



## Order Editor v1.2.8

Version author : [@BrockleyJohn](https://twitter.com/BrockleyJohn) [john@sewebsites.net](mailto:john@sewebsites.net)

[the latest in a long line of contributors – see attribution later]

Version 1.2.8 26/12/2019

### Overview

This addon offers the facility to update orders in admin. This version brings full function on Phoenix 1.0.4.0 with no core changes and adds an option for whether the Edit button in order list info box gets changed to the Editor. Up to three very simple core changes on earlier osc versions optionally add extra functionality.

### Compatibility

Updated for compatibility with CE Phoenix 1.0.4.0 and Frozen versions of the community responsive build of osCommerce (2.3.4BS); choose the version of modified files that goes with your shop.

Also compatible with all previous responsive builds and 2.3 mainstream releases of osCommerce, except:

- Due to filenames constants being deprecated, the page names orders.php and account\_history\_info.php are hard-coded. If they are non-standard in your store you will need to change the hard-coding.
  - o except some non-standard filename definitions are used to check if addons are installed; these tests have been left alone and if you use any of the addons you'll need to check them, eg.
- Similarly path definitions like DIR\_WS\_INCLUDES from configure.php are hard-coded in the order editor
- The button changes in admin > orders use the hook system installed for the paypal app. If your store is on an older version and does not have these applied, the easiest solution is to edit the buttons manually; see later. To find out if an older store has the hook system installed, look in the top of your admin orders.php file. If it has the following line after the include of application\_top.php, you have the hooks installed:

```
$OSCOM_Hooks->register('orders');
```

***PHOENIX does not have this line, the hook call is in application top instead***

## Installation (new)

As with all addons, backup your store files and database before installing. Copy the files to your store – making allowances if your admin directory is called something different. Then

- If you want the shipping method to be recorded against the order on creation, go to menu admin > modules > header tags and install the module “Record Shipping Method on Order”. This will allow the order editor to identify the current shipping method when you go to edit an order, but it’s not essential.

If you are installing on Phoenix 1.0.4.0 you have finished and you can now run the Order Editor to install the settings. On earlier versions check if you want to do the following:

- If your store records credit card details against orders (an unusual practice nowadays), you’ll need an extra function so that you can choose the credit card payment module in the configuration settings. Then the order editor will display these fields when that payment module is selected. On osc versions before Phoenix 1.0.4.0 you need to make an edit: in the file admin/includes/functions/general.php, paste the following before the closing ?> (or at the end if it doesn’t have one)

```
////////create a pull down for all payment installed payment methods for Order Editor configuration
```

```
// Get list of all payment modules available
function tep_cfg_pull_down_payment_methods() {
    global $language;
    $enabled_payment = array();
    $module_directory = DIR_FS_CATALOG_MODULES . 'payment/';
    $file_extension = '.php';

    if ($dir = @dir($module_directory)) {
        while ($file = $dir->read()) {
            if (!is_dir( $module_directory . $file)) {
                if (substr($file, strrpos($file, '.')) == $file_extension) {
                    $directory_array[] = $file;
                }
            }
        }
        sort($directory_array);
        $dir->close();
    }
}
```

```
// For each available payment module, check if enabled
for ($i=0, $n=sizeof($directory_array); $i<$n; $i++) {
    $file = $directory_array[$i];

    include(DIR_FS_CATALOG_LANGUAGES . $language . '/modules/payment/' . $file);
    include($module_directory . $file);

    $class = substr($file, 0, strrpos($file, '.'));
    if (tep_class_exists($class)) {
```

```

$module = new $class;
if ($module->check() > 0) {
    // If module enabled create array of titles
    $enabled_payment[] = array('id' => $module->title, 'text' => $module->title);
}
}
}

$enabled_payment[] = array('id' => 'Other', 'text' => 'Other');

    //draw the dropdown menu for payment methods and default to the order value
    return tep_draw_pull_down_menu('configuration_value', $enabled_payment, '', '');
}

```

/////end payment method dropdown

- This version of the addon uses the standard hooks in admin > orders to add an edit button to the view of an order. With an additional hook it will change the Edit button on the orders list view to go straight to the order editor (clicking the icon on the left still goes to the usual order view). To add the extra hook, go to the bottom of admin / orders.php and add the line before the last curly brace as shown:

```

<?php
    echo $OSCOM_Hooks->call('orders', 'orderList'); // ADD THIS
}

require('includes/template_bottom.php');
require('includes/application_bottom.php');
?>

```

- If you don't have the hook system installed in your store (see earlier for check), there are two edits to make in admin orders.php:

Find at about line 106:

```

<td class="smallText" align="right"><?php echo tep_draw_button(IMAGE_ORDERS_INVOICE, 'document', tep_href_link(FILENAME_ORDERS_INVOICE,
'oid=' . $HTTP_GET_VARS['oid']), null, array('newwindow' => true)) . tep_draw_button(IMAGE_ORDERS_PACKINGSLIP, 'document',
tep_href_link(FILENAME_ORDERS_PACKINGSLIP, 'oid=' . $HTTP_GET_VARS['oid']), null, array('newwindow' => true)) .
tep_draw_button(IMAGE_BACK, 'triangle-l-w', tep_href_link(FILENAME_ORDERS, tep_get_all_get_params(array('action')))); ?></td>

