

Current Checkout Process

Plugins used:

1. Checkout Field Editor for WooCommerce (Pro) (Additional Fields added)
2. WooCommerce Address Validation (Loqate integration)
3. WooCommerce Address Summary Plugin (Bespoke for address summary)

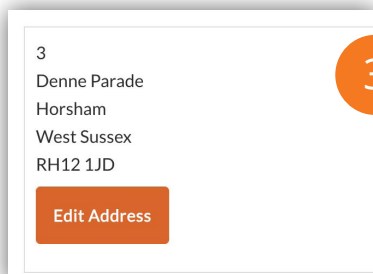
1. Checkout Fields

Here's the current list of billing/shipping checkout fields:

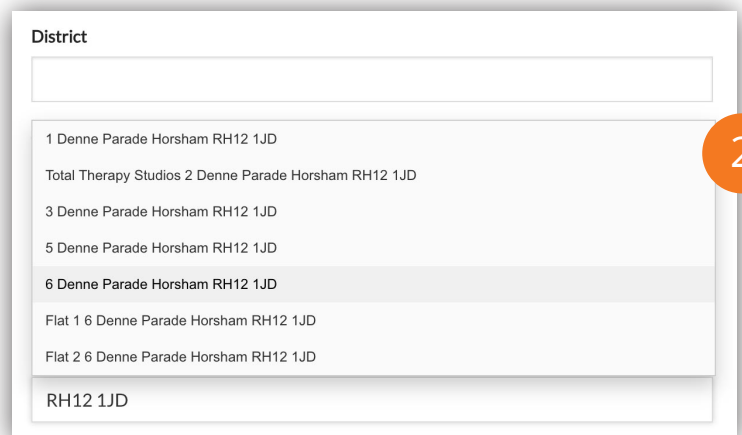
| | |
|----------------------|---------------------|
| billing_company | Company name |
| data_billing_custom | Building/House Name |
| billing_number | House/Flat Number |
| billing_address_1 | Address |
| billing_district | District |
| billing_city | Town / City |
| billing_state | County |
| billing_passdata | Postcode |
| billing_mobile | Mobile |
| billing_phonew | Landline Telephone |
| shipping_company | Company name |
| data_shipping_custom | Building/House Name |
| shipping_number | House/Flat Number |
| shipping_address_1 | Address |
| shipping_district | District |
| shipping_city | Town / City |
| shipping_state | County |
| shipping_passdata | Postcode |
| shipping_mobile | Mobile |
| shipping_phonew | Landline Telephone |

2. Loqate Lookup

Customer inputs postcode which triggers Loqate suggestion, which they then choose.



3
Denne Parade
Horsham
West Sussex
RH12 1JD
[Edit Address](#)



District

1 Denne Parade Horsham RH12 1JD
Total Therapy Studios 2 Denne Parade Horsham RH12 1JD
3 Denne Parade Horsham RH12 1JD
5 Denne Parade Horsham RH12 1JD
6 Denne Parade Horsham RH12 1JD
Flat 1 6 Denne Parade Horsham RH12 1JD
Flat 2 6 Denne Parade Horsham RH12 1JD
RH12 1JD

3. Address Summary

Once they have chosen their address, the Address Summary plugin cuts in and presents their address like this...

Notes:

Address Summary was created to try and stop people playing with their address once it was selected from Loqate.

However, the address needs to be editable as sometimes Loqate is not quite correct and the customer needs to amend what has been chosen.

