

Authorize.Net

Authorize.net

Authorize.net modules for osCommerce Online Merchant.

Chapters

 [osCommerce Online Merchant v2.3](#)

Copyright

Copyright (c) 2014 osCommerce. All rights reserved. Content may be reproduced for personal use only.

Authors

- Harald Ponce de Leon

Authorize.Net

osCommerce Online Merchant v2.3

Pages

- [Server Integration Method \(SIM\)](#)
- [Direct Post Method \(DPM\)](#)
- [Advanced Integration Method \(AIM\)](#)

Authorize.Net

Server Integration Method (SIM) v2.0

for osCommerce Online Merchant v2.3

Server Integration Method (SIM) provides a customizable, secure hosted payment form to make integration easy for Web merchants that do not have an SSL certificate, and uses scripting techniques to authenticate transactions with a unique transaction fingerprint.

The Authorize.Net Payment Gateway can handle all the steps in the secure transaction process - payment data collection, data submission and the response to the customer - while keeping Authorize.Net virtually transparent.

- Payment gateway hosted payment form employs 128-bit SSL data encryption.
- Digital fingerprints enhance security, providing multiple layers of authentication.
- Customize the look and feel of the payment gateway hosted payment form and/or receipt page.

[Apply for a Authorize.net Account](#)

Authorize.net Server Integration Method (SIM) is bundled with osCommerce Online Merchant and is also available as a separate Add-On package existing store owners can use to upgrade to.

This document can be found online at:

<http://library.oscommerce.com/Package&en&authorizenet&oscom23&sim>

Configuration Settings

Parameter	Description	Default
Enable Authorize.net Server Integration Method	Setting this parameter to <i>True</i> makes the payment method available to customers during the checkout procedure.	<i>True</i>
API Login ID	The API Login ID used to connect to the Authorize.net transaction gateway.	
API Transaction Key	The API Transaction Key used to authenticate transactions sent to Authorize.net.	
MD5 Hash	The MD5 hash value used to verify transactions with.	
Transaction Method	<p>The following transaction methods are available to process transactions with:</p> <p>Authorization The <i>Authorization</i> transaction method only verifies the card and card holder. Authorisations can be captured manually on the Authorize.net Merchant interface.</p> <p>Capture The <i>Capture</i> transaction method verifies, authorises, and transfers the funds to your Merchant account.</p>	<i>Authorization</i>
Set Order Status	The orders status will be updated to this value when a successful transaction has been made.	<i>Default Order Status</i>
Review Order Status	Set the status of orders flagged under review to this status level.	<i>Default Order Status</i>
Transaction Order Status	Transactions are logged in this private order status level.	<i>Authorize.net [Transactions]</i>

Parameter	Description	Default
Payment Zone	If set, this payment method will only be available to orders made within the defined zone.	
Transaction Server	<p>The following transaction servers are available to process transactions through:</p> <p>Live The <i>Live</i> server is used in production environments to process and receive payments from customers.</p> <p>Test The <i>Test</i> server is only used to test transactions with. No transactions are processed and no payments are made when transactions are being tested.</p>	<i>Live</i>
Transaction Mode	<p>The following transaction modes are available to process transactions through:</p> <p>Live The <i>Live</i> mode stores transactions on the gateway.</p> <p>Test The <i>Test</i> mode does not store transactions on the gateway and are not viewable in search results or reports.</p>	<i>Live</i>
Debug E-Mail Address	Send invalid transaction parameters to this e-mail address.	
Sort Order	The position to show the payment method on the checkout payment page against other available payment methods.	

Add-On Package

The separate Authorize.net Server Integration Method (SIM) Add-On package is available at:

<http://addons.oscommerce.com/info/5663>

Online Demonstration

An online demonstration is available at:

<http://demo.oscommerce.com>

For the online demonstration, Authorize.net Server Integration Method (SIM) has been configured with Authorize.net's test server environment and requires a test payment card to purchase an order.

Manual Installation (as an Add-On)

After extracting the Add-On package, copy the files located in the *catalog* directory to your osCommerce installation directory on the server. The files must be copied in the correct directory structure as extracted from the Add-On package.

The file listing is as follows:

- includes/languages/english/modules/payment/authorizenet_cc_sim.php
- includes/modules/payment/authorizenet_cc_sim.php

The standard language definitions in English are provided in the Add-On package. For additional languages, copy the English language definition file to the appropriate language directory and open the file with a text editor to edit the language definitions. The location of the language definition file must be in:

- includes/languages/*LANGUAGE*/modules/payment/authorizenet_cc_sim.php

When the files have been copied to their appropriate locations, the payment module will be available on the Administration Tool -> Modules -> Payment -> Install Module page where it can be installed, or on the Administration Tool -> Modules -> Payment page if the module was already previously installed.

If the module was already previously installed, it is important to visit the module configuration page to make sure new parameters are automatically installed in the database.

