

Advanced Order Manager

Version 1.0 — Full Capabilities and Features Reference

Author: Mort-Lemur

Release: Version 1.0

Platform: CE Phoenix Cart

Primary compatibility: Phoenix 1.1.0.6 and 1.1.0.7; PHP 8.2 and 8.3

Licence: GNU General Public License

1. Purpose and scope

Advanced Order Manager (AOM) provides a complete Phoenix administration workflow for creating new orders and editing existing orders. It is intended for telephone, email, mail-order and other manually entered sales where the standard Phoenix order interface does not provide sufficient flexibility.

AOM works with Phoenix's existing customer, address, product, order, tax, currency, shipping, payment and order-status data. It stores orders in the normal Phoenix tables, so orders remain available to the standard administration pages, invoices, packing slips and compatible addons.

The addon does not replace Phoenix's order system and does not require Phoenix core files to be modified.

Capabilities at a glance

Area	Capability
Order creation	Create complete Phoenix orders for existing customers or enter a new customer and address during the order process.
Order editing	Find and reopen existing orders, then revise customer details, addresses, products, quantities, prices, tax, shipping, payment, comments and status.
Catalogue and custom items	Add normal catalogue products with native attributes, or add non-catalogue custom lines with their own model, weight, quantity, price and tax.

Stock control	Follow Phoenix stock-check, checkout and stock-limiting settings; warn or block insufficient-stock orders and reconcile stock by the exact change made during editing.
Shipping and payment	Use installed Phoenix modules and friendly module titles, obtain shipping quotations or enter custom shipping.
Tax and currency	Use installed tax rates—including fractional rates—and the configured store currency and formatting.
Email and documents	Optionally send order emails and continue to the normal invoice or packing-slip workflow.

2. Order creation

Customer selection and entry

- Search existing customers using partial customer information and select a result from an automatic dropdown.
- Display useful identifying information in results, including customer number, name, company, postcode, telephone and email where available.
- Populate the order form from the selected customer's stored account and default address information.
- Create an order for a new customer by entering the required customer and address details directly in AOM.
- Create the necessary Phoenix customer, customer-information and address-book records as part of the transaction when a new customer is entered.

Billing and delivery addresses

- Enter or amend first name, surname, company, street address, suburb, town or city, postcode, country, state or province, email and telephone details.
- Use a separate delivery address when goods are being sent somewhere other than the billing address.
- Select countries from the Phoenix countries table.
- For countries with native Phoenix zones, select the state or province from a dropdown.
- For countries without defined zones, use a normal text field for state or province.
- Use the configured store country as appropriate when preparing a new order.

Products and order lines

- Search catalogue products by product name, model and manufacturer.

- Where the relevant columns exist, also search and display SKU, UPC and EAN identifiers.
- Rank exact and prefix matches ahead of less-specific matches.
- Display product price, model, manufacturer, identifiers and current stock information in the result dropdown.
- Add products to the order and change quantity, unit pricing and tax selections.
- Add multiple lines and remove unwanted lines before saving.
- Show clear product and order-total summaries as the order is assembled.

Native Phoenix product attributes

- Detect whether a catalogue product has native Phoenix attributes.
- Open an option-selection window before adding an attributed product.
- Present the available option groups and option values using their configured language names.
- Include attribute price adjustments in the displayed and saved order-line price.
- Store selections in the normal `orders_products_attributes` records.
- Restore the saved selections when an existing order is reopened for editing.

Custom non-catalogue items

- Add products or services that do not exist in the Phoenix catalogue.
- Enter a custom name, model, weight, quantity, unit price and tax rate.
- Prefix a supplied custom model with **C-**, making custom lines easy to identify.
- Leave the model blank when no custom model is supplied.
- Interpret custom weight using the store's normal weight unit rather than assuming kilograms.
- Exclude custom items from catalogue stock adjustment.

3. Existing-order search and editing

Finding an order

- Open an order directly when an order ID is supplied in the AOM Edit URL.
- Search by order number, customer name, customer first name, customer surname or postcode.
- Return multiple matching orders rather than collapsing people who share a surname into one result.
- Display matching orders in descending order-number order, making the newest matches easy to find.
- Show order number, customer name and supporting details in a scrollable result dropdown.

Editable order information

- Load the complete existing order into the AOM interface.
- Amend customer, billing and delivery details without requiring the order to be recreated.
- Add new catalogue or custom products.
- Remove existing products.
- Change product quantities, unit prices and tax treatment.
- Retain and restore native attribute selections.
- Change shipping method and shipping charge.
- Change the recorded payment method.
- Recalculate and replace the saved Phoenix order totals.
- Return stock for removed quantities and deduct stock only for additional quantities.

