your website.
   Behind the scenes, we determine the checkout originates from a mobile device and redirect the customer to the mobile-optimized Pay with Amazon login page.
2. After signing in, the customer sees the mobile-optimized order confirmation page on the Pay with Amazon website.
   The customer selects the shipping method, shipping address, payment information, and then places the order.
   Then, the customer is redirected to the order confirmation page on the merchant's website.

Chapter 3 - Handling Settlements

The order will get to the Ready-To-Ship state when the payment amount is reserved successfully on buyer's payment method. You must wait for the order to get to the Ready-To-Ship state before you fulfill the order. You can listen to the order notifications to know immediately when the order state changes. Please see the Instant Order Processing Notifications Guide for details about using instant order processing notifications.

Once the order is ready to ship, you must take action on them:

You must fulfill them (ship them and mark them as shipped), or You must cancel them (for test orders, it's easier to cancel them with your buyer (test) account).

The customer is charged only when you fulfill the order. You must take action on all orders within 30 days; otherwise, Pay with Amazon will cancel the order.

Please see the Managing Orders Guide to learn more about fulfilling, cancelling, refunding, and other order management functions.

Chapter 4 - Post-Setup Exit Criteria

Topics:

- *Exit Criteria*
- *Ten Types of Test Transactions*
- *Order Options Used When Testing Transactions*

After you've set up and tested your integration, you should follow the steps listed in this chapter to be sure you are ready to release your website's integration with Pay with Amazon to the public.

Exit Criteria

Here are the exit criteria for successfully building Checkout buttons for your website. Use this checklist to be sure that your checkout buttons work correctly before you release your website to the public.

1. The button works in the Sandbox and in Production (that is, the order pipeline completes successfully).
2. The Production order appears in Pay with Amazon site using buyer account login.
3. The Production order appears in Seller Central using merchant account login (on the Manage Orders page).
4. You cancel all test orders with your Buyer (Test) Account on the Pay with Amazon site.

Ten Types of Test Transactions

Use these additional transaction types to fully test your Pay with Amazon implementation:

1. Simple 1-item
2. Item with variations
3. Item for less than ₹1.00
4. Item for greater than ₹50
5. Item with multiple quantities
6. Item shipping to a domestic address using standard shipping
7. Item shipping to a domestic address using expedited shipping
8. Fulfill an order in Production

Order Options Used When Testing Transactions

When you build and test your full implementation of Pay with Amazon on your site, you should consider building transactions with the following types of conditions:

1. Item with per-item and per-shipment charges
2. Item with banded shipping

3. Item with One-day shipping
4. Item with Two-day shipping
5. Purchase made using Standard Checkout

Chapter 5 - Cart and Button Customizations

To complete integrating Pay with Amazon with your cart, you'll display Payment Button images that your buyers will click to successfully place orders.

You can choose from several buttons styles based upon the size of the button or the color of the button itself or its background.

For example, suppose you want to use the following button:

To display this button, set the following img src url in the div to render the widget:

```
// Replace "AEIOU1234AEIOU" with your own merchantId value
<img src="https://payments.amazon.com/gp/cba/button
     ?cartOwnerId=AEIOU1234AEIOU
     &color=orange
     &size=x-large
     &background=white"/>
```

To Display this button with different Styles use buttonSettings in a script section under function new CBA.Widgets.StandardCheckoutWidget as follows :

buttonSettings: { size: 'large', color: 'tan', background: 'dark' }

Herea are buttonSettings parameters and their allowed values

buttonSettings Parameters	**Allowed Values**
Size	▪ Medium(126 x 35) ▪ Large(151 x 38) ▪ X-large(173 x 38)
Color	▪ Orange ▪ Tan
Background	▪ Light ▪ Dark ▪ White

Appendix A - Code Samples

Topics:

- *Signing Carts*
- *Instant Order Processing Notification*

Use these code samples to help understand and build your own code.

Note that these code samples use a default Merchant ID of "AEIOU1234AEIOU" -- when you see this Merchant ID in the code samples, replace it with your own Merchant ID in your Sandbox and Production code.