Changelog

v2.0 02-Jun-2014

- Use API 3.1.
- Label public title with test tag when in testing mode.
- Allow debug transaction information to be sent in an e-mail for failed transactions.
- Include transaction information in a private order status level.
- Store orders flagged under review in a new review order status level.
- Use redirection page on order completion if SSL is disabled, to correctly show the checkout success page through Authorize.net's SSL page request.

v1.0 11-Jan-2008

- Initial Release

Authorize.Net

Direct Post Method (DPM) v1.0

for osCommerce Online Merchant v2.3

Direct Post Method (DPM) allows developers to fully customize the experience of the entire payment flow, while simplifying PCI compliance.

The Authorize.Net Payment Gateway handles all the steps in the secure transaction while remaining virtually transparent.

- Customer data is protected with 128-bit SSL encryption.
- Digital fingerprints enhance security, providing multiple layers of authentication.

[Apply for a Authorize.net Account](#)

Authorize.net Direct Post Method (DPM) is bundled with osCommerce Online Merchant and is also available as a separate Add-On package existing store owners can use to upgrade to.

This document can be found online at:

<http://library.oscommerce.com/Package&en&authorizenet&oscom23&dpm>

Configuration Settings

Parameter	Description	Default
Enable Authorize.net Direct Post Method	Setting this parameter to <i>True</i> makes the payment method available to customers during the checkout procedure.	<i>True</i>
API Login ID	The API Login ID used to connect to the Authorize.net transaction gateway.	
API Transaction Key	The API Transaction Key used to authenticate transactions sent to Authorize.net.	
MD5 Hash	The MD5 hash value used to verify transactions with.	
Transaction Method	<p>The following transaction methods are available to process transactions with:</p> <p>Authorization The <i>Authorization</i> transaction method only verifies the card and card holder. Authorisations can be captured manually on the Authorize.net Merchant interface.</p> <p>Capture The <i>Capture</i> transaction method verifies, authorises, and transfers the funds to your Merchant account.</p>	<i>Authorization</i>
Set Order Status	The orders status will be updated to this value when a successful transaction has been made.	<i>Default Order Status</i>
Review Order Status	Set the status of orders flagged under review to this status level.	<i>Default Order Status</i>
Transaction Order Status	Transactions are logged in this private order status level.	<i>Authorize.net [Transactions]</i>

Parameter	Description	Default
Payment Zone	If set, this payment method will only be available to orders made within the defined zone.	
Transaction Server	<p>The following transaction servers are available to process transactions through:</p> <p>Live The <i>Live</i> server is used in production environments to process and receive payments from customers.</p> <p>Test The <i>Test</i> server is only used to test transactions with. No transactions are processed and no payments are made when transactions are being tested.</p>	<i>Live</i>
Transaction Mode	<p>The following transaction modes are available to process transactions through:</p> <p>Live The <i>Live</i> mode stores transactions on the gateway.</p> <p>Test The <i>Test</i> mode does not store transactions on the gateway and are not viewable in search results or reports.</p>	<i>Live</i>
Debug E-Mail Address	Send invalid transaction parameters to this e-mail address.	
Sort Order	The position to show the payment method on the checkout payment page against other available payment methods.	

Add-On Package

The separate Authorize.net Direct Post Method (DPM) Add-On package is available at:

<http://addons.oscommerce.com/info/9077>

Online Demonstration

An online demonstration is available at:

<http://demo.oscommerce.com>

For the online demonstration, Authorize.net Direct Post Method (DPM) has been configured with Authorize.net's test server environment and requires a test payment card to purchase an order.

Manual Installation (as an Add-On)

After extracting the Add-On package, copy the files located in the *catalog* directory to your osCommerce installation directory on the server. The files must be copied in the correct directory structure as extracted from the Add-On package.

The file listing is as follows:

- includes/languages/english/modules/payment/authorizenet_cc_dpm.php
- includes/modules/payment/authorizenet_cc_dpm.php

The standard language definitions in English are provided in the Add-On package. For additional languages, copy the English language definition file to the appropriate language directory and open the file with a text editor to edit the language definitions. The location of the language definition file must be in:

- includes/languages/*LANGUAGE*/modules/payment/authorizenet_cc_dpm.php

When the files have been copied to their appropriate locations, the payment module will be available on the Administration Tool -> Modules -> Payment -> Install Module page where it can be installed, or on the Administration Tool -> Modules -> Payment page if the module was already previously installed.

If the module was already previously installed, it is important to visit the module configuration page to make sure new parameters are automatically installed in the database.

Changelog

v1.0 02-Jun-2014

- Initial Release

Authorize.Net

Advanced Integration Method (AIM) v2.0

for osCommerce Online Merchant v2.3

Advanced Integration Method (AIM) is Authorize.Net's recommended connection method and offers the most secure and flexible integration for all types of transactions, including mobile, websites and other business applications.

Authorize.net Advanced Integration Method (AIM) allows merchants to host their own secure payment form on a website, mobile device, etc., and send transactions to the payment gateway using an end-to-end secure sockets layer (SSL) connection.

- Employs industry standard secure data encryption technology - 128-bit Secure Sockets Layer (SSL) protocol.
- Uses transaction key authentication for ultimate security.
- Allows control over all phases of the customer's online transaction experience.