Status, comments and history

- Choose from the order statuses installed in the Phoenix database.
- Add an order comment while editing.
- Create a normal order-status-history entry when a status change or comment requires one.
- View existing status history, including date, status, comments and notification state.
- Choose whether the customer should be notified when the new history entry is created.
- Avoid generating an unnecessary notification when neither status nor comments create a new history event.

4. Stock and product-status behaviour

AOM follows the relevant Phoenix stock and checkout configuration instead of applying a separate stock policy.

Phoenix setting	AOM behaviour
Check stock level enabled	Current catalogue stock is shown during product search. AOM calculates whether the requested additional quantity would take stock below zero.
Check stock level disabled	Insufficient-stock warnings and blocks are not applied by AOM.
Allow checkout enabled	AOM warns that the order will take stock below zero and asks the administrator to confirm before saving.
Allow checkout disabled	AOM prevents the order from being saved when the additional quantity requires more stock than is available.

Subtract stock enabled	Catalogue products_quantity is adjusted when an order is created or edited.
Subtract stock disabled	Order creation does not reduce products_quantity.
Out-of-stock product marking	AOM uses Phoenix's configured marker and correctly renders text or icon-based markers in product-search results.

Stock reconciliation rules

- New orders deduct the ordered quantity of normal catalogue products when stock subtraction is enabled.
- Editing an order compares the old and new quantities and applies only the difference.
- Increasing a quantity deducts only the additional units.
- Reducing a quantity returns the difference to stock.
- Removing a catalogue line returns its previously ordered quantity.
- Adding a new line deducts its full quantity.
- Repeated lines for the same parent product are combined when checking available stock.
- The existing edited-order quantity is recognised as already deducted, preventing false shortage warnings.
- Phoenix products_ordered values are reconciled alongside stock quantities.
- Where Phoenix settings require it, a product that falls out of stock can be disabled.
- Stock returned during an edit does not automatically re-enable a product that an administrator may have deliberately disabled.

Supported stock scope: Version 1.0 manages native Phoenix parent-product stock. QTPro and other third-party attribute-stock tables are outside the supported scope; attribute combinations do not receive separate stock deductions.

5. Tax, currency and totals

- Read tax classes and tax rates from the Phoenix database.
- Support fractional tax percentages, including rates such as 25.5%.
- Apply the selected product tax rate to catalogue and custom order lines.
- Calculate product subtotal, product tax, shipping, shipping tax and grand total.
- Reconstruct an existing order's saved gross shipping amount and shipping tax rate when it is reopened.
- Save standard Phoenix order-total rows with the correct sort order and values.
- Use the configured store currency code, symbol, decimal precision and display format rather than hard-coding GBP.

- Preserve the order currency and currency value in the Phoenix order record.
- Update calculated totals immediately in the browser as products, quantities, prices, tax or shipping change.

6. Shipping

Installed shipping modules

- Discover enabled Phoenix shipping modules.
- Display friendly, customer-facing module and method titles where available.
- Request a quotation from the selected module using the order contents and destination.
- Provide destination country, country codes, zone, zone ID and address details for modules that calculate geographically.
- Supply a compatible temporary cart and order context for shipping modules that inspect Phoenix cart or order data.
- Use catalogue product weight and custom-item weight when calculating a quotation.
- Apply Phoenix tare weight or percentage padding and maximum-box-weight calculations.
- Calculate shipping tax from the module tax class or use tax supplied by the quotation where appropriate.

Custom shipping

- Enter a custom shipping title and charge when an installed module is not suitable.
- Select the appropriate shipping tax rate.
- Retain a saved custom or existing shipping method when an order is reopened.

Shipping quotations depend on each installed module's own requirements. Version 1.0 prepares the normal weight, cart, order and destination context, but an unusually specialised third-party module may require separate compatibility work.

7. Payment

- Discover installed and enabled Phoenix payment modules.
- Display friendly configured payment titles rather than internal module codes where available.
- Select the payment method to record on a newly created order.
- Retain or change the recorded payment method when editing an order.
- Use a custom or fallback payment label where required by the interface.

AOM records the payment method on the order. It does not itself capture a card payment, issue a refund or replace the payment provider's own administration workflow.

8. Email and documents

- Optionally send a new-order confirmation email after an order is created successfully.
- Keep new-order email sending off by default unless selected by the administrator.
- Build the email from the saved Phoenix order, products, attributes, totals and address information.
- Use store and customer information from Phoenix configuration and the saved order.
- On Edit, optionally notify the customer when a new status-history entry is created.
- Record whether the customer was notified in order history.
- Continue from a newly created order to the standard Phoenix invoice workflow.
- Continue to the standard Phoenix packing-slip workflow.