Current functions.php snippets

```
/** Customizes the Loqate (Addressy) field mapping for WooCommerce checkout.*/
```

```
function sv_wc_address_validation_addressy_custom_mapping( $params ) {
```

```
    // Billing: postcode drives the search
```

```
    $params['billing'] = array(
```

```
        array( 'element' => 'billing_postcode', 'field' => 'PostalCode', 'mode' => 'SEARCH' ),
```

```
        array( 'element' => 'billing_company', 'field' => 'Company', 'mode' => 'POPULATE' ),
```

```
        array( 'element' => 'data_billing_custom','field'=> '{SubBuilding} {BuildingName}', 'mode' => 'POPULATE' ),
```

```
        array( 'element' => 'billing_number', 'field' => 'BuildingNumber', 'mode' => 'POPULATE' ),
```

```
        array( 'element' => 'billing_address_1', 'field' => '{SecondaryStreet}, {Street}', 'mode' => 'POPULATE' ),
```

```
        array( 'element' => 'billing_district', 'field' => 'District', 'mode' => 'POPULATE' ),
```

```
        array( 'element' => 'billing_city', 'field' => 'City', 'mode' => 'POPULATE' ),
```

```
        array( 'element' => 'billing_state', 'field' => 'Province', 'mode' => 'POPULATE' ),
```

```
    );
```

```
    // Shipping: postcode drives the search
```

```
    $params['shipping'] = array(
```

```
        array( 'element' => 'shipping_postcode', 'field' => 'PostalCode', 'mode' => 'SEARCH' ),
```

```
        array( 'element' => 'shipping_company', 'field' => 'Company', 'mode' => 'POPULATE' ),
```

```
        array( 'element' => 'data_shipping_custom','field'=> '{SubBuilding} {BuildingName}', 'mode' => 'POPULATE' ),
```

```
        array( 'element' => 'shipping_number', 'field' => 'BuildingNumber', 'mode' => 'POPULATE' ),
```

```
        array( 'element' => 'shipping_address_1', 'field' => '{SecondaryStreet}, {Street}', 'mode' => 'POPULATE' ),
```

```
        array( 'element' => 'shipping_district', 'field' => 'District', 'mode' => 'POPULATE' ),
```

```
        array( 'element' => 'shipping_city', 'field' => 'City', 'mode' => 'POPULATE' ),
```

```
        array( 'element' => 'shipping_state', 'field' => 'Province', 'mode' => 'POPULATE' ),
```

```
    );
```

```
    return $params;
```

```
}
```

```
add_filter( 'wc_address_validation_addressy_addresses', 'sv_wc_address_validation_addressy_custom_mapping', 20 );
```

// Basic disable autocomplete checkout WooCommerce

```
add_action( 'wp_footer', function () {
    if ( ! is_checkout() ) return;
    ?>
    <script>
    jQuery(function($){
        $('form.checkout').attr('autocomplete','off');
        $('form.checkout').find('input, textarea, select').attr({
            autocomplete: 'off',
            autocapitalize: 'off',
            autocorrect: 'off',
            spellcheck: 'false'
        });
    });
    </script>
    <?php
});
```

// Minimal Loqate autofill postcode for WC/shipping.

```
add_action( 'wp_footer', function () {
    if ( ! is_checkout() ) return;
    ?>
    <script>
    jQuery(function($){
        $('#billing_postcode, #shipping_postcode').attr({
            autocomplete:'new-password',
            autocapitalize:'off',
            autocorrect:'off',
            spellcheck:'false',
            inputmode:'text',
            'data-lpignore':'true',
            'data-form-type':'other'
        });
        $(document).on('input change', '#billing_postcode, #shipping_postcode', function(){
            var val = $.trim($(this).val()).toUpperCase();
            if ($(this).val() !== val) $(this).val(val);
        });
    });
    </script>
    <?php
});
```

```
/**
 * Force hidden shipping options until address is entered */
add_filter('option_woocommerce_shipping_cost_requires_address', function($value){
    return 'yes';
}, PHP_INT_MAX);
```

Address Summary Plugin

woocommerce-address-summary-plugin.php:

```
<?php
/**
 * Plugin Name: WooCommerce Address Summary Plugin
 * Description: Shows a summary box for billing and shipping addresses in WooCommerce
checkout, with "Edit Address" functionality and Loqate support.
 * Version: 1.0.0
 */

if ( ! defined( 'ABSPATH' ) ) exit;

/**
 * Output empty summary containers after billing & shipping forms.
 */
add_action('woocommerce_after_checkout_billing_form', function() {
    echo '<div id="billing-address-summary" style="display:none; border:1px solid #ddd;
padding:10px; margin-bottom:10px;" title="Billing Address Summary"></div>';
});
add_action('woocommerce_after_checkout_shipping_form', function() {
    echo '<div id="shipping-address-summary" style="display:none; border:1px solid #ddd;
padding:10px; margin-bottom:10px;" title="Shipping Address Summary"></div>';
});

/**
 * Enqueue JS for address summary logic
 */
add_action('wp_enqueue_scripts', function() {
    // Only on checkout
    if ( function_exists('is_checkout') && ! is_checkout() ) return;

    wp_enqueue_script(
        'wc-address-summary',
        plugins_url('wc-address-summary.js', __FILE__),
        array('jquery'),
        '1.0.0',
        true
    );
}, 20);

/**
 * Inline helper JS that moves the summary boxes next to the email field (keeps original behav-
ior).
 * This is intentionally limited in scope and does not attempt to touch Save button placement.
 */
add_action('wp_footer', function(){
    if( ! function_exists('is_checkout') || ! is_checkout() ) return;
    ?>
    <script>
jQuery(function($){
    function moveSummaryAfterEmail(type) {
        var $summary = $('#' + type + '-address-summary');
```

```

var $email = $('#' + type + '_email_field');
if ($summary.length && $email.length) {
    if (!$summary.is($email.next())) {
        $summary.detach();
        $email.after($summary);
    }
} else {
    setTimeout(function(){ moveSummaryAfterEmail(type); }, 200);
}
}

var billingWrapper = document.querySelector('.woocommerce-billing-fields__field-wrapper');
if (billingWrapper) {
    var billingObserver = new MutationObserver(function() {
        moveSummaryAfterEmail('billing');
    });
    billingObserver.observe(billingWrapper, {childList: true, subtree: false});
    moveSummaryAfterEmail('billing');
}
var shippingWrapper = document.querySelector('.woocommerce-shipping-fields__field-wrapper');
if (shippingWrapper) {
    var shippingObserver = new MutationObserver(function() {
        moveSummaryAfterEmail('shipping');
    });
    shippingObserver.observe(shippingWrapper, {childList: true, subtree: false});
    moveSummaryAfterEmail('shipping');
}
});
</script>
<?php
});