[Apply for a Authorize.net Account](#)

Authorize.net Advanced Integration Method (AIM) is bundled with osCommerce Online Merchant and is also available as a separate Add-On package existing store owners can use to upgrade to.

This document can be found online at:

<http://library.oscommerce.com/Package&en&authorizenet&oscom23&aim>

Features

SSL Certificate Verified Transactions

All secured API transaction calls sent to Authorize.net are performed by verifying Authorize.net's SSL certificates to guarantee the identity of Authorize.net. This is performed by using the following public keys in order if they are available:

- ext/modules/payment/authorizenet/authorize.net.crt
- includes/cacert.pem (public key bundle)
- the web server configured public key bundle

A test connection link is available on the payment module configuration page which can be used to test if API transaction calls to Authorize.net can be verified from your web server. If the test connection fails, please contact your hosting provider as they may need to update your web server configuration. The SSL certificate verification can also be disabled on the payment module configuration page however it is highly discouraged to disable.

Some web hosting providers require a curl proxy to be used which can be configured on the payment module configuration page.

Configuration Settings

Parameter	Description	Default
Enable Authorize.net Advanced Integration Method	Setting this parameter to <i>True</i> makes the payment method available to customers during the checkout procedure.	<i>True</i>
API Login ID	The API Login ID used to connect to the Authorize.net transaction gateway.	
API Transaction Key	The API Transaction Key used to authenticate transactions sent to Authorize.net.	
MD5 Hash	The MD5 hash value used to verify transactions with.	
Transaction Method	<p>The following transaction methods are available to process transactions with:</p> <p>Authorization</p> <p>The <i>Authorization</i> transaction method only verifies the card and card holder. Authorisations can be captured manually on the Authorize.net Merchant interface.</p> <p>Capture</p> <p>The <i>Capture</i> transaction method verifies, authorises, and transfers the funds to your Merchant account.</p>	<i>Authorization</i>
Set Order Status	The orders status will be updated to this value when a successful transaction has been made.	<i>Default Order Status</i>
Review Order Status	Set the status of orders flagged under review to this status level.	<i>Default Order Status</i>

Parameter	Description	Default
Transaction Order Status	Transactions are logged in this private order status level.	<i>Authorize.net [Transactions]</i>
Payment Zone	If set, this payment method will only be available to orders made within the defined zone.	
Transaction Server	<p>The following transaction servers are available to process transactions through:</p> <p>Live The <i>Live</i> server is used in production environments to process and receive payments from customers.</p> <p>Test The <i>Test</i> server is only used to test transactions with. No transactions are processed and no payments are made when transactions are being tested.</p>	<i>Live</i>
Transaction Mode	<p>The following transaction modes are available to process transactions through:</p> <p>Live The <i>Live</i> mode stores transactions on the gateway.</p> <p>Test The <i>Test</i> mode does not store transactions on the gateway and are not viewable in search results or reports.</p>	<i>Live</i>
Verify SSL Certificate	Verify Authorize.net's identity when API communication calls are sent.	<i>True</i>
Proxy Server	Send API communication calls through this proxy server.	
Debug E-Mail Address	Send invalid transaction parameters to this e-mail address.	

Parameter	Description	Default
Sort Order	The position to show the payment method on the checkout payment page against other available payment methods.	

Add-On Package

The separate Authorize.net Advanced Integration Method (AIM) Add-On package is available at:

<http://addons.oscommerce.com/info/5662>

Online Demonstration

An online demonstration is available at:

<http://demo.oscommerce.com>

For the online demonstration, Authorize.net Advanced Integration Method (AIM) has been configured with Authorize.net's test server environment and requires a test payment card to purchase an order.

Manual Installation (as an Add-On)

After extracting the Add-On package, copy the files located in the *catalog* directory to your osCommerce installation directory on the server. The files must be copied in the correct directory structure as extracted from the Add-On package.

The file listing is as follows:

- ext/modules/payment/authorizenet/authorize.net.crt
- includes/languages/english/modules/payment/authorizenet_cc_aim.php
- includes/modules/payment/authorizenet_cc_aim.php

The standard language definitions in English are provided in the Add-On package. For additional languages, copy the English language definition file to the appropriate language directory and open the file with a text editor to edit the language definitions. The location of the language definition file must be in:

- includes/languages/*LANGUAGE*/modules/payment/authorizenet_cc_aim.php

When the files have been copied to their appropriate locations, the payment module will be available on the Administration Tool -> Modules -> Payment -> Install Module page where it can be installed, or on the Administration Tool -> Modules -> Payment page if the module was already previously installed.

If the module was already previously installed, it is important to visit the module configuration page to make sure new parameters are automatically installed in the database.

Changelog

v2.0 02-Jun-2014

- Use API 3.1.
- Label public title with test tag when in testing mode.
- Allow debug transaction information to be sent in an e-mail for failed transactions.
- Include transaction information in a private order status level.
- Store orders flagged under review in a new review order status level.
- Verify SSL connections with Authorize.net public key certificate.
- Add Test Connection link in module configuration page.

v1.0 11-Jan-2008

- Initial Release