High Performance Order Storage (HPOS)

- What's New with High-Performance Order Storage?
  - Scalability
  - Reliability
  - Simplicity
- Background
- Enabling the feature
- Database tables
- Authoritative tables
- Synchronization
  - Immediate synchronization
  - Manual synchronization
  - Scheduled synchronization
  - Deletion synchronization
- Placeholder records
- Order Data Storage
- Incompatible Plugins
- Disabling HPOS

WooCommerce has traditionally stored store orders and related order information (like refunds) as custom WordPress post types or post meta records. This comes with performance issues.

High-Performance Order Storage (HPOS) also previously known as "Custom Order Tables" is a solution that provides an easy-to-understand and solid database structure - specifically designed for eCommerce needs. It uses the WooCommerce CRUD design to store order data in custom tables - optimized for WooCommerce queries with minimal impact on the store's performance.

In January 2022, we published the initial plan for the Custom Order Tables feature and since then, we've been working hard to bring the High-Performance Order Storage (HPOS) to WooCommerce

Core. In May 2022, we invited you to test the order migration process and provide feedback on how our initial work performs on real stores of varied configurations.

From WooCommerce 8.2, released on October 2023, High-Performance Order Storage (HPOS) is officially released under the stable flag and will be enabled by default for new installations.

What's New with High-Performance Order Storage?

Bringing High-Performance Order Storage (HPOS) to WooCommerce improves these three essential properties for eCommerce stores.

Scalability

The rise in the number of customers and customer orders increases the load on your store's database - making it difficult to handle customer order requests and deliver a seamless user experience.

With High-Performance Order Storage, you get dedicated tables for data like orders and order addresses and thus dedicated indexes which results in fewer read/write operations and fewer busy tables. This feature enables eCommerce stores of all shapes and sizes to scale their business to their maximum potential - without expert intervention.

Reliability

High-Performance Order Storage makes implementing and restoring targeted data backup easier. You'll no longer need to worry about losing orders, inventory numbers, or client information with reliable backup in these custom order tables. It'll also facilitate implementing read/write locks and prevent race conditions.

Simplicity

You no longer have to go through a single huge database to locate underlying data and WooCommerce entries.

With High-Performance Order Storage, you can easily browse through the separate tables and easy-to-handle entries, independent of the table `_posts`, to find data or understand the table

structure. It also lets you easily develop new plugins, implement designs for shops and products, and modify WooCommerce with more flexibility.

Background

Before the release of version 8.2, WooCommerce relied on the `_post` and `_postmeta` table structures to store order information, which has served well over the years.

However, High-Performance Order Storage introduces dedicated tables for data like orders and order addresses and thus dedicated indexes which results in fewer read/write operations and fewer busy tables. This feature enables eCommerce stores of all shapes and sizes to scale their business to their maximum potential - without expert intervention.

The order data is synced from `_posts` and `_postmeta` table to four custom order tables:

1. `_wc_orders`
2. `_wc_order_addresses`
3. `_wc_order_operational_data`
4. `_wc_orders_meta`

Enabling the feature

From WooCommerce 8.2, released on October 2023, HPOS is enabled by default for new installations. Existing stores can check How to enable HPOS

Database tables

A number of database tables are used to store order data by HPOS. The `get_all_table_names` method in the OrdersTableDataStore class will return the names of all the tables.

Authoritative tables

At any given time, while the HPOS feature is enabled, there are two roles for the involved database tables: *authoritative* and *backup*. The authoritative tables are the working tables, where order data will be stored to and retrieved from during normal operation of the store. The *backup* tables will receive a copy of the authoritative data whenever synchronization happens.

If the `woocommerce_custom_orders_table_enabled` options is set to true, HPOS is active and the new tables are authoritative, while the posts and post meta tables act as the backup tables. If the option is set to false, it's the other way around. The option can be changed via admin UI (WooCommerce - Settings - Advanced - Custom data stores).

The CustomOrdersTableController class hooks on the `woocommerce_order_data_store` filter so that `WC_Data_Store::load( 'order' );` will return either an instance of OrdersTableDataStore or an instance of WC_Order_Data_Store_CPT, depending on which are the authoritative tables.

In order to preserve data integrity, switching the authoritative tables (from the new tables to the posts table or the other way around) isn't allowed while there are orders pending synchronization.

Synchronization

Synchronization is the process of applying all the pending changes in the authoritative tables to the backup tables. *Orders pending synchronization* are orders that have been modified in the authoritative tables but the changes haven't been applied to the backup tables yet.

This can happen in a number of ways:

Immediate synchronization

If the `woocommerce_custom_orders_table_data_sync_enabled` setting is set to true, synchronization happens automatically and immediately as soon as the orders are changed in the authoritative tables.

Manual synchronization

When immediate synchronization is disabled, it can be triggered manually via command line as follows: `wp wc cot sync`. It can also be triggered programmatically as follows:

```
$synchronizer = wc_get_container()-
>get(Automattic\WooCommerce\Internal\DataStores\Orders\DataSynchronizer::class);
$order_ids = $synchronizer->get_next_batch_to_process( $batch_size );
if ( count( $order_ids ) ) {
$synchronizer->process_batch( $order_ids );
}
```

where `$batch_size` is the maximum count of orders to process.

