**OC**

Inside Order Create RPC

Avante

Web

**PC**

**PCF**

**PD**

**D-a**

**D-b**

**D-c**

**RT**

**RTO**

**RTX**

**D+a**

**D+b**

**D+c**

**D+d**

**R1**

**R2**

PREFERENCES:



* **UI (2)** Remove Price Discrepancy section and related code in My Account - Customer Experience area and test to make sure everything still works correctly in that page, fix any derived issues.
* **UI (1)**  Move Bool PriceDiscrepancy property (currently NoPriceDiscrepancy in UI code) from MyAccountCustomerExperienceModel to String PriceDiscrepancyAction in MyAccountShipPreferencesModel (keep it mapped to NOPRICEDDISCRP if it’s too much trouble to rename it in the DB) and from MyCustomerExperience and UpdateCustomerExperience methods to MyShippingPresets and UpdatePreferences methods. Modify Customer. PriceDiscrepancy make it String PriceDiscrepancyAction.
* **UI (2) -**\_MyShippingPresets.vbhtml Add New Price Discrepancies Section to My Account – Shipping All (under Attach Paperwork section as shown in spec. Add required text and 3 radio buttons linked to MyAccountShipPreferencesModel.PriceDiscrepancyAction, with 3 possible values:

0=I’m good with this (Policy).

1=No need to wait. Always process orders with slight discrepancies right away.

2=Wait for my approval. Never process these orders without my authorization.

Test that the value gets saved in the DB and displays back in screen.

~~WILL THE 3 RADIO BUTTONS BE UNSELECTED ON FIRST VISIT (PriceDiscrepancyAction=BLANK BECAUSE WE WON’T GRANDFATHER OLD SETTINGS)?~~ Value will come from DB preset to 0 if NULL.

* DB - Modify dictionary item GLBL.PRICE.DISCREP.PREF (attribute <172>) in the Customer Master User file (CUSTMST.USR).
	+ New definition for GLBL.PRICE.DISCREP.PREF – (0=I’m good with this (Policy). 1=No need to wait. Always process orders with slight discrepancies right away. 2=Wait for my approval. Never process these orders without my authorization). Update the description in SYSFILES (SYS3002 F8-Ext Desc)
	+ Will need to clear the existing data from this field (CUSTMST.USR<172>), as Sue explained that this was our way to “force” the customer to experience this new part of the system and its functionality. Ladd and Dawn want the user to get familiar with this section, so customers that currently have a preference set will have to redo their preference in the new format.
* DB - Modify LDLIB RPC$CUSTOMER\_PREFSAVE. If we use the existing param NOPRICEDISCRP then verify that no changes should be needed for this program as it will continue to update CUSTMST.USR<172>.
	+ What will be needed is to remove the default/assignment logic related to NOPRICEDDISCRP (should be 1 line of code setting “True” to 1 etc.) in LD.INCLUDES LD.CONSTANTS.

 

Windows Service:

UI (18) We’ll need to create a Windows Service to run continuously, just like the I’m Still Here process, this service will call an RPC$PROCESS\_PODISCRPORDERS which will return all orders that need processing and will loop through the results, cancelling, releasing, updating orders and sending notifications or acknowledgements as necessary.

DB – Create a new RPC$PROCESS\_PODISCRPORDERS that will be called by the UI and will return a recordset with all Orders that need PO Price Discrepancy processing.

* Select records versus the Master Order Header (MSTORDHDR.USR) file.
* The select statement will choose PO.PRICE.DISCRP.DISPOSITION.TYPE = “” ;

PO.PRICE.DISCR.TYPE = 1 or 2; Operator = New Operator Code, Cutoff Time and Date <= Current Date and Time.

* Will need a CASE statement to identify type of record processing…

Type=1 and Status=2 then Slight Discrepancy, 2 hours, release order UI Send order ack.

Type=1 and Status=24 then Slight Discrepancy, 24 hours. UI Send appropriate message and update date & time with an additional 24 hours and change status=48.

Type=1 and Status=48 then Slight Discrepancy, 48 hours. Cancel order. UI Send appropriate message. Update disposition fields.