9. Administration and usability

- Provide a dedicated Advanced Order Manager landing page with clear Create and Edit choices.
- Add Advanced Order Manager to the Phoenix Orders administration menu.
- Add an **Edit in AOM** action to the selected order's normal Phoenix information box.
- Support the information-box data shapes used by the relevant Phoenix 1.1.0.6 and 1.1.0.7 administration implementations.
- Preserve the current native Orders list page number in the AOM edit link for future return navigation.
- Provide a Return to Orders button on the Edit page.
- Use debounced automatic searches to avoid sending a request on every immediate keystroke.
- Use scrollable dropdowns for longer customer, product and order result sets.
- Warn before leaving an Add or Edit page when there are unsaved changes.
- Clear the unsaved-change warning after a successful save or an intentional navigation action.
- Version JavaScript URLs from the file content so updated scripts are loaded instead of stale browser-cached copies.
- Keep AOM styling scoped so it does not alter unrelated Phoenix pages.

10. Data integrity and technical design

- Use the standard Phoenix database connection and administration bootstrap.
- Store customers, addresses, orders, products, attributes, totals and history in native Phoenix tables.
- Wrap save operations in a database transaction.
- Roll back the transaction when a save-stage failure is detected.
- Validate stock server-side before creating customer or order records, so browser-side checks are not the only protection.

- Return structured success, warning or error information from the shared save engine.
- Use one shared `aom_save_order()` workflow for Add and Edit, reducing differences between the two save paths.
- Use the database-aware `aom_perform_insert()` helper for generated INSERT statements.
- Escape submitted SQL values through the Phoenix database object.
- Cast identifiers and numeric values before they are used in database queries or calculations.
- Use JSON endpoints for customer, order, product, attribute and shipping requests.
- Render product and customer database text safely in browser search results.
- Keep interface wording in Phoenix language files for the main Add, Edit and manager pages.
- Require no database-schema changes for AOM itself.
- Require no Phoenix core-file replacements.

11. Integration points

- **Native Phoenix Orders:** Orders remain visible and usable in the normal Orders administration page.
- **Invoices and packing slips:** AOM-created orders use the normal Phoenix document workflow.
- **Products and attributes:** AOM reads the existing product catalogue, native attributes and installed tax classes.
- **Shipping and payment modules:** Enabled modules are discovered and presented within AOM.
- **Countries and zones:** Address controls use native country and zone records.
- **Orderlist addon:** An Orderlist order-number link can optionally be directed to `manual_order_edit.php?oID=...`
- **Other order tools:** Because AOM stores native order records, compatible reporting and fulfilment tools can continue to use them.

12. Compatibility and installation model

Item	Version 1.0 position
CE Phoenix Cart	Developed for the Phoenix 1.1.0.x generation, with primary use and compatibility work covering 1.1.0.6 and 1.1.0.7.
PHP	Designed and beta-tested for PHP 8.2 and PHP 8.3.
Database	Uses the existing Phoenix database structure; no AOM-specific schema installation is required.
Core files	No Phoenix core file modification is required.
Admin directory	Files supplied under <code>admin</code> are uploaded into the shop's actual, possibly renamed,

	administration directory.
Hook	The order-information-box hook is uploaded beneath <code>includes/hooks/admin/orders</code> from the package's catalogue tree.
CSS	The packaged AOM CSS block is copied into the site's existing override <code>user.css</code> ; it is not intended to overwrite that file.

13. Deliberate exclusions and limitations

Version 1.0 is an order creation and editing system. The following are not included:

- QTPro or third-party attribute-combination stock management.
- Returns or RMA processing.
- Refund processing or payment-provider transaction capture.
- Accounting-system integration.
- Bulk order import or export.
- Warehouse management.
- Subscription or recurring-order management.
- Automatic re-enabling of products when stock is returned.
- Guaranteed compatibility with every bespoke third-party shipping module without module-specific testing.

14. Suggested first-use verification

1. Search for and select an existing customer.
2. Create a test order for a new customer and confirm the account and default address records.
3. Add a standard catalogue item and an item with native attributes.
4. Add a custom item with a model and weight.
5. Confirm product tax, shipping tax, currency formatting and final totals.
6. Test an installed shipping module against a controlled destination.
7. Create an order and verify stock, products ordered, totals and order history.
8. Reopen the order, increase and reduce quantities, remove a line and confirm difference-based stock reconciliation.
9. Test both Phoenix insufficient-stock policies: checkout allowed and checkout disallowed.
10. Test optional new-order and edited-order notifications using a controlled email address.
11. Open the invoice and packing slip and confirm all saved information is displayed correctly.