

1 Purpose of the Page:  
2 This page is a "Customer Informed Dates Listing" tool.  
3  
4 It's designed for internal use by logged-in users to review and update customer-informed  
delivery dates for product orders that are overdue, incomplete, or backordered.  
5  
6  What It Does – Step by Step:  
7 1. Initial Setup  
8 Includes getq.php to ensure user is logged in and gives access to key system variables/  
functions (e.g., safetext2, \$x, etc).  
9  
10 Reads hostcompanyid from the URL to determine which host company's data to show.  
11  
12 2. If hostcompanyid Is Not Set:  
13 Prompts the user to select a host company from a list filtered by their vui (user ID).  
14  
15 Presents host companies the user is linked to via pw\_companys.  
16  
17 3. If hostcompanyid Is Set:  
18 Displays the selected host company's branding and title area.  
19  
20 Fetches and caches delivery settings per stock\_ref into arrays \$hide and \$isdelivery2.  
21  
22 4. Main Data Table:  
23 Queries viamedquote\_orders and viamedquote\_orders\_details:  
24  
25 Filters only open and confirmed orders with unshipped items, that are not cancelled or  
complete.  
26  
27 Orders by duedate.  
28  
29 For each row:  
30  
31 It displays order and delivery details, including:  
32  
33 Order ID, Customer ID, Stock Reference, Due Date, Last Delivery Date (if set), Customer  
Informed Date (if set), etc.  
34  
35 Allows the user to set/update the "customer informed" date via a date picker and button.  
36  
37 Triggers a call to updatecustomerdateinformed.php using XMLHttpRequest.  
38  
39 Conditionally adds the order line to either:  
40  
41 \$mylist: if the user is the one who created the order.  
42  
43 \$restoflist: otherwise.  
44  
45 5. Extra Functions / AJAX Features:  
46 Each row includes buttons to:  
47  
48 Show/hide order details.  
49  
50 Get stock levels via /functions/search\_Stock\_Levels.php.  
51  
52 Get supplier order data via /functions/supplierordersstockreport.php.  
53  
54 These are loaded dynamically using plain JS and XMLHttpRequest.  
55  
56  What It's Used For:  
57 This script helps users track and manage customer expectations by:  
58  
59 Identifying overdue orders.  
60  
61 Seeing what delivery dates were communicated.  
62  
63

63 Letting staff update when customers have been informed of new expected delivery times.  
64  
65 Let me know your issues with it – happy to dig in and improve it with you.  
66  
67  
68  
69  
70  
71  
72 You said:  
73 yeah i have an order, where we have informed the customer of the date,  
74 but there are other items on the order we havent shipped as they will go together,  
75 but those obvioulsy now show, whne we dont want them to , (the date informaed in the  
table iwll be 0),  
76  
77 however we have another case where a customer has 2 different items onthe list, and we  
nede both to flag f we go past that date,  
78  
79 how would you do this?  
80  
81  
82 ChatGPT said:  
83 Ah, classic multi-item order logic clash – totally get where you're coming from.  
84  
85 You're facing two conflicting requirements:  
86  
87  Problem Summary  
88 Case A – Don't show:  
89  
90 A customer was informed about the delivery date (for the full order).  
91  
92 But only one line item has the updated "customer informed" date.  
93  
94 You don't want the other items (still pending shipment) to appear as "not informed" –  
even though they technically are.  
95  
96 Case B – Show:  
97  
98 A customer has two unrelated items (maybe different delivery dates).  
99  
100 You do want to be alerted if either one has not been informed, and the due date is coming  
up or passed.  
101  
102  Goal  
103 Avoid false positives for orders where the customer has been informed, but not all line  
items show it.  
104  
105 Still flag items that are overdue and not communicated.  
106  
107  Suggested Solution Logic  
108 We'll use a per-order aggregation approach to cover both use-cases:  
109  
110  Approach:  
111 Group by Order ID, then:  
112  
113 Check if any line has custinformdateon > 0.  
114  
115 If so, assume the customer has been informed about the order as a whole.  
116  
117 Display Logic Per Line:  
118  
119 For each line:  
120  
121 Show it only if:  
122  
123 Due date is overdue, and  
124  
125

125 No line in the same order has `custinformdateon > 0` → meaning nobody told the customer anything yet.

126

127  Bonus – Optional Enhancements:

128 If you want granular item-level tracking, you can also:

129

130 Introduce a new flag: `customer_fully_informed_on_order` (maybe stored/cached).

131

132 Or add a button to "mark all lines on order as informed", if only one date was set but it applies globally.

133

134  How It Would Look in Pseudocode

135 php

136 Copy

137 Edit

```
138 $informed_orders = array();  
139  
140 $result = sqlquery("SELECT orderid, MAX(custinformdateon) as maxinformed FROM  
    viamedquote_orders_details WHERE custinformdateon > 0 GROUP BY orderid");  
141 while ($row = $result->fetch_assoc()) {  
142     $informed_orders[$row['orderid']] = $row['maxinformed'];  
143 }  
144 Then later when looping through lines:  
145  
146 php  
147 Copy  
148 Edit  
149 $is_informed = isset($informed_orders[$orderid]);  
150  
151 if ($duedate < $duedatelim && $current_del_date < $nowis && !$is_informed &&  
    $isdelivery2[$stkref] == 0) {  
152     // show row  
153 }  
154 This handles your Case A cleanly and still allows Case B to show if neither item was  
    flagged.  
155  
156 Would you want me to retrofit this logic into your existing code and hand you the  
    modified section? Or want to talk through variations of it?
```