Type=2 and Status=24 then Significant Discrepancy, 24 hours. UI Send appropriate message and update date & time with an additional 24 hours and change status=48.

Type=2 and Status=48 then Significant Discrepancy, 48 hours. Cancel order. UI Send appropriate message. Update disposition fields.

Programmer Note: If slight or significant, and greater than 48 hours then cancel the entire order by running the new cancel program created in RPC$ORDER\_CANCEL. We do not want to call the RPC as we do not want to call one RPC from another.

PCF - Price Check Flagging.

In order to support the checkbox and textbox on each ShoppingCartItem’s CE Drawer, we will need to:

* UI (2) In ShoppingCartItem, CheckoutSingleBundleModel and CheckoutSummaryStepModel, Add Boolean property isPurchaseOrderPriceDiscrepancy mapped to PO.PRICE.DISCRP and String property PurchaseOrderPrice mapped to PO.PRICE.
* DB - Create 2 new shopping cart fields (SHOPPINGCART.USR)
	+ PO.PRICE.DISCRP – Multi value field associated with F1 – ITEM.NBRS. (0=No ; 1=Yes)
	+ PO.PRICE – Multi value field associated with F1 – ITEM.NBRS (Numeric MD2)
	+ Create an entry in SYSFILES for both new fields.
* DB - Modify LD.INCLUDES SHOPPINGCARTITEM.READ – Being called from RPC$SHOPPINGCARTITEM\_GETBYFILTER and GETBYID.
	+ Add logic to assign the two new fields.
* UI (0.5) - Add isPurchaseOrderPriceDiscrepancy check box to slider.
* UI (0.5) -Add PurchaseOrderPrice text box to slider.
* UI (2.5) -Add event to textbox to trigger calling RPC$SHOPPINGCARTITEM\_SAVE and test that it saves and displays correctly.
* DB - Modify RPC$SHOPPINGCARTITEM\_SAVE
	+ If the Price Discrepancy box is checked, then write a “1” to the new PO.PRICE.DISCRP field, and the Price of the P.O. to the new PO.PRICE field. If the box is null then write out a “0” and a NULL respectively.
* UI (0.5) - Additionally, In **CHECKOUT** CheckoutShippingStepModel.InitiateStep transfer isPurchaseOrderPriceDiscrepancy and PurchaseOrderPrice fields’ data to every singleItem in bundle.Items.

SUE: HOW WILL PRICE DISCREPANCIES AFFECT THE SHIPPING & PACKAGING SCREEN IF AT ALL?

~~WHAT ARE WE DOING WITH NS ORDERS?~~ Not in the web, don’t worry about it for now.

SUE NEED A DESIGN FOR CONFIRM EXIT FROM EMPTY TEXTBOX AS NO PRICE DISCREPANCY. Yes / No dialog box, uncheck IsPurchaseOrderPriceDiscrepancy check box if they click Yes.

PD - Price Discrepancy (inside Order Create).

When the user clicks on the Create Order button, the system will figure it out whether there are any price discrepancies in the order(s) and what type they are, and it will either release the order or create requests for the child orders that have a price discrepancy.

DB – Create a new operator code in SYSTBL “OER” to represent Price Discrepancy operator.

DB – Create a new request type *nnn* in SYSTBL “RTC.LAD”. Assign Fields 1, 2 and 3 equal null and Field 4 Description = Price Discrepancy.

DB - Create a new System Constant for Significant Discrepancy. Add attribute <2> to the existing SYSCON WEB.SETTINGS.LAD and assign 6.00 as the cutoff percent. Currently the SYSCON has 1 attribute which is storing the Max Cart Age (used for selecting).

DB - Create 1 new field in MSTRORDDET.USR

* PO.PRICE. Numeric MD2.