Signing Carts

File name	Version	Updated
Java-based code for signing your cart	1.0.2	2009-03-10 16:00
PHP-based code for signing your cart	1.0.2	2008-11-05 16:00
C#-based code for signing your cart	1.0.0	2009-01-20 16:30

Instant Order Processing Notification

File name	Version	Updated
Instant Order Processing Notification Sample Code - Java	1.1.0	2014-12-09
Instant Order Processing Notification Sample Code - PHP	1.0.0	2009-02-03

Appendix B - Button Styles

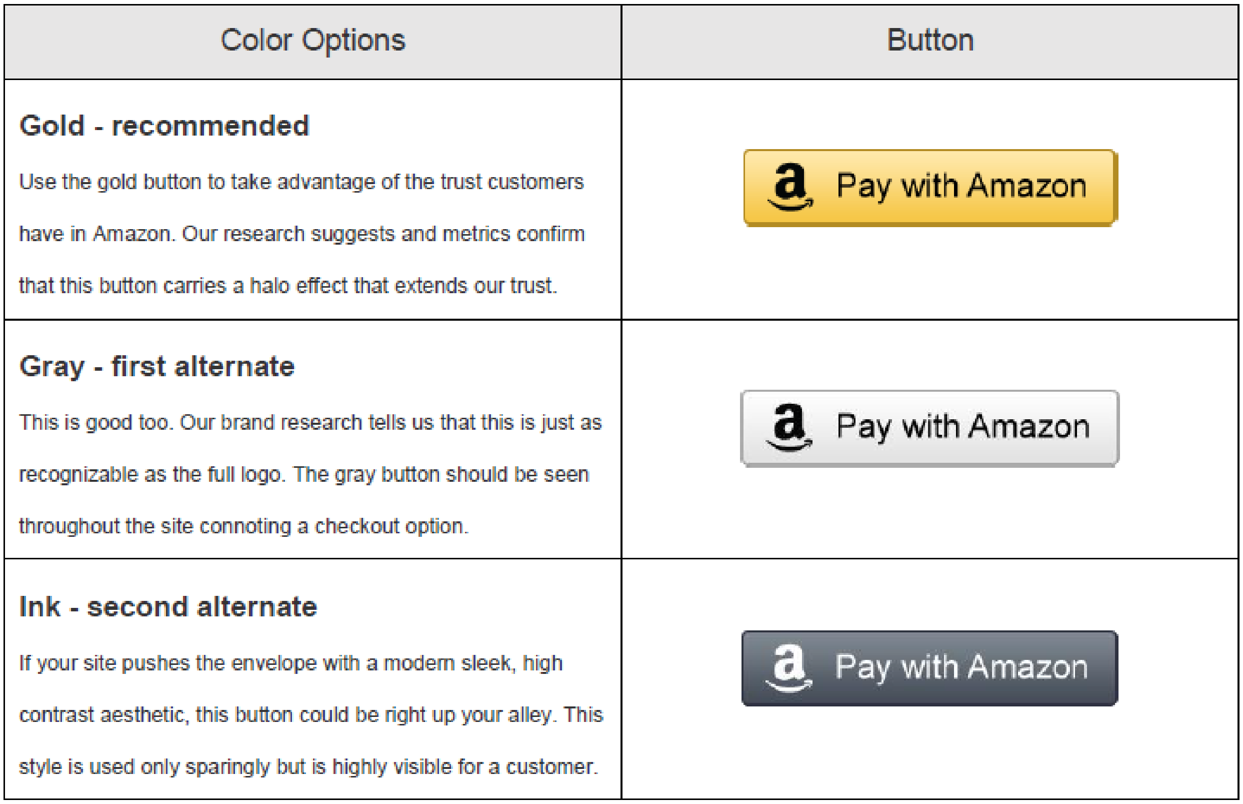
Topics:

- *White Background*
- *Light Background*
- *Dark Background*

To complete integrating Pay with Amazon with your cart, you'll display Payment Button images that your buyers will click to successfully place orders. Choose a Payment Button image you want to use, and then copy and paste the URL to your website.

Here are the payment button images you can place on your website.

Note: Please refer to the Pay with Amazon Image Library for acceptance marks and to see all the button variations. Use is subject to our trademark and usage guidelines.



