

User manual

ACI WooCommerce Extension

Installation and Configuration Guide





Table of Contents

Overview	3
Pre-requisites	3
Things to Be Aware Of.....	3
Installation	4
Setup and Configuration	4
Navigating to Payment Configuration	4
Navigating to General Settings	4
Card Settings	6
Alternative Payment Settings.....	7
Fast Checkout Settings	8
Onboarding Configurations	9
Troubleshooting Guide.....	10
FAQs	11
ACI Specific Guidelines:	11

ACI WooCommerce Installation and Configuration Guide

Overview

This documentation provides detailed steps to install and configure the ACI Worldwide Payments plugin in your WooCommerce store. By integrating ACI Payments, you can offer a secure and seamless payment experience for your customers, supporting various payment methods and enhancing the overall checkout process.

Follow this guide to ensure a smooth setup and optimal configuration, enabling you to leverage the full capabilities of ACI Payments in your online store.

Pre-requisites

It is strongly recommended that installation should be completed first on a non-production environment: i.e. QA environment, before installation to production.

The following pre-requisites must be adhered to:

- PHP 8.2 or higher
- WordPress 6.6.1 or higher (tested up to 6.8.1)
- WooCommerce 9.1.2 or higher (tested up to 9.8.5)
- This plugin is designed to work with (OOTB) native WooCommerce checkout.
- If your store uses a custom checkout, additional customization by merchants or their agency will be required to ensure compatibility.

Important: *Please uninstall or deactivate any previous version of the ACI plugin before installing this version to avoid conflicts – see [Troubleshooting Guide](#).*

Things to Be Aware Of:

- Integration is designed to work with OOTB native checkout, custom checkouts require customisation by the merchant/agency to ensure compatibility
- There is a large number of plug-ins out there that could potentially cause conflicts.

Please refer to the [Troubleshooting Guide](#) at a later part of this document for more guidance.

Installation

- Download the .zip file from the ACI GitHub repository.
- Extract aci-woo-commerce-main.zip.
- Inside, locate the aci-payments folder.
- Compress the aci-payments folder into aci-payments.zip.
- Go to your WordPress Dashboard → Plugins → Add New → Upload Plugin.
- Select aci-payments.zip and click Install Now.
- Activate the plugin.

Upgrade Instructions

Important: Before upgrading to v1.2.4, please ensure there are no pending 'wp_aci_background_job_cron' events left unprocessed in your system.

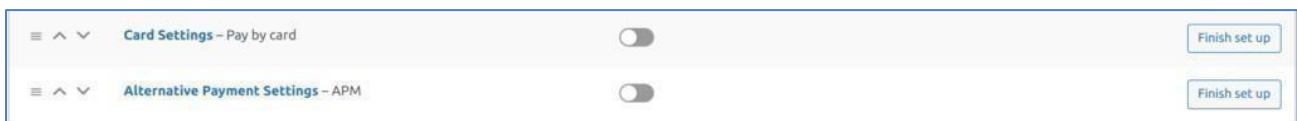
Check Scheduled Events

- Install and activate the WP CronControl plugin.
- Navigate to Tools → Cron Events.
- Search for wp_aci_batch_job_cron.
- If you see any wp_aci_batch_job_cron events in Pending status, they must be executed before you proceed with the upgrade.
- Run Events Manually.

Setup and Configuration

Navigating to Payment Configuration

- Go to: **WooCommerce > Settings > Payments**.
- Click '**Finish set up**' to see the Configuration Page for specific payment method.



Navigating to General Settings

Go to: **WooCommerce > Settings > Payments > Click 'Finish set up'**.

To see configuration page for specific payment method >Click **General Settings**

Below is the short description of the **General Settings** configuration

- **Enabled** - Yes/No
- **Environment** - Test/Live (Use 'Live' in the production)
- **Entity ID (Test/Live)** - Entity ID of your Channel or Merchant from BIP Portal
- **API Key (Test/Live)** - Access token of your Channel or Merchant from BIP Portal
- **Test Mode** - INTERNAL/EXTERNAL - This field is only applicable if the Environment is 'Test'. External mode will redirect the APM methods to their dedicated sites. For example, if the customer selects 'Klarna,' the payment

ACI WooCommerce
Installation & Configuration Guide

will be processed on Klarna's site.