DB Create 9 new fields in MSTRORDHDR.USR

* PO.PRICE.DISCRP.TYPE (1=Slight Discrepancy ; 2=Significant Discrepancy)
* PO.PRICE.DISCRP.DATE - Date when request type will expire.
* PO.PRICE.DISCRP.TIME - Time when request type will expire.
* PO.PRICE.DISCRP.STATUS – Status of the discrepancy, the number of hours the (cutoff) date and time represents. (2, 24 OR 48)
* PO.PRICE.DISCRP.DISPOSITION.TYPE – Most likely create a new code table definition (Process=TABLE.DEFN File=IIDEFN). Create codes that represent the disposition of the discrepancy. (i.e. Released ; Cancelled).
* PO.PRICE.DISCRP.DISPOSITION.DATE
* PO.PRICE.DISCRP.DISPOSITION.TIME
* PO.PRICE.DISCRP.DISPOSITION.OPERATOR (CE Code, WEB, *New Price Discrepancy Operator Code*)
* PO.PRICE.DISCRP.LOGIN.VERIFIER (Date+Time+Random#) Enables login from email link, acts as a password as far as credentials along with OwnerID.

For calculating the PO.PRICE.DISCRP.DATE and TIME, Use the SHIPPING calendar in SHOPCALMST for valid dates. For Start (8 AM) and End (5 PM) times, add two new attributes to the SYSCON SHIP.CUTOFF.LAD. At this point all times/logic will be based on Eastern Standard Time (EST).

DB - Modify logic when creating orders.

* RPC$SHOPPINGCART\_CREATEORDER. In the INIT.VARIABLES section. Update the variable MVARR with the positions of the 2 new attributes that were created in the Shopping Cart. Let the current logic flow per normal (including bundling). If MOD item then assign a new USERDATA(10)<?> field with the PO Price (for an example see how USERDATA(10)<85> ITEM.NO gets assigned).
* Modify RPC$CREATE\_STOCKORDER. Need to assign the PO Price to either a USERDATA field or RECORD
* Modify SYSS0123.1 – When the Master Order Detail file (MSTRORDDET.USR) record is created. For MOD items we will write out the entered PO Price (USERDATA(10)<?>) for each individual order (order:suffix) with a discrepancy. For Stock items we need to load USERDATA(10)<?> with the appropriate PO Price. When the Master Order Header file (MSTRORDHDR.USR) record is created, calculate if it is a slight discrepancy or significant discrepancy (using the system constant WEB.SETTINGS.LAD<2>) and write out the PO Price Discrepancy Type. If the Master Order Header record already exists then calculate discrepancy and update if higher than existing code. (So always update the record with the highest PO Price Discrepancy code). Programmer Note: It may be easier to do the calculation at the time of updating the detail record and just track the highest discrepancy code.

UI - If a discrepancy exists at the time of order creation, instead of displaying the standard message with the Order# and sending the normal order acknowledgement, display popup message based on Slight or Significant Discrepancy (NEED DESIGN) and send a discrepancy notification email with a “high importance” indicator to the contact that placed the order and to the CC’s.

* UI (7) Modify Order.vb and any models that may need adding the following properties:
	+ String PO\_PriceDiscrepancyType 🡪 PO.PRICE.DISCRP.TYPE (1=Slight Discrepancy ; 2=Significant Discrepancy)
	+ String PO\_PriceDiscrepancyDate 🡪 PO.PRICE.DISCRP.DATE - Date when request type will expire.
	+ String PO\_PriceDiscrepancyTime 🡪 PO.PRICE.DISCRP.TIME - Time when request type will expire.
	+ String PO\_PriceDiscrepancyStatus 🡪 PO.PRICE.DISCRP.STATUS - STATUS OF THE DISCREPANCY, NUMBER OF HOURS THE (CUTOFF) DATE AND TIME REPRESENTS. (2, 24 OR 48)
	+ String PO\_PriceDiscrepancyDispositionType 🡪 PO.PRICE.DISCRP.DISPOSITION.TYPE
	+ String PO\_PriceDiscrepancyDispositionDate 🡪 PO.PRICE.DISCRP.DISPOSITION.DATE
	+ String PO\_PriceDiscrepancyDispositionTime 🡪 PO.PRICE.DISCRP.DISPOSITION.TIME
	+ String PO\_PriceDiscrepancyDispositionOperator 🡪 PO.PRICE.DISCRP.DISPOSITION.OPERATOR (CE, WEB, TBD)
* UI (6) Modify \_OrderConfirmation.vbhtml POPUP with new wording notifying user of the discrepancy (NEED DESIGN).
* UI (9) Build \_PO\_PriceDiscrepancyNotification.vbhtml partial that will serve as the email body (email 1 if slight, 2a if significant or 3a if slight with preference action = 3 “Wait for my approval”). Add logic that will dynamically build the content for each case.
* UI (6) Add PO\_PriceDiscrepancyNotification method that will create the email object (or fax) and send it. Call it from within CreateOrder method right below the With ReturnModel…End With block, pass ReturnModel as param. New method will figure it out what type of notification (1,2a,2b,3a or 3b) is needed based on the (ReturnModel)CheckoutModel.CheckoutSummaryStep.Order’s properties : PO\_PriceDiscrepancyType, PO\_PriceDiscrepancyStatus and Customer. PriceDiscrepancyAction and will use \_PO\_PriceDiscrepancyNotification.vbhtml (see previous bullet) to build the email. Use RPC$SHIPMENTITEMS\_GETSUMARY to get ahold of the items that have discrepancies and feed that to the partial.