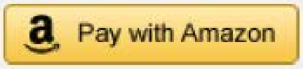
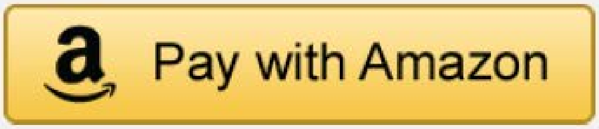
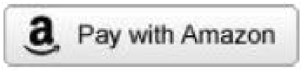
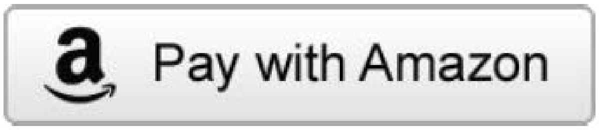
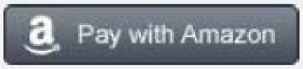
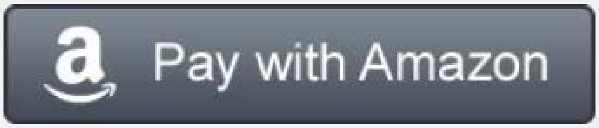
Color Options	Button
Gold - recommended Use the gold button to take advantage of the trust customers have in Amazon. Our research suggests and metrics confirm that this button carries a halo effect that extends our trust.	a Pay with Amazon
Gray - first alternate This is good too. Our brand research tells us that this is just as recognizable as the full logo. The gray button should be seen throughout the site connoting a checkout option.	a Pay with Amazon
Ink - second alternate If your site pushes the envelope with a modern sleek, high contrast aesthetic, this button could be right up your alley. This style is used only sparingly but is highly visible for a customer.	a Pay with Amazon

These images are pre-approved for use on your website, provided you do not make any modifications. If you have a use case in which none of the provided images fit, please contact us to work with you on a solution.

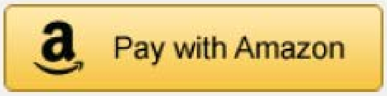
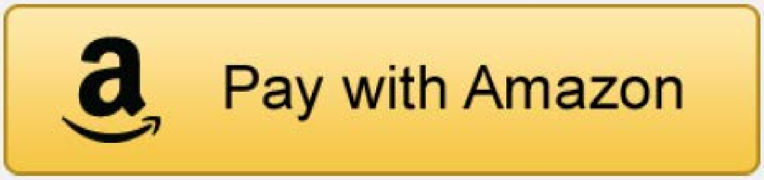
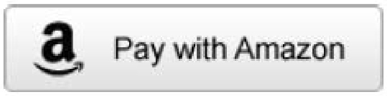
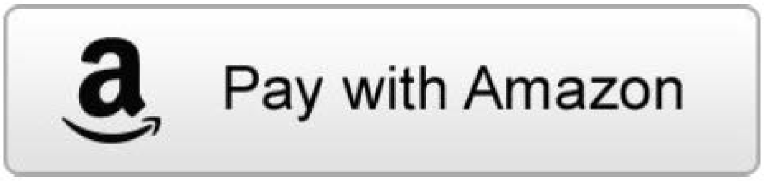
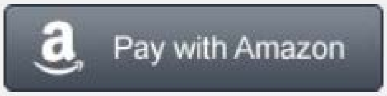
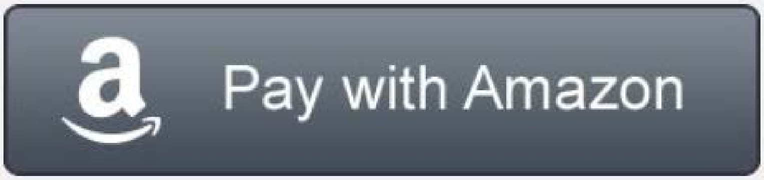
To render these buttons, please call the graphic directly from our servers rather than hosting your own copy. This will ensure you always have the latest, most secure version of the button graphic. Check your integration guide for further details on how to do this.

For buttons related to your account-based use cases, please refer to the Button Guidelines.

Desktop Buttons

@1x: 72 x 30 px	@2x: 144 x 60 px
	
	
	

Touch Buttons

@1x: 200 x 45 px	@2x: 400 x 90 px
	
	
	

Appendix C - The XML Schema and Definitions

Topics:

- *Definition Tables*

Here is the complete listing of cart elements and their definitions for orders processed by Pay with Amazon.

Definition Tables

A

Amount (price)

Definition	The monetary charge for an Item.
Required	Yes
Type	Price
Value/Range	0.01--99999.99
XPath	/Order/Cart/Items/Item/Price/Amount

Amount (weight)

Definition	The Weight value for your Item.
Required	Yes, if Weight is specified.
Type	Positive double
Value/Range	0.01--99999.99
XPath	/Order/Cart/Items/Item/Weight/Amount

Amount (discount percent)

Definition	The discount amount expressed as a percentage of the Price.
Required	Yes, if DiscountRate is specified.
Type	Nonnegative double. A 10 percent discount is 0.10; a 0.10 discount is 0.001
Value/Range	0.0--1.0
XPath	/Order/Promotions/Promotion/DiscountRate/Amount