- **Webhook URL** - Read only webhook URL. Use this URL to configure webhook in BIP Portal.
- **JavaScript** - Can use JavaScript based on the instruction given in the ACI documentation
<https://docs.aciworldwide.com/integrations/widget/advanced-options>
- **CSS** - Can use CSS based on the instruction given in the ACI documentation
<https://docs.aciworldwide.com/integrations/widget/customization#cssreference>
- **Webhook Decryption Key** - When completing webhook setup in ACI BIP Portal, you will get this key.
- **ACI Logger Settings** - Debug Log and Error Log (Disable/Enable)

IMPORTANT: Do not leave the “Debug Log” and “Error Log” enabled. This could have a negative impact on server performance. The Debug Log and Error Log should only be used to investigate issues when they come up and must be switched back to “Disabled” after that.

General Settings

Card Settings

Alternative Payment Settings

ACI Fast Checkout Settings

General Settings

Enabled

☒ Enabled

Environment

Test

Entity ID(Test)

Fill in Test/Live Entity ID based on the environment selected.

API Key(Test)

Fill in Test/Live API key based on the environment selected.

Test Mode

INTERNAL

Webhook URL

https://qa-aci-woo.tryzens-ignite.com/wp-json/payment/v1/w

Please configure this in ACI portal.

JavaScript

window.wpwfOptions = { style: "plain"}

JavaScript helps to customize default checkout payment page experience. [Click here](#) for more details.

Webhook Decryption Key

22DFDDB0902D7CA409C4A288A5332162935EAEFF37DECB5A

CSS

.wpwf-apple-pay-button {
width: 100%;
display: inline-block;

CSS helps to customize default checkout payment page experience. [Click here](#) for more details.

Logger Settings

Error Logging

Disable

Debug Logging

Enable

Save changes

Card Settings

- **Enabled** - Yes/No
- **Title** - Any Text (It will be displayed on the checkout payment page)
- **Charge Type** - Authorization (PA)/Sale (DB)
- **Supported Card Types** - Authorization/Sale (Ex, VISA, MASTER, AMEX). You will get the brand codes - <https://docs.aciworldwide.com/integrations/widget/customization#optionsbrands>
- **Card Type Icons** - Multiple brand image upload,
- **Card Type Icon** maximum size allowed is 100 KB and supported file types are JPG, JPEG, and PNG.
- **Save Payment Option** - Yes/No - Allow consumer to save card details for future use.

General Settings

Card Settings

Alternative Payment Settings

Card Settings ↗

Enabled

☐ Enabled

Card Payments will not be displayed in storefront if this field is disabled.

Charge Type

Auth

▼

Save Payment Option

On

▼

Selecting this will give an option for user to save the card for future use.

Card Type Icons

Choose file

No file chosen

Card Type Icon maximum size allowed is 100 KB and supported file types are JPG, JPEG, and PNG.

Save changes

Alternative Payment Settings

Enabled - Yes/No - Tick the checkbox to enable **Alternative** Payment

Alternative Payment Settings -Used to a create new payment method

- **Enabled** - Yes/No
- **Title** - Any Text (It will be displayed on the checkout payment page)
- **Charge Type** - Authorization (PA)/Sale (DB) - As applicable for APM brand.
- **APM Icon** - Single Image Upload,
- APM Icon maximum size allowed is 100 KB and supported file types are **JPG**, **JPEG**, and **PNG**.
- **Payment Key** – Payment Method Key - You will get the Payment Key from <https://docs.aciworldwide.com/integrations/widget/customization#optionsbrands>

Currently certified payment methods and its payment keys are:

- Google Pay - GOOGLEPAY
- Apple Pay - APPLEPAY
- Klarna - KLARNA_PAYMENTS_ONE
- PayPal - PAYPAL

The screenshot shows the 'Alternative Payment Settings' tab in the WooCommerce settings. At the top, there are three tabs: 'General Settings', 'Card Settings', and 'Alternative Payment Settings'. Below the tabs, the 'Alternative Payment Settings' section is active. It features a toggle for 'Enabled' which is currently set to 'No'. A note states: 'Alternative payment methods will not be displayed in storefront if this field is disabled.' Below this, there is a form for creating a new payment method. The form includes fields for 'Enabled' (a dropdown menu currently set to 'No'), 'Title' (a text input field), 'Charge Type' (a dropdown menu currently set to 'Auth'), 'APM Icon' (a file upload button labeled 'Choose file' with the text 'No file chosen'), and 'Payment Key' (a text input field). A 'Save changes' button is located at the bottom left of the form area.