Scheduled synchronization

If immediate synchronization gets activated (`woocommerce_custom_orders_table_data_sync_enabled` is set to true) while there are orders pending synchronization, an instance of DataSynchronizer will be enqueued using BatchProcessingController so that the synchronization of created/modified/deleted orders will happen in batches via scheduled actions. This scheduling happens inside CustomOrdersTableController, by means of hooking into `woocommerce_update_options_advanced_custom_data_stores`.

If for some reason immediate synchronization is already active but synchronization is not scheduled, a trick to restart it is to go to the settings page (WooCommerce - Settings - Advanced - Custom data stores) and hit "Save" even without making any changes. As long as "Keep the posts table and the orders tables synchronized" is checked the synchronization scheduling will happen, even if it was checked before.

If the `woocommerce_auto_flip_authoritative_table_roles` option is set to true (there's a checkbox for it in the settings page), the authoritative tables will be switched automatically once all the orders have been synchronized. This is handled by the CustomOrdersTableController class.

Deletion synchronization

Synchronization of order deletions is tricky: if an order exists in one set of tables (new tables or posts) but not in the other, it's not clear if the missing orders need to be created or if the existing orders need to be deleted. Theoretically, the orders missing from the backup tables imply the former and the orders missing from the authoritative tables imply the latter; but that's dangerous as a bug in the involved code could easily lead to the deletion of legitimate orders.

To achieve a robust order deletion synchronization mechanism the following is done. Whenever an order is deleted and immediate synchronization is disabled, a record is created in the `wp_wc_orders_meta` table that has `deleted_from` as the key and the name of the authoritative table the order was deleted from (`wp_wc_orders` or the posts table). Then at synchronization time these records are processed (the corresponding orders are deleted from the corresponding tables) and deleted afterwards.

An exception to the above are the placeholder records: these are deleted immediately when the corresponding order is deleted from `wp_wc_orders`, even if immediate synchronization is disabled.

When the "**High-Performance Order Storage**" and "**Compatibility mode**" are enabled, WooCommerce populates the HPOS tables with data from posts & postmeta tables. The synchronization between the tables is explained in detail in this document.

> You can find a deeper explanation about the synchronization between the tables in [this document about high-performance-order-storage-backward-compatibility-and-synchronization](high-performance-order-storage-backward-compatibility-and-synchronization).

Placeholder records

Order IDs must match in both the authoritative tables and the backup tables, otherwise synchronization wouldn't be possible. The order IDs that are compared for order identification and synchronization purposes are the ones from the `id` field in both the `wp_wc_orders` table and the posts table.

If the posts table is authoritative, achieving an order ID match is easy: the record in `wp_wc_orders` is created with the same ID and that's it. However, when the new orders tables are authoritative there's a problem: the posts table is used to store multiple types of data, not only orders; and by the time synchronization needs to happen, a non-order post could already exist having the same ID as the order to synchronize.

To solve this, *placeholder records* are used. Whenever the new orders tables are authoritative and immediate synchronization is disabled, creating a new order will cause a record with post type `shop_order_placehold` and the same ID as the order to be created in the posts table; this effectively "reserves" the order ID in the posts table. Then, at synchronization time, the record is filled appropriately and its post type is changed to `shop_order`.

Order Data Storage

You can switch between data stores freely to sync the data between the tables.

- If you select **"WordPress Post Tables"**, the system will save the order data within `_post` and `_postmeta` tables. The order tables are not utilized in this scenario.

Select WordPress Post Tables

- If you select **"High-Performance Order Storage"**, the system will save the order data within the new WooCommerce order tables

Select High-Performance Order Storage

- If you select **"WordPress Post Tables"** and **"Enable compatibility mode"**, the system will sync the order data between the posts/postmeta and the WooCommerce order tables.

Select WordPress Post Tables and Enable compatibility mode

Incompatible Plugins

If you are using a plugin that is not compatible with High-Performance Order Storage, then the HPOS option will be disabled under **WooCommerce > Settings > Advanced > Features**.

Incompatible plugin

- You can click on "**View and manage**" to review the list of incompatible plugins
- Or you can visit `https://example.com/wp-admin/plugins.php?plugin_status=incompatible_with_feature&feature_id=custom_order_tables` to review the list of incompatible plugins (please replace `example.com` with your site domain)

Plugins page

> **Note:** If you are using a third-party extension that isn't working properly with High-Performance Order Storage then please notify the developers of the extension and ask them to update their extension to add support for HPOS. It's up to the extension developers to add support for HPOS. We have developer resources and documentation available to help with their integration efforts.

Disabling HPOS

If you encounter problems or if you need to continue working with plugins that are not yet compatible with HPOS, then we recommend temporarily switching back to **WordPress posts storage**.

To do this, navigate to **WooCommerce ▸ Settings ▸ Advanced ▸ Features** and start by making sure that **compatibility mode** is enabled. If it was not already enabled, you may find you need to wait for some time while order data is synchronized across data-stores.

WooCommerce ▸ Settings ▸ Advanced ▸ Features Screen

Once synchronization has completed, you can select **WordPress posts storage (legacy)** as your preferred option. You can also disable compatibility mode at this point. Once you are ready to re-enable HPOS, simply follow the instructions posted at the start of this doc. Finally, remember to save this page between changes!

As noted earlier, we also strongly recommend reaching out to the support teams of any plugins that are incompatible, so they can take corrective action.

□□□ #2
□ □□□ □□□ 2026-10-31 13:12:29 UTC
□ □□□ □□□ 2026-10-31 13:12:31 UTC