```

REPLACE with this code:

```

<td class="smallText" align="right"><?php echo tep_draw_button(IMAGE_EDIT, 'document', tep_href_link('edit_orders.php', 'oid=' .
$_GET['oid']), null, array('newwindow' => true)) . tep_draw_button(IMAGE_ORDERS_PACKINGSLIP, 'document',
tep_href_link(FILENAME_ORDERS_PACKINGSLIP, 'oid=' . $HTTP_GET_VARS['oid']), null, array('newwindow' => true)) .
tep_draw_button(IMAGE_BACK, 'triangle-l-w', tep_href_link(FILENAME_ORDERS, tep_get_all_get_params(array('action')))); ?></td>

```

Find at about line 410:

```
$contents[] = array('align' => 'center', 'text' => tep_draw_button(IMAGE_EDIT, 'document', tep_href_link(FILENAME_ORDERS, tep_get_all_get_params(array('oID', 'action')) . 'oID=' . $oInfo->orders_id . '&action=edit')) . tep_draw_button(IMAGE_DELETE, 'trash', tep_href_link(FILENAME_ORDERS, tep_get_all_get_params(array('oID', 'action')) . 'oID=' . $oInfo->orders_id . '&action=delete')));  
$contents[] = array('align' => 'center', 'text' => tep_draw_button(IMAGE_ORDERS_INVOICE, 'document', tep_href_link(FILENAME_ORDERS_INVOICE, 'oID=' . $oInfo->orders_id, null, array('newwindow' => true)) . tep_draw_button(IMAGE_ORDERS_PACKINGSLIP, 'document', tep_href_link(FILENAME_ORDERS_PACKINGSLIP, 'oID=' . $oInfo->orders_id, null, array('newwindow' => true))));
```

REPLACE with this code:

```
$contents[] = array('align' => 'center', 'text' => tep_draw_button(IMAGE_EDIT, 'document', tep_href_link('edit_orders.php', tep_get_all_get_params(array('oID', 'action')) . 'oID=' . $oInfo->orders_id) . tep_draw_button(IMAGE_DETAILS, 'document', tep_href_link(FILENAME_ORDERS, tep_get_all_get_params(array('oID', 'action')) . 'oID=' . $oInfo->orders_id . '&action=edit')) . tep_draw_button(IMAGE_DELETE, 'trash', tep_href_link(FILENAME_ORDERS, tep_get_all_get_params(array('oID', 'action')) . 'oID=' . $oInfo->orders_id . '&action=delete')));  
$contents[] = array('align' => 'center', 'text' => tep_draw_button(IMAGE_ORDERS_INVOICE, 'document', tep_href_link(FILENAME_ORDERS_INVOICE, 'oID=' . $oInfo->orders_id, null, array('newwindow' => true)) . tep_draw_button(IMAGE_ORDERS_PACKINGSLIP, 'document', tep_href_link(FILENAME_ORDERS_PACKINGSLIP, 'oID=' . $oInfo->orders_id, null, array('newwindow' => true))));
```



## Update from v1.2.7 to 1.2.8

Changes:

- No core changes on Phoenix 1.0.4.0 – you should revert admin/orders.php and admin/includes/functions/general.php to the core versions
- Add option for whether to change Edit button on Order list Info panel to editor

Changed & new files. Upload to your store:

- catalog new files/admin/edit\_orders.php
- catalog new files/admin/includes/languages/english/edit\_orders.php
- catalog new files/includes/hooks/admin/siteWide/listOrders.php
- catalog new files/includes/hooks/admin/siteWide/paymentMethodPulldown.php
- catalog new files/includes/modules/ hooks/admin/orders\_edit\_order.php

## Update from v1.2.6 to 1.2.7

Bug fixes:

- hook to inject Edit Order button wasn't working on Phoenix 1.0.4.0
- redirect after installing settings was missing ? before params
- if you ran Order Editor without installing the header tags module you got a database error
- if you added another line for a product already on the order (even with different options chosen), price and weight totals for shipping modules were calculated wrongly

Changed files. Upload to your store:

- catalog new files/admin/edit\_orders.php
- catalog new files/admin/edit\_orders\_ajax.php
- catalog new files/admin/order\_editor/cart.php
- catalog new files/admin/order\_editor/order.php
- catalog new files/admin/order\_editor/functions.php
- catalog new files/includes/modules/ hooks/admin/orders\_edit\_order.php

## Update from v1.2.5 to 1.2.6

There are changes to two files to fix bugs. Upload these to your store:

- catalog new files/admin/edit\_orders.php
- catalog new files/includes/modules/header\_tags/ht\_record\_order\_shipping\_module.php

## Settings

- The settings for the Order Editor are created in their own group under configuration

## Support

<https://forums.oscommerce.com/topic/54032-new-complete-order-editing-tool/>



## Version history

V1.2.7 – Phoenix 1.0.4.0 compatibility

V1.2.6 – Phoenix compatibility

v1.2.5.1 - move images folder under admin/order\_editor

v1.2.5 – change installation to avoid core code changes and use of phpmyadmin as far as possible, plug ajax vulnerabilities

## Attribution

Top of the code:

```
$Id: edit_orders.php v5.0.5 08/27/2007 djmonkey1 Exp $  
...
```

The original Order Editor contribution was written by Jonathan Hilgeman of SiteCreative.com

Much of Order Editor 5.x is based on the order editing file found within the MOECTOE Suite Public Betas written by Josh DeChant

Many, many people have contributed to Order Editor in many, many ways. Thanks go to all- it is truly a community project.

...