Amount (discount amount)

Definition	The discount expressed as units of currency.
Required	Yes, if FixedAmountDiscount is specified.
Type	Nonnegative double
Value/Range	0.00--99999.99
XPath	/Order/Promotions/Promotion/FixedAmountDiscount/Amount

Amount (shipping)

Definition	The monetary charge for a shipment.
Required	Yes, if ShipmentBased is specified.
Type	Price
Value/Range	0.01--99999.99
XPath	/Order/ShippingMethods/ShippingMethod/Rate/ShipmentBased/Amount

Amount (weight)

Definition	The monetary calculation for an order. Weight is calculated based upon Quantity of the Item times Amount.
Required	Yes, if WeightBased is specified.
Type	Boolean
Value/Range	Price
XPath	/Order/ShippingMethods/ShippingMethod/Rate/WeightBased/Amount

C

Cart

Definition	The essential container in your Order. A simple Cart has one Items element containing your
Required	Yes
XPath	/Order/Cart

Category (item)

Definition	The selling group your Item belongs to.
Required	No
Type	String from enumerated list
Value/Range	Antiques | Apparel, Shoes & Accessories | Automotive | Beauty | Books | Business | Cameras & Photography | Collectibles | Computer Hardware & Accessories | Electronics | Food, Drink & Nutrition | Gifts & Flowers | Healthcare | Hobbies, Toys & Games | Home & Garden | Jewelry & Watches | Media & Entertainment | Pets & Animals | Software | Sports & Recreation | Travel
XPath	Order/Cart/Items/Item/Category

Condition

Definition	The state of the Item.
Required	Yes, if you are using Fulfillment by Amazon for this item and you sell the same Item in multiple conditions. No, if you fulfill the Item yourself, or if your Fulfillment by Amazon Item has only one Condition.
Type	String from enumerated list
Value/Range	Any | Club | Collectible | New | Refurbished | Used
XPath	/Order/Cart/Items/Item/Condition

CountryCode

Definition	The 2-letter ISO abbreviation for the country.
Required	No
Type	String 2
Value/Range	ISO countries
XPath	/Order/ShippingMethods/ShippingMethod/IncludedRegions/IncludedRegion/WorldRegion/ CountryCode /Order/ShippingMethods/ShippingMethod/ExcludedRegions/ExcludedRegion/WorldRegion/ CountryCode

CurrencyCode (item)

Definition	The type of currency for the Item charge. Currently, only USD is accepted.
Required	Yes
Type	String from enumerated list
Value/Range	USD
XPath	The type of currency for the Item charge. Currently, only USD is accepted.

CurrencyCode (discount rate)

Definition	The type of currency for the DiscountRate charge. Currently, only USD is accepted.
Required	Yes
Type	String from enumerated list
Value/Range	USD
XPath	/Order/Promotions/Promotion/DiscountRate/CurrencyCode

CurrencyCode (fixed amount)

Definition	The type of currency for the FixedAmountDiscount charge. Currently, only USD is
Required	Yes, if FixedAmountDiscount is specified.
Type	String from enumerated list
Value/Range	USD
XPath	/Order/Promotions/Promotion/FixedAmountDiscount/CurrencyCode

CurrencyCode (shipment)

Definition	The type of currency for the ShipmentBased charge. Currently, only USD is accepted.
Required	Yes, if ShipmentBased is specified.
Type	String from enumerated list
Value/Range	YSD
XPath	/Order/ShippingMethods/ShippingMethod/Rate/ShipmentBased/CurrencyCode

CurrencyCode (weight)

Definition	The type of currency for the WeightBased charge. Currently, only USD is accepted.
Required	Yes, if WeightBased is specified.
Type	String from enumerated list
Value/Range	USD
XPath	/Order/ShippingMethods/ShippingMethod/Rate/WeightBased/CurrencyCode

D

Description (item)

Definition	The details about your Item.
Required	No
Type	String 255
XPath	/Order/Cart/Items/Item/Description

Description (shipment)

Definition	The description (or abbreviation) for the shipment method.
Required	No
Type	String 40
XPath	/Order/ShippingMethods/ShippingMethod/Description

DiscountRate

Definition	The container for the discount benefit.
Required	Yes, if PromotionBenefit is specified
XPath	/Order/Promotions/Promotion/DiscountRate

