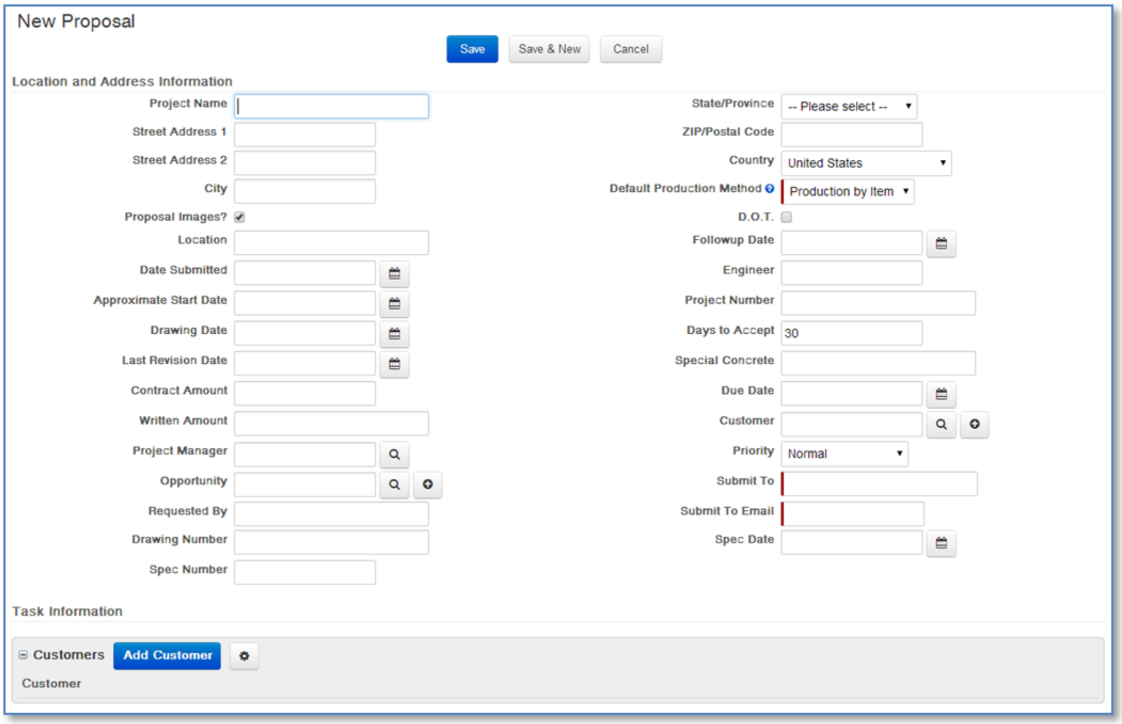
Proposal Screens

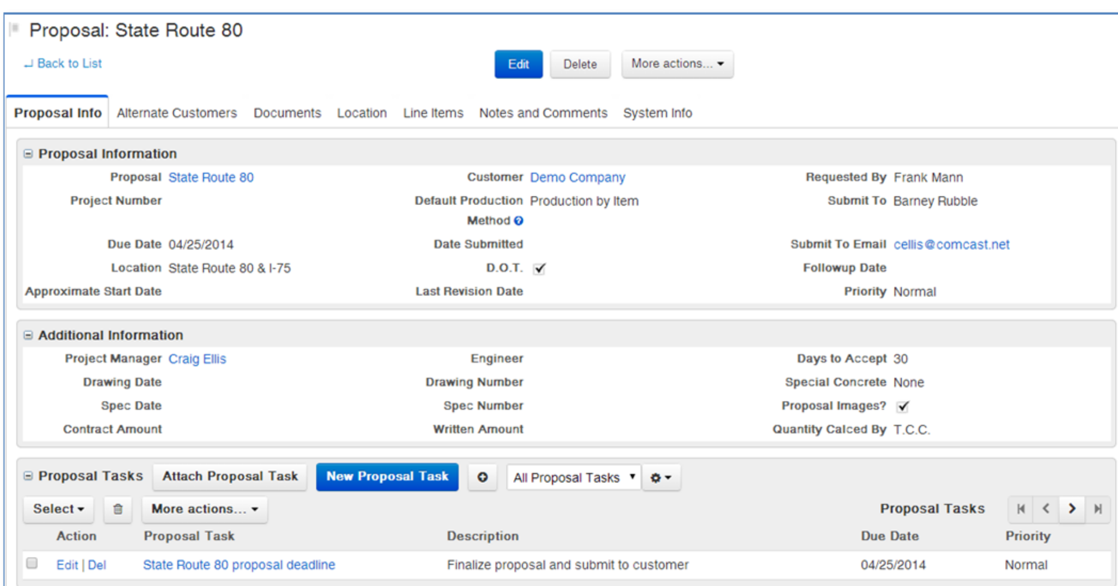
Enhancing the Proposal user interfaces

The proposal is one of the most central objects and events within this system. We will revisit it frequently during the development process. Prior to the preparation of this document, I took the liberty of doing a first pass of organizing the screens. They currently look like this:

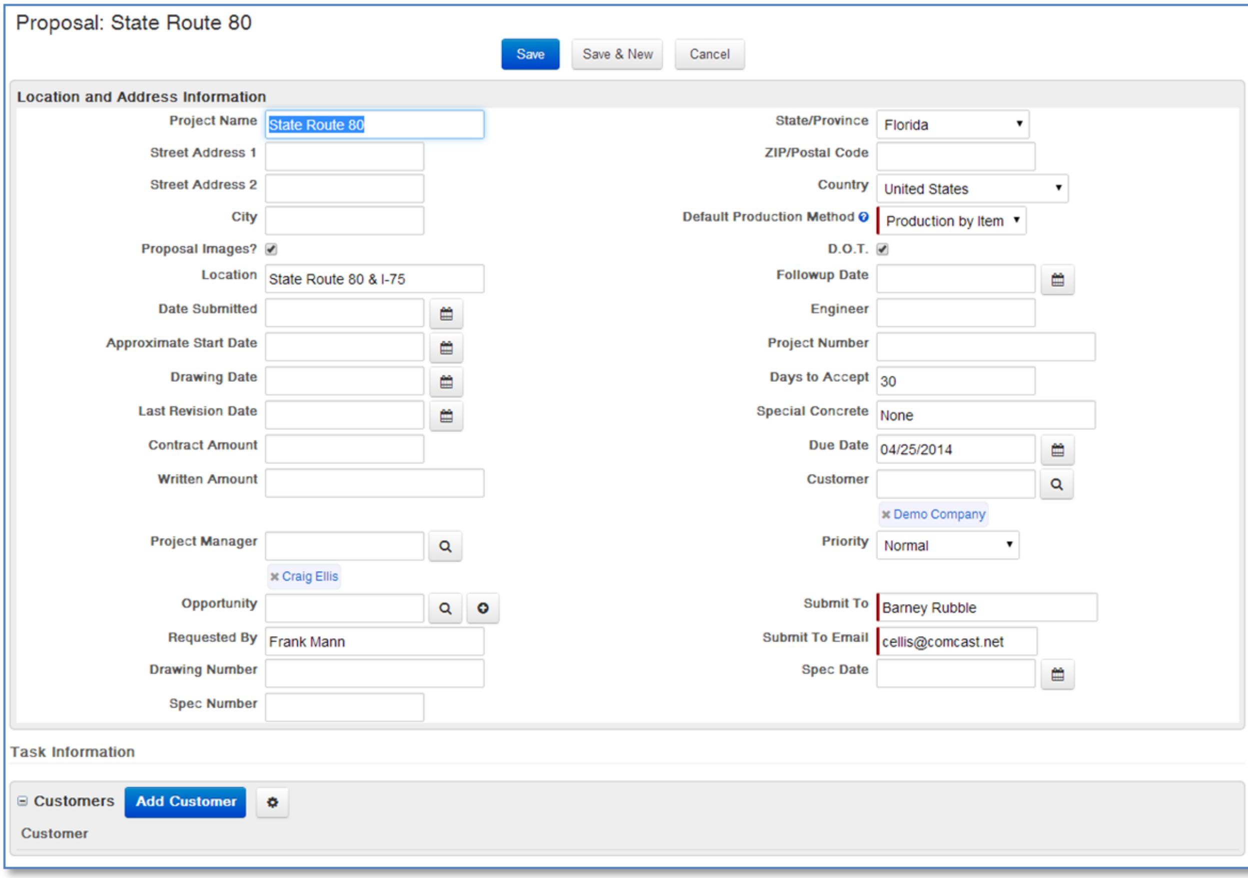
New



View

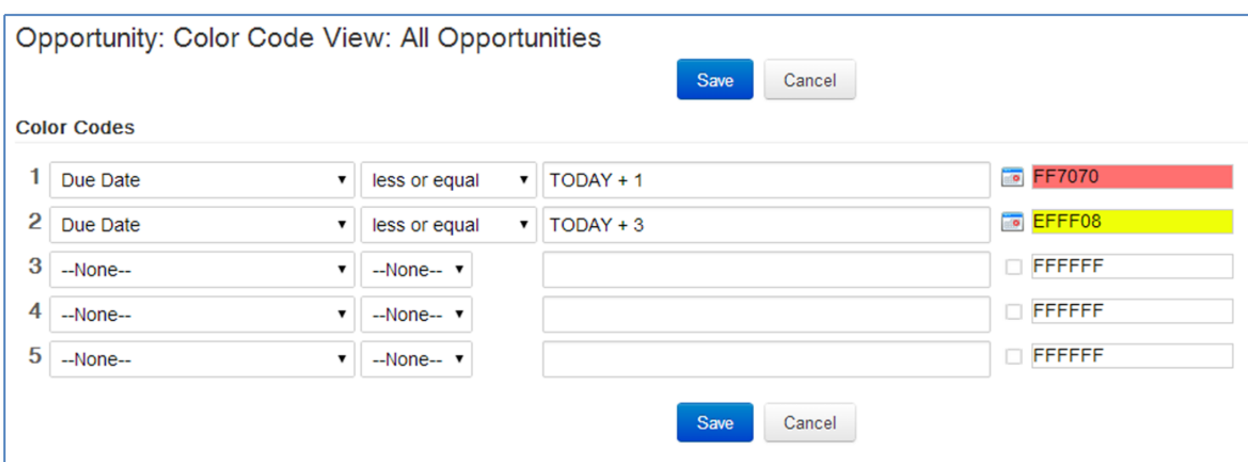


Edit



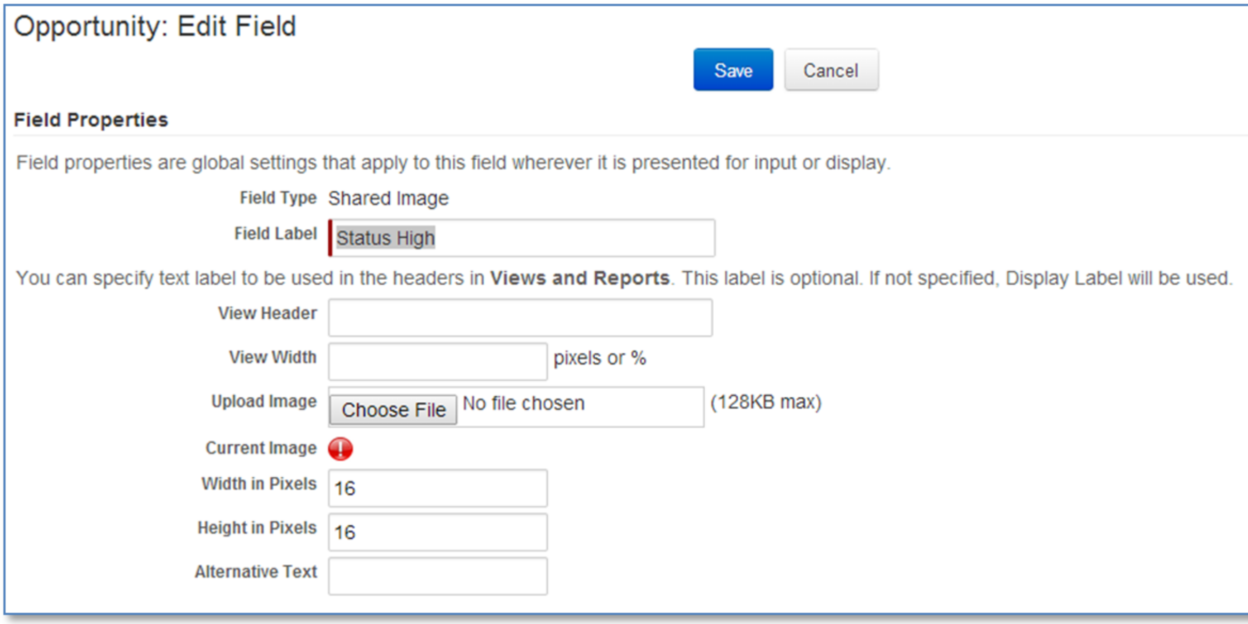
Clearly the New and Edit screens need to be improved. I strongly urge you to master one screen, for us, that'll be the "View" screen and when you have settled on what you like, then track back and update the other two screens.

Before we get to those screens, I want to duplicate what we did on the views for opportunities, specifically, I like the row coloring and the status flags. We want to simply copy/paste the Edit Row Color settings we specified in the Opportunity view. I find it useful to