```

wc-address-summary.js:

```

jQuery(function($){
    'use strict';

    var fields = {
        billing: [
            'billing_company','data_billing_custom','billing_number','billing_address_1',
            'billing_district','billing_city','billing_state','billing_postcode','billing_pasdata','billing_country'
        ],
        shipping: [
            'shipping_company','data_shipping_custom','shipping_number','shipping_address_1',
            'shipping_district','shipping_city','shipping_state','shipping_postcode','shipping_
            pasdata','shipping_country'
        ]
    };

    var postcodePattern = /^[A-Z]{1,2}\d[A-Z0-9]?[s?]\d[A-Z]{2}$/;

    var lastPC = { billing:"", shipping:" };
    var updateTimer = { billing:null, shipping:null };

```

```
function normalizePostcode(v){ return jQuery.trim(v | "").toUpperCase(); }
```

```
function resolvePostcodeFieldId(type){  
  var pcd = (type==='billing') ? 'billing_postcode' : 'shipping_postcode';  
  var pdId = (type==='billing') ? 'billing_passdata' : 'shipping_passdata';  
  if ($('#'+pcd).length) return pcd;  
  if ($('#'+pdId).length) return pdId;  
  var $id = $('[id^="'+pdId+'"]'); if ($id.length) return $id.first().attr('id');  
  var $name = $('[name^="'+pdId+'"]'); if ($name.length) return $name.first().attr('id');  
  return pcd;  
}
```

```
function getPostcodeValue(type){  
  var realId = (type==='billing') ? 'billing_postcode' : 'shipping_postcode';  
  var passId = resolvePostcodeFieldId(type);  
  var realVal = normalizePostcode($('#'+realId).val());  
  if (realVal) return realVal;  
  if (passId && passId!==realId) {  
    var passVal = normalizePostcode($('#'+passId).val());  
    if (passVal) return passVal;  
  }  
  return "";  
}
```

```
function requestUpdateCheckout(type) {  
  var pc = getPostcodeValue(type);  
  if (!pc || pc === lastPC[type]) return;  
  lastPC[type] = pc;  
  clearTimeout(updateTimer[type]);  
  updateTimer[type] = setTimeout(function(){  
    $(document.body).trigger('update_checkout');  
  }, 200);  
}
```

```
function getFieldValues(type){  
  var vals=[];  
  $.each(fields[type], function(_, fieldId){  
    if (fieldId.endsWith('_country')) return;  
    if (fieldId.endsWith('_postcode') || fieldId.endsWith('_passdata')) return;  
    var $el = $('#'+fieldId);  
    if (!$el.length) return;  
    var val = $el.is('select') ? $el.find('option:selected').text() : $el.val();  
    val = $.trim(val | "");  
    if (val) vals.push(val);  
  });  
  var pc = getPostcodeValue(type);  
  if (pc) vals.push(pc);  
  return vals;  
}
```

```
function isAddressPopulated(type){  
  var cityField = (type==='billing') ? 'billing_city' : 'shipping_city';  
  var $city = $('#'+cityField);
```

```

var pc = getPostcodeValue(type);
if (!pc || pc.length < 5 || !postcodePattern.test(pc)) return false;
if (!$city.length || !$city.trim()) return false;
return true;
}

```

```

// Deterministic placement: below last name, above mobile. No anchor div.

