Goal: Adding a new check box to the search pages to show hide the salesperson column in the report results. As a note, the results come in four different flavors: normal (details and groups), exports to excel, super invoice line item search results normal, and super invoice line item search exports to excel. This may sound easy, but it has to show up in two places and has to criss cross between the details and groups, the normal and super mode, and back and forth between refining searches. Maintaining this setting on all the pages is an important part of this project.

I)Directions: From classic homepage go to assets and then invoice homepage. Once there, click on advanced invoice search. As a note, the advanced invoice search page has multiple forms. We are focusing on the line items section. This is the url: <https://www.adilas.biz/top_secret/secure/advanced_invoice_search.cfm> The line item search is the second section from the top. As a special note, there are multiple things you can select from the criteria to build the report and alter the number of columns that can be shown. Starting at the top of the filter line details, there is report type: details or groups.There are show extra fields (barcode number, weight, reference number) and show tax detail, which completely alter how many columns are shown.

This page has a buddy that also needs to be modified. This is the second url (super invoice line item search): <https://www.adilas.biz/top_secret/secure/super_invoice_line_search.cfm> You can get to this by scrolling to the bottom of the normal search and it will take you to the super search. There are more criteria that can be set for the report. It is important to note that users can go from basic to super and back so we need to maintain the settings in both directions.

II)Changes to the logic: a)Inside of code, you will need to start from advanced\_invoice\_search.cfm page. Put a note at the top (notifying modification). Create a new param for Form.include\_salesperson\_column.0 Once param'd come down and the next thing is: there are a number of places that this will need to be translated in urls. Do a search for advanced\_invoice\_line\_results.cfm. Wherever you find this, you will need to cascade a new variable. &showSalesperson=#FORM.include\_salesperson\_column#

Would need to add this same piece down lower as well.

b)Next folder would be advanced\_invoice\_search\_body.cfm.  Search for filter line details. This is close to where we need to be. Salesperson is variable. Need to maintain that. Right after the drop down for salesperson, need to add a break tag. Add the new check box here.

 <br />

<input type="checkbox" name="include\_salesperson\_column" value="1" <cfif FORM.include\_salesperson\_column IS 1>checked="checked"</cfif>> Show a column for the #LCase(variables.currentDefaultSalespersonName)#

c)Next go find instances of super\_invoice\_line\_search.cfm. Need to find the one that is long and pass that whole piece of code down here.&showSalesperson=#FORM.include\_salesperson\_column#.

d)Now we go to the results. Go to secure folder. advanced\_line\_results.cfm. There are some warnings. Make sure new paths or params get passed to this page. Need to find the section that deals with the 'i's." Form.include\_salesperson\_column# Look for 's'. Add a new url param. URL.show\_salesperson 0

e)Now we get to the switches, where we go between the form and the URL . <cfset FORM.include\_salesperson\_column = Trim(URL.showSalesperson)> This creates a toggle. Only one scope.

1)Next search for: getAdvancedInvoiceLineItems. Need to go to that method under search\_4.cfc. Need to modify the query. Search for this: name="qryGetInvoiceLineItems" Currently the payee table is only doing line items. We need it to connect to the main invoice table to translate the salesperson id's and names. This is called a self join.(need to alias the payee)Once you alias the payee (ie p1) then you cascade the change to all the places where it says payee and replace with p1. Also need to change the payees in the top portion to p1.

2) Next need to do another join. Need to search inv.salesperson\_id, Need to add these. Also need to add payee AS p2 to the from line.

3) A third join: inv.other\_salesperson.id. (p3)

, inv.salesperson\_id, inv.other\_salesperson\_id, CONCAT(p2.payee\_first\_name, ' ', p2.payee\_last\_name) AS salesperson\_name, CONCAT(p3.payee\_first\_name, ' ', p3.payee\_last\_name) AS other\_salesperson\_name

On the from line: payee AS p1, payee AS p2, payee AS p3,

Here are the joins:  AND p1.payee\_id = pili.vendor\_payee\_id  AND p2.payee\_id = inv.salesperson\_id  AND p3.payee\_id = inv.other\_salesperson\_id

4) Next search for useColumnCount variable. Copy, comment, paste  <cfif FORM.include\_salesperson\_column IS 1>

<cfset useColumnCount = (variables.useColumnCount + 1)>  </cfif>

This should go right after show\_customer. Instead of adding 2 columns, we are only adding 1 column.

5) Search for excel. Once you get to the excel portion, need to come down. Will need this: currentDefaultSalesperson. Create a column heading in excel(right after show customer piece) <cfif FORM.include\_salesperson\_column IS 1>  <td><div align="left"><strong>#variables.currentDefaultSalespersonName  #</strong></div></td>  </cfif>

6)Need to make sure you are within the loop. Loop over the results. Find the if statement for the show customers. Add a new comment and a new one and do something similar again:  <cfif FORM.include\_salesperson\_column IS 1>

<!--- put the main salesperson in --->

<cfset useSalespersonOutput = retInvoiceLines\_st.qryGetInvoiceLineItems.salesperson\_name>

<!--- next we need to check to see if there was a second salesperson. the id of 1 is for the dummy record. --->

<cfif retInvoiceLines\_st.qryGetInvoiceLineItems.other\_salesperson\_id GT 1>

<!--- add the second salesperson. --->

<cfset useSalespersonOutput = variables.useSalespersonOutput & ", " & retInvoiceLines\_st.qryGetInvoiceLineItems.other\_salesperson\_name>

</cfif>

<!--- actual output --->

<td><div align="left">#variables.useSalespersonOutput#</div></td>

</cfif>

III)Showing those in the web version (html) IV)Grouped and detail views (need to be able to switch between views) V)Passing data to the export