have a notepad open to copy/paste both the color values and the formulas, so when I replicate for Proposals, I don't have to retype values.



Now that we have the row colors, we'll need to duplicate the formula field from opportunities, as well as the two shared image fields. In the opportunity object, we created 3 shared image fields, but only ended up using two; one for high and one for low. We will do the same for the Proposal Object.

A quick look back to how that's done:



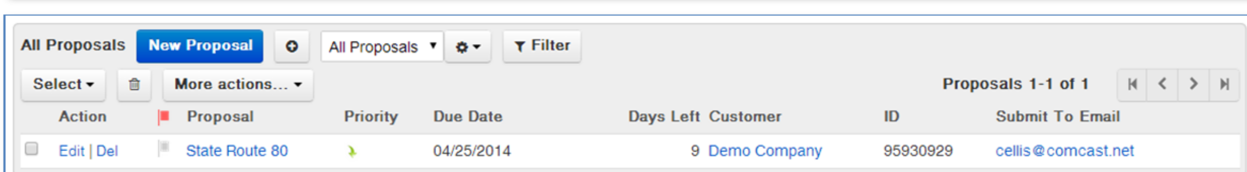
"Status" was not really representative, as we're indicating "Priority", so we'll use that for the Proposal Object. Create the Priority High and Priority Low shared image fields.

Edit | Del	Priority High	Shared Image	priority_high	☐
Edit | Del	Priority Low	Shared Image	priority_low	☐

Now we want to create the formula field to display the icons. Create a formula field "Visual Priority" and paste the formula from the opportunity. Grab the correct integration tokens from the helper and update the formula to support the proposal. Be SURE to set the return type to string. I made this mistake and it took a few minutes to figure out why my icons weren't showing up. Adjust the view to reflect the order desired. I put priority right after the Proposal field.