Note: Order records will still be created but will be placed in a special type of OnHold status due to PO price discrepancy (PO.PRICE.DISCRP.TYPE > 0).

* DB – Updates when WEB session completes
	+ Modify process after screen accept (SOPP9116.5) -When the WEB session completes, a multi-valued (by transaction) TXX record is created. For all 030 SALES ORDER ENTRY transactions, check if the order has a Price Discrepancy. If so check the customer’s preferences. If discrepancy is still applicable then create a new request (in the REQUESTS.USR file) with all the same values from the previous 030 entry with the exception of Operator Code (New Operator Code) and Request Type (New Price Discr Code). Also verify that the Req Status equals 0 (zero). Based on the discrepancy type, calculate the cutoff date and time and update the Master Header record with the date, time and type (2, 24 or 48 hours – at this point would either be 2 or 24). If discrepancy is not applicable then update the new Disposition fields in the Master Header record.
* DB - For the Open C/E Requests process (SOP9116). Modify the Request Type prompt to not display the new Price Discrepancy type during F3 Select logic. If that appears too difficult then continue displaying it, but if it is chosen then give the user invalid choice error message.
* DB – We probably should review the AUTOPICK logic, may want to change select logic to be based off of master order record discrepancy fields and not only based in open requests since we have the 15min? issue.

~~HOW DOES A PRICE DISCREPANCY GO FROM A SHOPPINGCARTITEM INTO A BUNDLEITEM?~~ In CheckoutShippingStepModel.InitiateStep.

RTO (Request type Online), RTX (Request Type Call, Email, Fax).

At this point the Order has been placed in Pending status by the Create Order process, in order to resolve this status we need to implement an “Order Summary” screen so a user can get to it from the email or logging in and clicking on the Pending link for that order (where Summary would normally be) and release the order or cancel it through that screen.

* Through Email:
	+ UI (6) The email body with have a “here” link that will contain the login info necessary to redirect the user to our website and hit a new login method if the request string contains “OrderSummary” in it, once logged in, load the “Order Summary” partial.
	The link will look similar to this: [https://laddawn.com/Account/login/?contactNbr=XXXXX&guid= GUID&redirect=OrderSummary& orderdetail=XXXXXX](https://laddawn.com/Account/login/?contactNbr=XXXXX&guid=%20GUID&redirect=OrderSummary&%20orderdetail=XXXXXX), where GUID will be a unique identifier for that given session and order detail is the order ID (PO.PRICE.DISCRP.LOGIN.VERIFIER) generated in Avante at Order Creation and stored in the master order record which will return in the Order object.

SUE: WHO WILL BE GETTING THIS EMAIL, JUST THE CONTACT OR ALL THE ADDRESSES THAT THE USER PUTS ON THE CC’s FIELD? AND WHO GETS THE ACKNOWLEDGEMENT LATER ON? WE’RE NOT KEEPING TRACK OF THE CC’s. DO WE NEED TO WRITE TO THE MASTER ORDER ALL THE CC’s?