Fast Checkout Settings

Enabled - Yes/No - Tick the checkbox to Enable Fast Checkout

Apple Pay Fast Checkout Settings

- **Enable Apple Pay Fast Checkout in Cart/Mini Cart** - Yes/No
- **Charge Type** - Authorization (PA)/Sale (DB)

Google Pay Fast Checkout Settings

- **Enable Google Pay Fast Checkout in Cart/Mini Cart**- Yes/No
- **Charge Type** - Authorization (PA)/Sale (DB)

General Settings

Card Settings

Alternative Payment Settings

ACI Fast Checkout Settings

ACI Fast Checkout Settings [↗](#)

Enabled ☒ Enabled
Fast Checkout Payments will not be displayed in storefront if this field is disabled.

Apple Pay Fast Checkout Settings

Enable Apple Pay Fast Checkout in Cart/Mini Cart

Yes

Charge Type

Sale

Google Pay Fast Checkout Settings

Enable Google Pay Fast Checkout in Cart/Mini Cart

Yes

Charge Type

Sale

Save changes

Onboarding Configurations

Credentials for the following admin fields to be provided by PSP or from ACI BIP Portal

- Test EntityId
- Test API Key
- Live EntityId
- Live API Key

Merchant to configure webhooks and generate the webhook decryption key in the merchant dashboard for webhook handling.

Webhook Configuration is mandatory to send payment notification to plugin platform from payment orchestration platform.

- Login to merchant dashboard
- Select your division.
- Go to the menu: Administration > Webhooks.
- Click on 'Add New Webhook' button.

- Enter relevant details

- **URL:** Get the webhook URL from WooCommerce ACI Admin configuration page.
- **Notification Types:** Select ALL PAYMENTS notification types.
- **Fields to include:** ALL.
- **Secret for encryption:** Copy the value and add in the 'Webhook Decryption Key' field under general settings of WooCommerce ACI Admin configuration page.
- **Wrapper for encrypted notification:** JSON
- Click on **Save**

Troubleshooting Guide

The following recommended steps should be shared by ACI with the onboarding merchants to help them determine any issues before reporting it to ACI.

If issues are encountered after installation, follow these steps in the following order:

1. Check Old Plugin Version

- Ensure any previous version of the ACI plugin is fully uninstalled.

2. Deactivate Any Other Plugins Temporarily

- Deactivate all other plugins
- Install the ACI Plugin and ensure it is ACTIVE
- Try just the ACI Plugin to confirm it works in isolation.

3. Re-activate Plugins Incrementally

- If the ACI Plugin is working, leave it activated
- Re-activate other plugins one at a time
- Test each re-activated plugin alongside the ACI Plugin to identify conflicts.
- If you find conflict, follow the Suggested Approach below.

Suggested Approach

After identifying the plugin that is causing the issue, we suggest one of the following options:

1. Your payment plugin should be customized by the merchant's own SI or development team.
2. ACI could commission Fuse team to perform the development and implementation on behalf of the merchant.

FAQs

1. Does this plugin support High-Performance Order storage?

Yes. The plugin is fully compatible with WooCommerce HPOS and stores order data in custom tables for better performance and scalability. You can safely enable HPOS from WooCommerce settings.

2. Does this plugin support Cart and Checkout Blocks?

No. The plugin currently relies on classic shortcodes and is not yet integrated with WooCommerce's block-based checkout experience.

3. Does this plugin support Multi store?

Not currently. The plugin is designed for single-store setups and does not yet support synchronization across multiple WooCommerce sites. Multi-store functionality may be considered later.

4. Does this plugin support Multicurrency?

Not currently. Prices and transactions are handled in the store's default currency. Integration with multicurrency plugins is under evaluation for future support.

5. Does this plugin support Multi language?

Not currently, the plugin does not include support for multilingual content or translation plugins like WPML. Language support may be added in future updates.

ACI Specific Guidelines:

- Always verify the ACI plugin in a lower environment before configuring it on the live production store.
- Make sure the plugin is installed on supported versions of PHP, WordPress, and WooCommerce. (Refer readme - <https://github.com/ACI-plugin/aci-woocommerce>)
- Decimals - Note that the ACI Plugin is designed to only accept amount with two decimal places
- Merchant should enable proper permission for WP Uploads and Upgrade folders.
- It is recommended to use Crontab to overcome the limitations of WP-Cron, which only runs during user visits to the site.
- **IMPORTANT:** Do not leave the “**Debug Log**” and “**Error Log**” enabled. This could have a negative impact on server performance. The **Debug Log** and **Error Log** should only be used to investigate issue when they come up and must be switched back to “**Disabled**” after that.
- Once the issue is identified, please share the following information with our support team:
 - **Entity ID**
 - **Relevant logs** from WooCommerce -> status -> logs
 - Screenshots or screen recordings
 - Console error details (if any)
 - **A detailed description of the issue**, including steps to reproduce