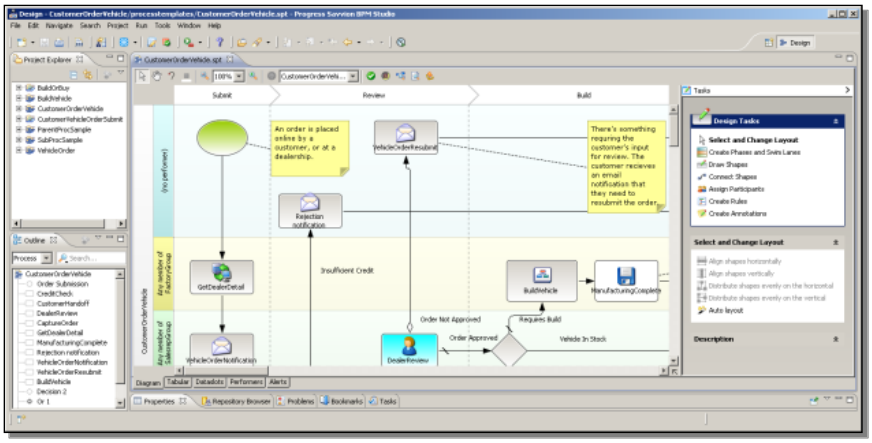
SAVVION MANAGEMENT OVERVIEW

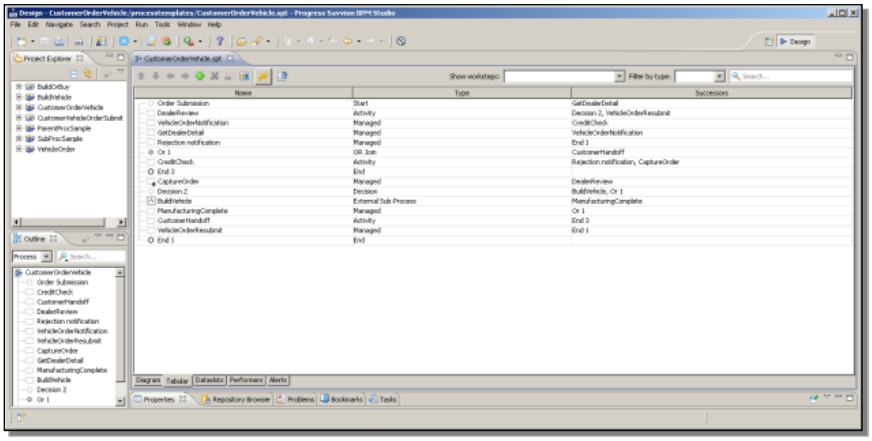
John Sadd
Fellow and OpenEdge Evangelist
Document Version 1.0
August 2011

DISCLAIMER

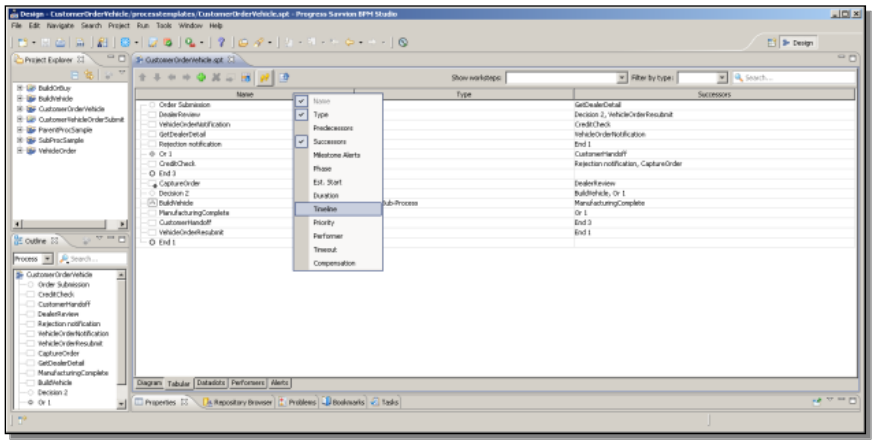
In the overview presentation on OpenEdge Business Process Management, I showed you just a bit of an actual BPM model for an AutoEdge sample application module called **The Factory**. And I showed you how that model design translates directly into executable behavior at runtime. In this session I'll give you just a taste of some of the application management features of Progress Savvion, and how they help you to analyze your application when you're designing it, and to identify and fix problems at runtime. Below is the sample application for ordering a car that I showed you. You're looking at it in BPM Studio, the Savvion design tool, in the diagram view, which shows all the steps in a graphical form:



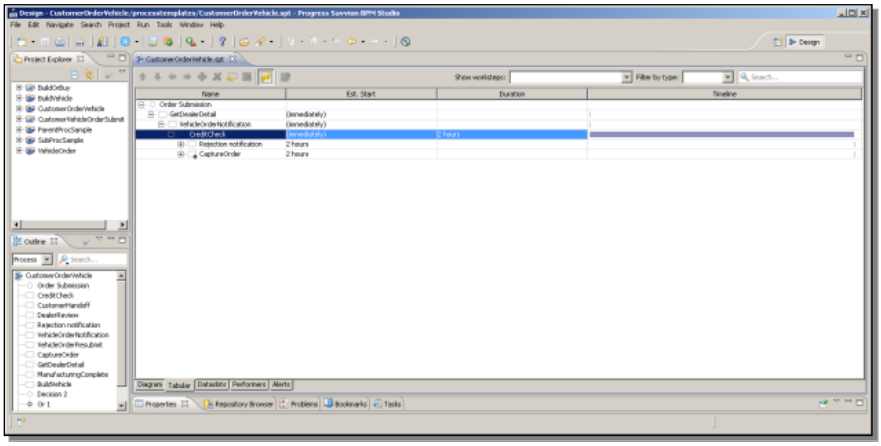
If I select the tab labeled **Tabular** then you see the same steps in a tabular form:



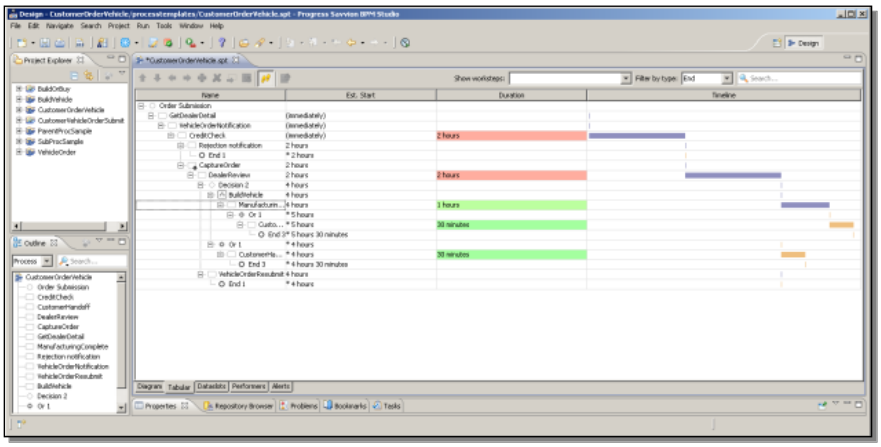
This has various advantages as an alternative, at least one to go back and forth to along with the diagram. You can sort the steps by their type or their name or any other characteristic. You can enter steps quickly in this format without worrying yet about lining them all up properly in a diagram. And you can even import existing application designs from Microsoft Project to this format, and then turn them into a diagram later on. There's lots of information you can display here. If I right-click on a column header, you can see the other information I can choose to display, such as a timeline:



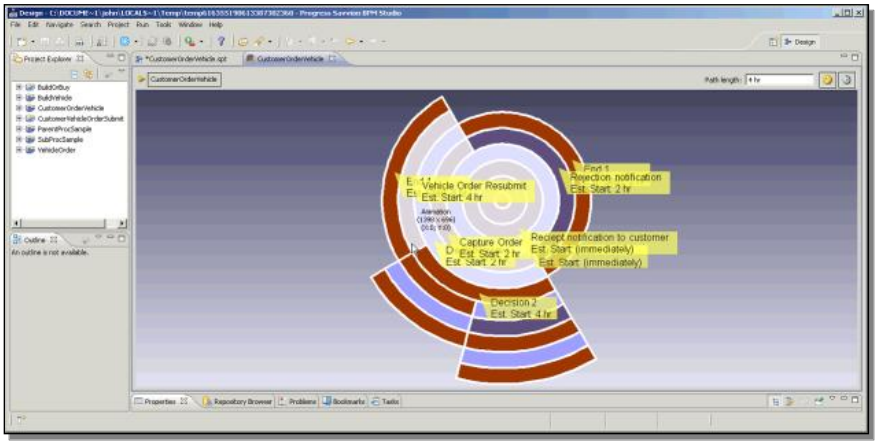
The timeline summarizes the estimated time any path through the process will take, based on durations you can assign to each workstep. There's another way to look at this as well, if I click on the **Path Browser** button in the Tabular View toolbar. This displays all the steps in a hierarchical format, instead of as a table, along with their assigned durations and estimated start time:



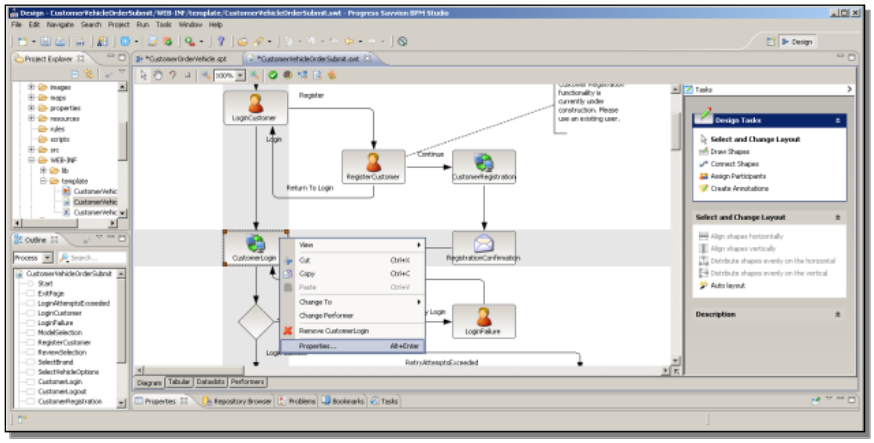
You can filter and sort in many ways here, too. For instance, if I filter by End steps, it forces all the paths to expand to show all the End steps. If I change a duration, for instance, by double-clicking on it, and, change an estimate of two hours to one hour, then the duration estimates for the path that includes that step will change, along with the timeline on the right. This is another way to view and share the design process with all the people who participate in it.



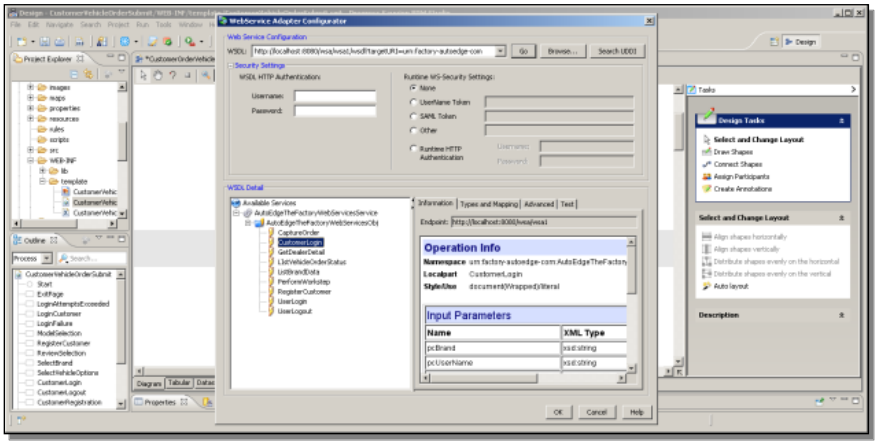
If I go back to the Diagram view, I can select another tool that can be useful for graphical analysis of a process, called the **360-degree view**. I can only take a limited look at this, because the amount of information such as workstep labels that is displayed depends on the screen real estate you're using for BPM Studio. I can move in and out on various paths of the diagram, and if I right-click I can rotate it to examine different parts of it, and see the implications of all the steps in various paths through the process:



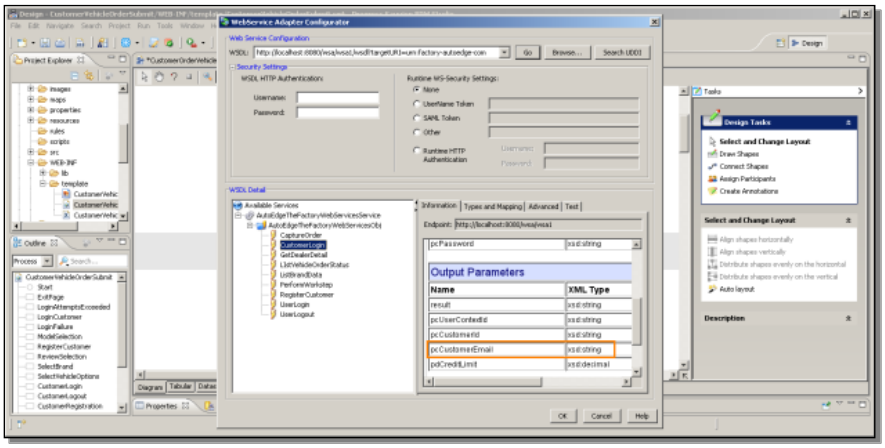
Now I'm going to look at another part of the Factory process. This has been separated out into a subprocess, which lets you break up a larger application into multiple pieces. And I want to show you a couple of things that have been defined in the worksteps. There's a step labeled **CustomerLogin**. You saw the forms in the overview video that led to the customer logging in after selecting a vehicle. Right-clicking on the step, I can see the workstep **Properties** I can set:



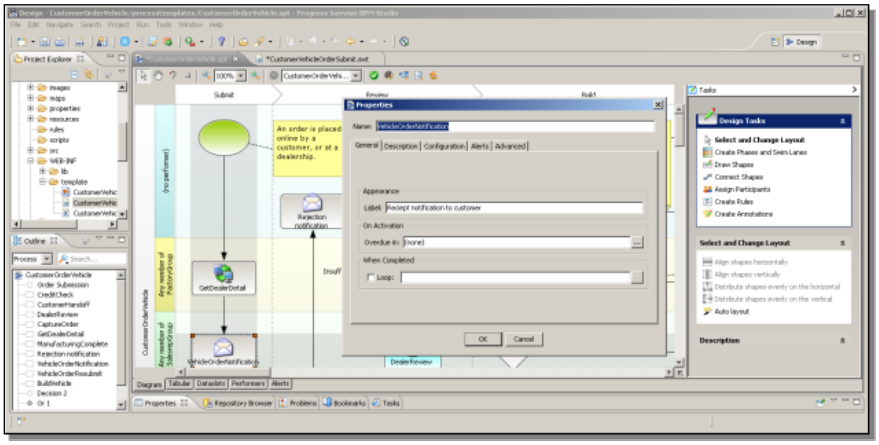
In this case, the workstep isn't a form, but an adapter that I can configure. This is a call to a Web service to run an ABL procedure; I showed you one of these in the overview. Here the call passes a brand and a username and password in:



The call gets back some user information, including the customer's email address, retrieved from a prior registration process:

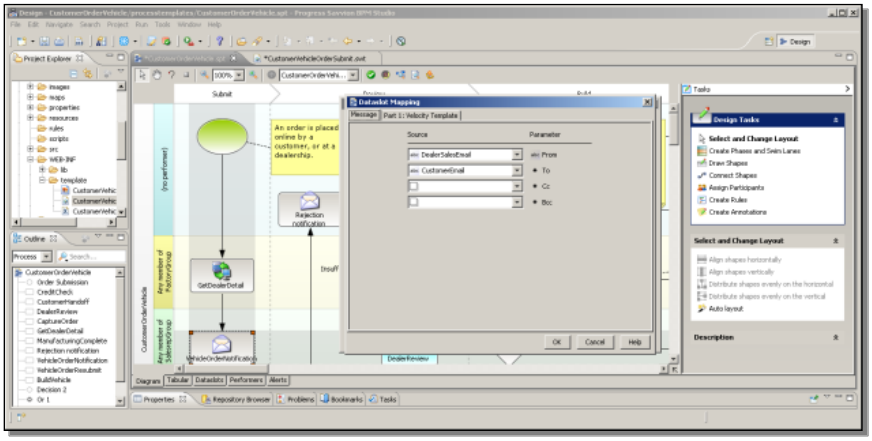


Now let's look at a place where that returned information is used. Back in the parent process there's a step that sends an email. Looking at its properties, it has configuration of its own:



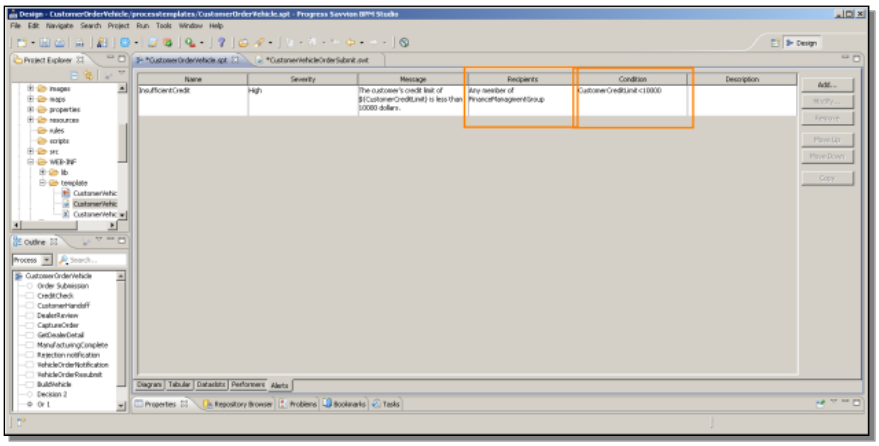
I can look at how the values the process defines, which we call Dataslots, are used in this step. Here you can see that the **CustomerEmail** address retrieved earlier in the

Web service call is used to supply the **To** address for an email message sent to the customer:

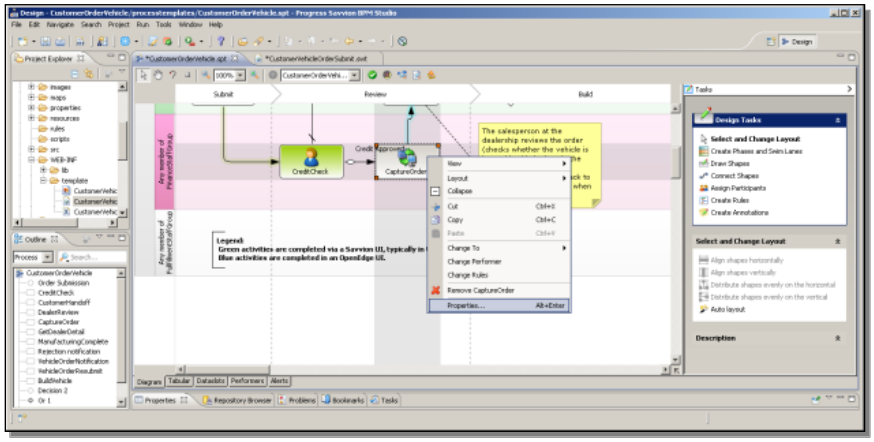


The process retrieves the customer's email address when they login, and uses it later to send an email message back to the customer. Remember that for a minute and I'll show you how that affects the process behavior at runtime.