* + UI (6) Modify AccountController.Login() to redirect to: OrderHistoryController.ShowPriceDiscrepancySummary method which will in turn load the OrderSummary partial into div #tabs with the info of the order.

SUE: What about Fax Only customers? How are those going to be handled?.

* Through Pending link in Order History grid.
	+ UI (2) Once the Order has been flagged with price discrepancy it will show in the Order History grid with Pending instead of Summary, so add logic in \_OrderHistoryResultsByOrderName.vbhtml to present “Pending” link in grid when appropriate and disable Reprice and Track links as well. Base these decisions on OrderStatus (ORDER.STATUS) and PO.PRICE.DISCRP.TYPE.
	+ UI (3.5) Attach event to Pending link to open \_OrderSummary.vbhtml in “full content” mode (not inside a dialog) as per spec. Note that this will be the default process for a CE rep trying to take Cancel or Release action on the order.
* UI (22) Modify \_OrderSummary.vbhtml to behave differently (or new partial?) based on whether the order is in Price Discrepancy Pending status or not, if in Pending status then:
	+ Add title “~~Review your order~~” -> “Order Summary”.
	+ ~~Add hardcoded cart step graphic~~. No longer needed.
	+ Add “Copy order summary to” label and email text box.
		- In $(document).ready Apply laddawnAutoComplete pluggin to email text box.
	+ Add “Order Number” label and field for Master order (i.e. “11767A,B,C”), parse and concatenate to achieve required format.
	+ Add “Order Number” label and field for each Child order (i.e. “11767A”).

DB - How am I going to get the A,B,C child orders? New RPC$ORDER\_GETCHILDREN or new field in Order object? Or does ShipmentItem.ID contains the child order number?

Ask Sue how you should show one bundle that becomes two orders (stock riding with mod) or two bundles that become one order (split shipments).

* + Change column header “Ordered” to “Qty” accordingly.
	+ Add Price and Total columns to bundling grids, including PO Discrepancy data in red.
		- Add (String)UnitPrice, (String) PurchaseOrderPrice and (String)TotalPrice to ShipmentItem and map to UNIT.PRICE, PO.PRICE and PRICE. Also add to CheckoutBundleSingleItemModel.
		- DB – Add UNIT.PRICE, PO.PRICE and PRICE (Total Price) to RPC$SHIPMENTITEM\_GETBYFILTER AND RPC$SHIPMENT\_GETSUMMARY.
		- Have ShipmentItemService.GetOrderSummaryItems return UNIT.PRICE, PO.PRICE and PRICE (Total Price).
		- Add (String)TotalPrice to CheckoutSingleBundleModel.
		- Add up the total per bundle into CheckoutSingleBundleModel. TotalPrice at OrderHistoryController. GetBundleModel.
		- Using UnitPrice, PurchaseOrderPrice and TotalPrice add columns to grid. Set PurchaseOrderPrice to red text.
	+ Re-enable Labeling section, show if called from OrderHistory only.
		- In the OrderSummaryInfo method, make sure viewModel. ShippingLabel gets all the necessary properties to correctly display the label, take a look at CheckoutLabelingStepModel.InitiateStep for this.
		- Tweak code in partial as necessary to make the label work here.
		- Make sure UI gets everything that’s needed to paint the label in ShipmentItemService.GetOrderSummaryItems.
	+ Add “Optional Note to your Customer” section. ~~WHERE WILL THIS NOTE BE TYPED IN?~~ AT THE BOTTOM OF THE CHECKOUT SUMMARY SCREEN.
	+ Add “Approve/Cancel Order” section to the right column of the screen.
	+ Attach functionality to Approve and Cancel buttons:
* Create a jquery click event for the Approve button that will call the ApprovePendingOrder method in the OrderHistoryController.
* Create a jquery click event for the Cancel button that will call the ShowValidationMessageDialog().
* Modify and use ShowValidationMessageDialog() in dialogs.js (or clone it into this partial) to display confirmation popup.
* Create a new Action.Type = “3” and set options.title=”Are you sure?”
* Copy function GetValidationActions() from \_Shop.vbhtml and add case for Action.Type=”3” with Yes/No buttons.
* Attach click event to “Yes” button to call the CancelPendingOrder method in the OrderHistoryController.
* Attach click event to “No” button to close confirmation dialog.
* DB - Create RPCs to release the Order or cancel it based on the action of the user.
	+ For Cancel: New RPC$ ORDER\_CANCEL. (Params: MasterOrderNumber). For each child order, mimic Sales Order Entry F4-Cancel. Make sure it closes the duplicate price discrepancy requests by setting REQUESTS.USR<5> = 1. Will also need to update the disposition fields in the Master Header record. Return whole Order object.
	+ For Release: New RPC$ORDER\_RELEASE. (Params: MasterOrderNumber). Make sure it closes the duplicate price discrepancy requests by setting REQUESTS.USR<5> = 1. Will also need to update the disposition fields in the Master Header record. Return whole Order object.