Formula return type: **String**

```
if ("{!proposal_priority#code}" == "NORMAL"){
  return "";
}
else if ("{!proposal_priority#code}}" === "HIGH"){
  return "{!ccProposal.priority_high#html}";
}
else if ("{!proposal_priority#code}" === "LOW"){
  return "{!ccProposal.priority_low#html}";
}
```



Alternate Customers

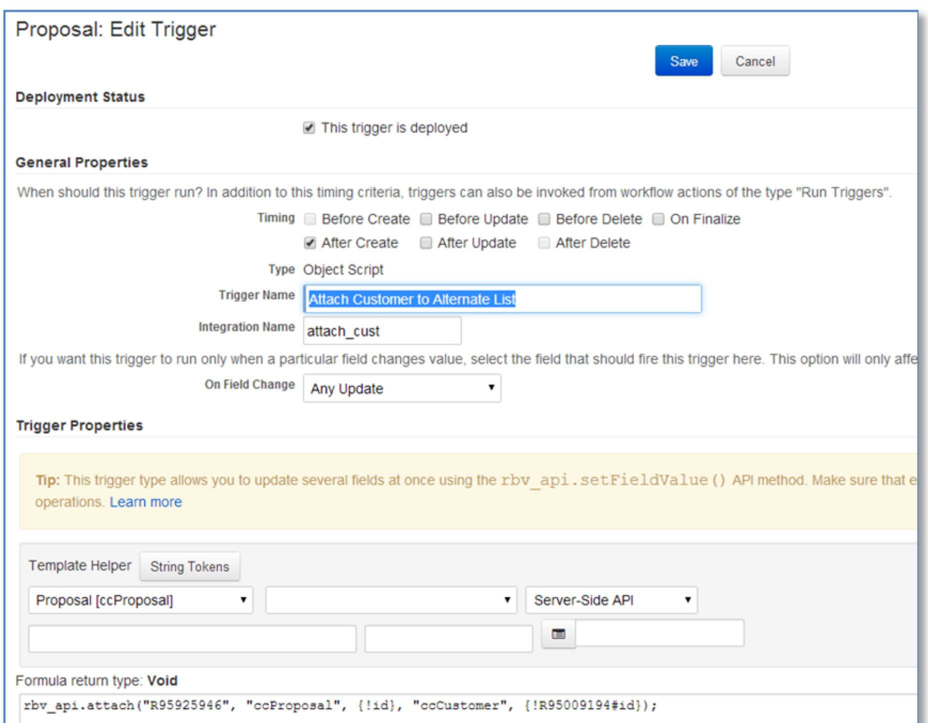
By Tincher Concrete's requirement, not only is a proposal assigned to a customer, but multiple customers bid on the same contract and ask for the same proposal from Tincher. As such, we want to create a new relationship to customers that will allow us to accomplish this. Under the Relationships section of the Proposal Object I created a new relationship "One Proposal to Many Customers".

Relationships New Relationship

Action	Relationship Name ▲	Object	Cardinality	Integration Name	Last Updated	Last Updated By
Edit | Del | Convert	Customer	Customer	One Proposal to Many Customers	R95925946	04/16/2014 09:29 AM	Craig Ellis
Edit | Del | Convert	Customer	Customer	Many Proposals to One Customer	R95009194	04/03/2014 08:55 AM	Craig Ellis
Edit | Del | Convert	Opportunity	Opportunity	One Proposal to One Opportunity	R95013586	04/03/2014 09:38 AM	Craig Ellis
Edit | Del | Convert	Opportunity Document	Project Document	One Proposal to Many Project Documents	R95794029	04/16/2014 10:10 AM	Craig Ellis
Edit | Del | Convert	Proposal Task	Proposal Task	One Proposal to Many Proposal Tasks	R95911408	04/16/2014 10:11 AM	Craig Ellis
Edit | Del | Convert | Permissions	User	User	Many Proposals to Many Users	assignedTo4	04/02/2014 11:33 AM	Craig Ellis
Edit | Del | Convert | Permissions	User	User	Many Proposals to One User	R95008842	04/03/2014 08:45 AM	Craig Ellis