```

```

function placeShippingSummary(){
  var $summary = $('#shipping-address-summary');
  if (!$summary.length) return;

```

```

  // Clear floated first/last name fields
  $summary.css('clear', 'both');

```

```

  var $last = $('#shipping_last_name_field');
  var $mobile = $('#shipping_mobile_field');

```

```

  // If both exist, put summary between them

```

```

  if ($last.length && $mobile.length) {
    // Ensure it's after last name
    $summary.insertAfter($last);
    // If something else (plugin) moved it, enforce before mobile too
    if ($summary.nextAll('#shipping_mobile_field').length === 0) {
      $summary.insertBefore($mobile);
    }
    return;
  }

```

```

  // Fallbacks

```

```

  if ($last.length) {
    $summary.insertAfter($last);
    return;
  }
  if ($mobile.length) {
    $summary.insertBefore($mobile);
    return;
  }

```

```

  // Last fallback: keep inside wrapper but at end

```

```

  var $wrap = $('.woocommerce-shipping-fields__field-wrapper').first();
  if ($wrap.length) $wrap.append($summary);
}

```

```

function showSummary(type){

```

```

  var vals = getFieldValues(type);
  var summaryHtml = vals.join('<br>') +
    '<br><button type="button" class="edit-address-btn" data-type="'+type+'" style="margin-top:8px;font-size:0.95em;display:inline-block;background:#ea5b0c;color:#fff;border:none;padding:8px 16px;border-radius:4px;cursor:pointer;">Edit Address</button>';

```

```

  var $sum = $('#'+type+'-address-summary').html(summaryHtml).show();

```

```

  $.each(fields[type], function(_, fieldId){ $('#'+fieldId+'_field').hide(); });
  $('#'+type+'-save-address-btn').remove();
}

```

```

if (type === 'shipping') {
  $sum.css('clear','both');
  placeShippingSummary();
}
}

```

```

function showFields(type){
$.each(fields[type], function(_, fieldId){ $('#'+fieldId+'_field').show(); });
$('#'+type+'-address-summary').hide();
$('#'+type+'-save-address-btn').remove();

```

```

var passWrapper = $('#'+type+'_passdata_field');
var resolvedId = resolvePostcodeFieldId(type);
var $wrapper = $();

```

```

if (passWrapper.length) $wrapper=passWrapper;
else if ($('#'+resolvedId+'_field').length) $wrapper=$('#'+resolvedId+'_field');
else {

```

```

  var $cand = $('[id^="'+type+'_passdata"][id$="_field"]').first();
  if ($cand.length) $wrapper=$cand;
}

```

```

if (!$wrapper.length){
  for (var i=fields[type].length-1;i>=0;i--){
    var cand=$('#'+fields[type][i]+'_field');
    if (cand.length){ $wrapper=cand; break; }
  }
}

```

```

var $btn = $('<button type="button" id="'+type+'-save-address-btn" class="save-address-btn"
data-type="'+type+'" style="margin-top:8px;font-size:1em;display:block;text-align:left;background:
#ea5b0c;color:#fff;border:none;padding:8px 16px;border-radius:4px;cursor:pointer;">Save
Address</button>');

```

```

if ($wrapper.length) $wrapper.after($btn);
else $('form.checkout').append($btn);
}

```

```

$(document).on('click','.edit-address-btn',function(){
  var type=$(this).data('type') || 'billing';
  if (confirm('Are you sure you want to edit your address? This may delay your order.))
showFields(type);
});

```

```

$(document).on('click','.save-address-btn',function(e){
  e.preventDefault();
  var type=$(this).data('type') || 'billing';
  if (isAddressPopulated(type)) {
    showSummary(type);
    requestUpdateCheckout(type);
  } else {
    alert('Please fill in required fields (city, postcode).');
  }
});

```

```
// When WC redraws checkout, re-render + re-place shipping summary
$(document.body).on('updated_checkout', function(){
  if (isAddressPopulated('billing')) showSummary('billing');
  if (isAddressPopulated('shipping')) showSummary('shipping');
  placeShippingSummary();
});

// On load (shipping fields may appear after toggling checkbox, so do a couple passes)
setTimeout(function(){
  if (isAddressPopulated('billing')) showSummary('billing');
  if (isAddressPopulated('shipping')) showSummary('shipping');
  placeShippingSummary();
}, 300);

setTimeout(placeShippingSummary, 1200);

$(document).on('change blur', '#billing_postcode, #shipping_postcode', function(){
  var type = this.id.indexOf('shipping')===0 ? 'shipping' : 'billing';
  requestUpdateCheckout(type);
});

$('#billing_postcode, #shipping_postcode').attr({
  autocomplete:'off', autocapitalize:'off', autocorrect:'off', spellcheck:'false'
});
});
```

continues...