E

ExcludedRegion

Definition	A container for the list of the regions you will not ship to.
Required	No
XPath	/Order/ShippingMethods/ShippingMethod/ExcludedRegions/ExcludedRegion

ExcludedRegions

Definition	A container for the regions you will not ship to.
Required	No
XPath	/Order/ShippingMethods/ShippingMethod/ExcludedRegions

ExpirationDate

Definition	The date the cart expires. After the ExpirationDate has passed, the cart content will not be processed.
Required	No
Type	ISO 8601 date
XPath	/Order/ExpirationDate

F

FixedAmountDiscount

Definition	A container for the fixed amount discount from the Price.
Required	No
XPath	/Order/Promotions/Promotion

FulfillmentNetwork

Definition	The method you use for fulfillment, yourself, or Amazon.
Required	No, if you have not signed up for Fulfillment by Amazon or if you ship the order yourself. Yes, if you sign up for Fulfillment by Amazon and the item is fulfilled by Amazon. Note that if you set FulfillmentNetwork to AMAZON_NA, you **must** also include the SKU
Type	String 10
Value/Range	MERCHANT | AMAZON_NA
XPath	/Order/Cart/Items/Item/FulfillmentMethod

I

IncludedRegion

Definition	A container for the list of the regions you will ship to.
Required	Yes, if ShippingMethods are specified. The IncludedRegion must include at least one of the following: PreDefinedRegion, USStateRegion, USZipRegion,
XPath	/Order/ShippingMethods/ShippingMethod/IncludedRegions/ IncludedRegion

IncludedRegions

Definition	A container for the regions you will ship to.
Required	Yes, if ShippingMethods are specified.
XPath	/Order/ShippingMethods/ShippingMethod/IncludedRegions

IntegratorId

Definition	The unique identifier for the vendor who is assisting you integrating your website with Pay with Amazon.
Required	No
Type	String 13
XPath	/Order/IntegratorId

IntegratorName

Definition	The name of the vendor who is assisting you integrating your website with Pay with Amazon.
Required	No
Type	String 100
XPath	/Order/IntegratorName

IsShippingTaxed

Definition	Specifies whether you charge tax on shipping.
Required	No
Type	Boolean
XPath	/Order/DefaultTaxTable/TaxRules/TaxRule/IsShippingTaxed /Order/TaxTables/TaxTable/TaxRules/TaxRule/IsShippingTaxed

Item

Definition	The container for the information about the product you’re selling.
Required	Yes
XPath	/Order/Cart/Items/Item

Items

Definition	The container for every Item.
Required	Yes
XPath	/Order/Cart/Items

O

Order

Definition	The main container used for your Cart. All other elements are children of the Order element. The simplest Order contains only a Cart, with one Items element containing your MerchantId along with the one Item that has a Price/Amount, a Price/CurrencyCode, and a Quantity. The Order element must include the version information that matches the XSD used to create the XML. For example, if you are using the XSD dated 20-Nov-08, then your namespace must include the following
Required	Yes
XPath	/Order

P

PostalRegion

Definition	The 2-letter ISO abbreviation for the country.
Required	No
Type	String 2
XPath	/Order/DefaultTaxTable/TaxRules/TaxRule/Regions/WorldRegion/PostalRegion /Order/TaxTables/TaxTable/TaxRules/TaxRule/Regions/WorldRegion/PostalRegion /Order/ShippingMethods/ShippingMethod/IncludedRegions/IncludedRegion/WorldRegion/ PostalRegion /Order/ShippingMethods/ShippingMethod/ExcludedRegions/ExcludedRegion/WorldRegion/ PostalRegion

PredefinedRegion

Definition	A a pre-defined shipping region. (Not a state, ZIP code, or specific world region.)
Required	No
Type	String from an enumerated list
Value/Range	USContinental48States | USFull50States | USAll | WorldAll (USAll includes all states and all possessions of the U.S. such as Puerto Rico, Guam, and the
XPath	/Order/DefaultTaxTable/TaxRules/TaxRule/Regions/PredefinedRegion /Order/TaxTables/TaxTable/TaxRules/TaxRule/Regions/PredefinedRegion /Order/ShippingMethods/ShippingMethod/IncludedRegions/IncludedRegion/PreDefinedRegion /Order/ShippingMethods/ShippingMethod/ExcludedRegions/Excluded Region/ PreDefinedRegion
Definition	Container for the charge for an Item. Quantity time Price equals the charge for the Item. Price includes the Amount and CurrencyCode.
Required	Yes
Xpath	/Order/Car/Items/Item/Price