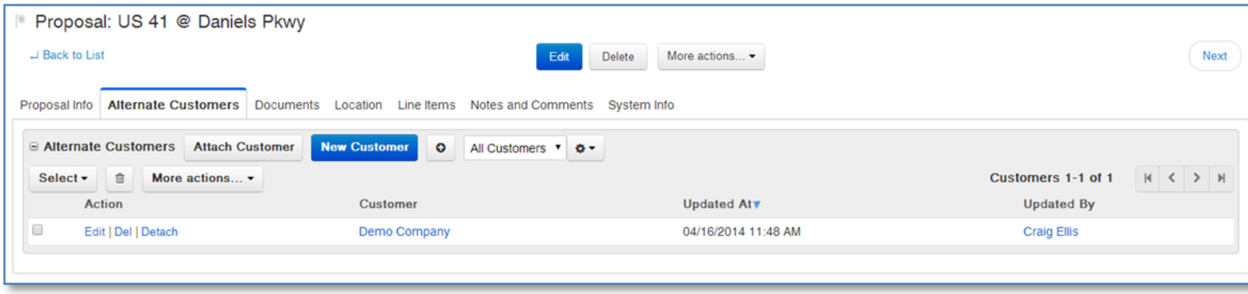
Now I will need to go back to my trigger and reflect this new relationship, because by default, the customer assigned to this proposal ALSO will need to appear in the list of alternate customers. Copy the integration name for the new relationship (R95925946) in this example to the clipboard.

While there is a type of trigger called "Attach Related Record", that doesn't really apply in this case, so we'll have to do it with an Object Script Trigger. The trigger is quite easy. Create the trigger below and test. Below the screen shot is the "layman" explaination of the api call.



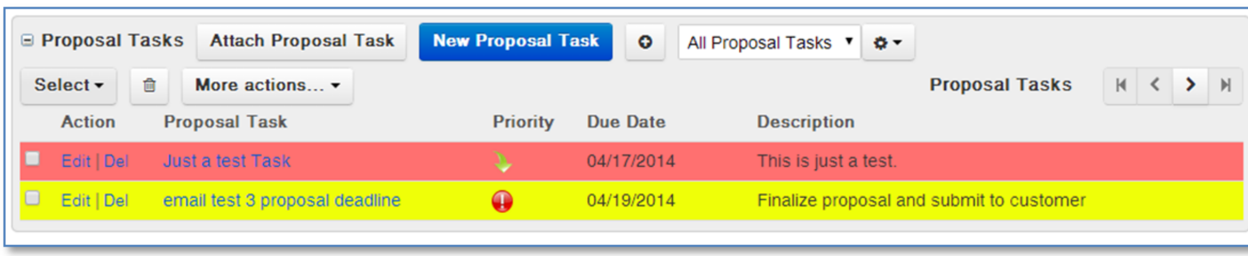
Rbv_api.attach("new relationship","proposal integration name",current proposal id,"customer integration name",id of the single customer attached to proposal).

Create a new opportunity and test it. If all works correctly, the proposal will not only have the customer attached on the main screen, but you should now see it in the Alternate Customers tab.

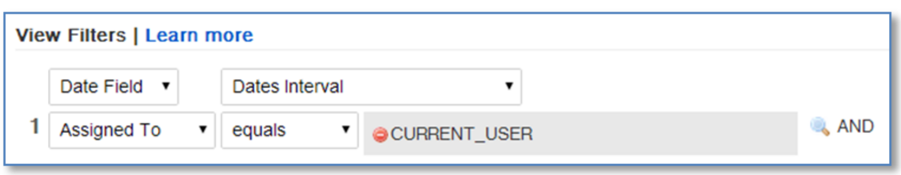


Proposal Views

Now we want to dress up the grid for Proposal Tasks to reflect the same row coloring and priority icons as we have done twice before; once on the Opportunity view and once on the Proposal view. Duplicate this logic on the Proposal Task view.



Once you are happy with the grid view, clone the view and call it “My Proposal Tasks”. Make sure you update the filter to include the “Assigned To” equals “CURRENT_USER”.



This will allow other users to create tasks for this proposal, but allow the current user to view only what applies to them. Remember, you will have to duplicate the row color logic.

Google Maps

Since we applied the “Location” attribute to our proposal, we are able to display a Google Map. I’ve added a tab to the Proposal View for Location. I simply added one section for the address information and a second section to hold the Google Map component.