Example EPOS Integration utils.php File

```
// Creating an assoc array for the mapped current order
$mappedOrder = array(

    "customer_rba_account_method" => "A",
    "customer_id" => '2' . $order["customer_id"],
    "customer_name" => $order["billing"]["first_name"] . " " . $order["billing"]["last_name"],

    "customer_address_style" => "RBA",

    "customer_company" => $order["billing"]["billing_company"],
    "customer_road" => $order["billing"]["address_1"],
    "customer_building" => $order["billing"]["data_billing_custom"],
    "customer_town" => $order["billing"]["city"],
    "customer_district" => $order["billing"]["billing_district"],
    "customer_postcode" => $order["billing"]["postcode"],
    "customer_county" => $order["billing"]["state"],
    // "customer_country" => $order["billing"]["country"],

    "customer_telephone" => $order["billing"]["billing_phonenew"],
    "customer_phone" => $order["billing"]["billing_phonenew"],
    "customer_email_address" => $order["billing"]["email"],
    "customer_mobile" => $order["billing"]["billing_mobile"],
    "customer_number" => $order["billing"]["billing_number"],

    "customer_vatable" => "Y",

    "delivery_name" => $order["shipping"]["first_name"] . " " . $order["shipping"]["last_name"],

    "delivery_company" => $order["shipping"]["shipping_company"],
    "delivery_road" => $order["shipping"]["address_1"],
    "delivery_building" => trim($order["shipping"]["shipping_company"] . ' ' . $order["shipping"]
["data_shipping_custom"]),
    "delivery_town" => $order["shipping"]["city"],

    "delivery_district" => $order["shipping"]["shipping_district"],
    "delivery_postcode" => $order["shipping"]["postcode"],
    "delivery_county" => $order["shipping"]["state"],
    //"delivery_country" => $order["shipping"]["country"],

    "delivery_telephone" => trim($order["shipping"]["shipping_mobile"] . ' ' . $order["shipping"]
["shipping_phonenew"]),
    "delivery_phone" => $order["shipping"]["shipping_phonenew"],
    "delivery_email" => $order["billing"]["email"],
    "delivery_number" => $order["shipping"]["shipping_number"],

    // "delivery_address1" => $order["shipping"]["address_1"],
    // "delivery_address2" => $order["shipping"]["address_2"],
    // "delivery_city" => $order["shipping"]["city"],
    // "delivery_postcode" => $order["shipping"]["postcode"],
    // "delivery_country" => $order["shipping"]["country"],
    "delivery_number" => $order["shipping"]["shipping_number"],

    "delivery_date" => $order["date_completed"],
```

EPOS Address Fields

Account

General **Sales** **Invoicing** **Audit** **Stock** **Service**

Main

Class Type

Order No? EEC/PVA? Batch Letters Reporting Code Natural

Salesman Code Under

Opt in?

By

Address / Contact Details

| | | | |
|-----------|---------------------------------|-----------------|----------------------|
| Name | <input type="text"/> | Phone 1 | <input type="text"/> |
| Firstname | <input type="text"/> | Phone 2 | <input type="text"/> |
| Name Ext. | <input type="text"/> | Mobile | <input type="text"/> |
| | | Fax | <input type="text"/> |
| Building | <input type="text"/> | Gen Contact | <input type="text"/> |
| Number | <input type="text"/> | Phone | <input type="text"/> |
| Road | <input type="text"/> | Acct Contact | <input type="text"/> |
| District | <input type="text"/> | Phone | <input type="text"/> |
| Town | <input type="text"/> | Email-General | <input type="text"/> |
| County | <input type="text"/> | Email-Invoice | <input type="text"/> |
| Postcode | <input type="text"/> | Email-Statement | <input type="text"/> |
| Country | <input type="text" value="GB"/> | User 1 | <input type="text"/> |
| | | User 2 | <input type="text"/> |

Staging site:

<https://staging.rentfreegascylinders.co.uk/>

Possible options:

1. Disable autofill across all browsers, desktop, tablet/mobile
2. Redesign the checkout process from scratch to stop autofill.