OC – Order Cancelled.

When an order gets cancelled either by the customer, the CE rep or Customer Preferences, we will display this fact in the website in following manner:

* UI (3) Order will show in Order History grid with additional text in Order Name column “Order 11713A,B,C Cancelled” and Track link will be disabled. Note that when the order has been cancelled by PO Price Discrepancy, the whole master order will be cancelled whilst in other scenarios the order can be partially cancelled. We will display both scenarios but the second one will have the track link enabled and showing the popup per the spec.
* UI (0.5) Modify \_OrderTrackingInfo.vbhtml to show “CANCELLED” in the tracking column when appropriate.
* UI (0.5) Modify \_OrderInfoTooltip.vbhtml, append “- CANCELLED” to the Description column when appropriate.
* UI (0.5) Modify \_OrderInfo.vbhtml, append “- CANCELLED” to the Description column when appropriate.
* UI (0.5) Modify \_OrderSummary.vbhtml, append “(Order Cancelled)” to order name per spec.

We will also send notifications to the customer:

* UI (5) if cancelled from the \_OrderSummary.vbhtml screen, tweak and use the same partial as the body of the email.
* UI (2.5) If cancelled from the windows service: Call PO\_PriceDiscrepancyCancelOrder method, use \_OrderSummary.vbhtml as the body of the email.

~~WHAT HAPPENS WHEN SIGNIFICANT DISCREPANCY AND “WAIT FOR MY APPROVAL”?~~ Our rules.

~~HOW TO FORWARD EMAIL TO CE QUEUE?~~ Email CE@LADDAWN.COM and that puts it in the queue.

SUE: IN “DO NOT CALL MESSAGE” SECTION IN SPEC, IS THIS FOR REVIEW ORDER SUMMARY SCREEN OR CHECKOUT SUMMARY SCREEN?

SUE / OWEN: WHY ARE THE CALL MESSAGES BY BUNDLE. MESSAGE SHOULD APPLY TO MASTER ORDER NOT TO A BUNDLE

~~NEED CLARIFICATION ON:~~

* To create the order, CE will have started a new 005 web request as they do today, and create the order(s) with line(s) flagged with price discrepancies.  If they look in the purple CE Request Tab, they will see the order(s) that were created.  These orders will NOT be flagged as On Hold.~~RUN OUR THOUGHTS BY JUDY AND JOHN TO MAKE SURE WE’RE GOING IN THE RIGHT DIRECTION HERE.~~ If there is another reason that any of these orders should be on hold, CE can set the onhold flag and route them as usual.
* When we create the CE requests when they are back in Avante, we will continue to create a request for each order created.  If they manually flagged the order as on hold, the request will be placed on hold and remain in their on hold queue.
* In addition to the normal requests that we spin off: if there is at least one price discrepancy, we will create a second request for every order of that master order, with a new request type of ‘price discrepancy’.  The price discrepancy requests will be assigned to a special operator (TBD) so that they don’t show in the ce rep’s on hold queue.
* ~~What about NS orders?~~ Handled in Avante, don’t worry about it for now.