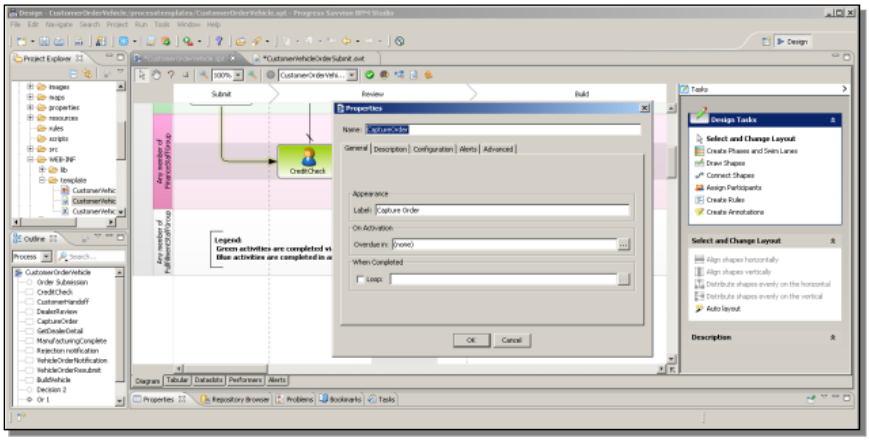
Here's one more thing on the design side. Another of the tabs on the design window is labeled **Alerts**. You can define alerts that tell people at runtime when things go wrong with a process or some event occurs that needs to be flagged and brought to someone's attention. In this example, the alert is checking whether the customer's credit limit is less than 10000, and sending an alert to the Finance Management Group if that occurs:



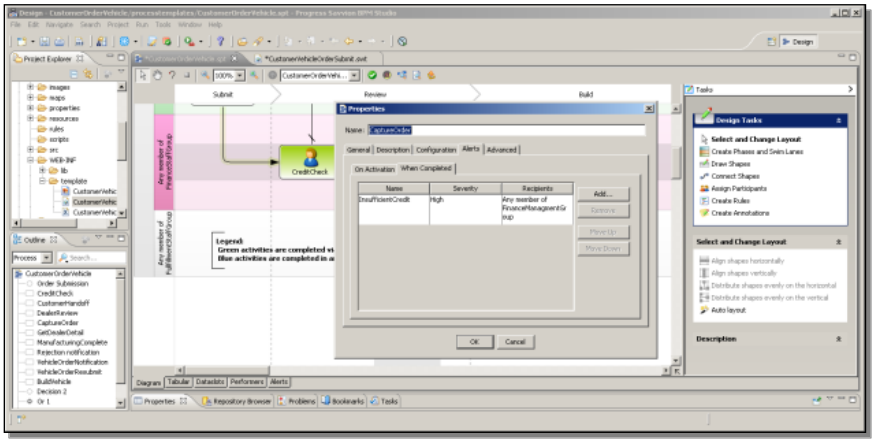
Now let's look at a step that uses that alert. You can associate an alert with a particular workstep in a process. Here, following the Credit Check step where a finance person decides whether to approve the customer's credit or not in a form, we've associated the alert we just looked at with the next step, which sends the order information back to OpenEdge:



Opening that step's **Properties**, I click on the **Alerts** tab:

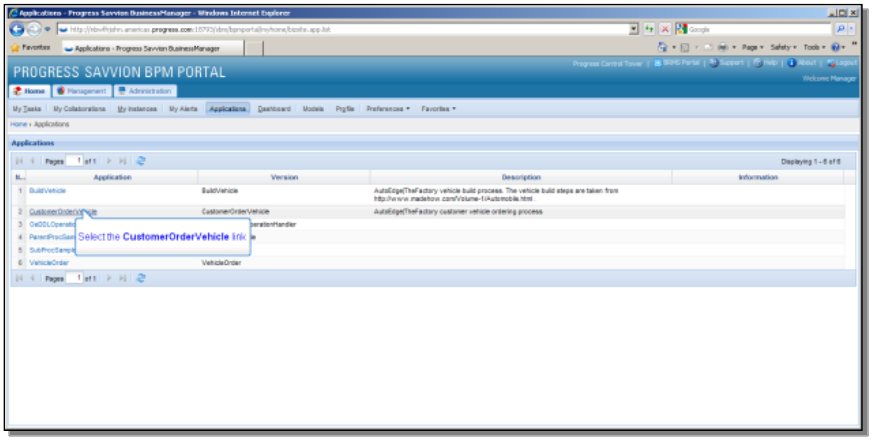


You can see that you can check an alert condition when a step is activated, or when it's completed. Here we've associated the **InsufficientCredit** alert with this workstep, to check when the step completes, as a backup to the manual credit approval done in the step before:

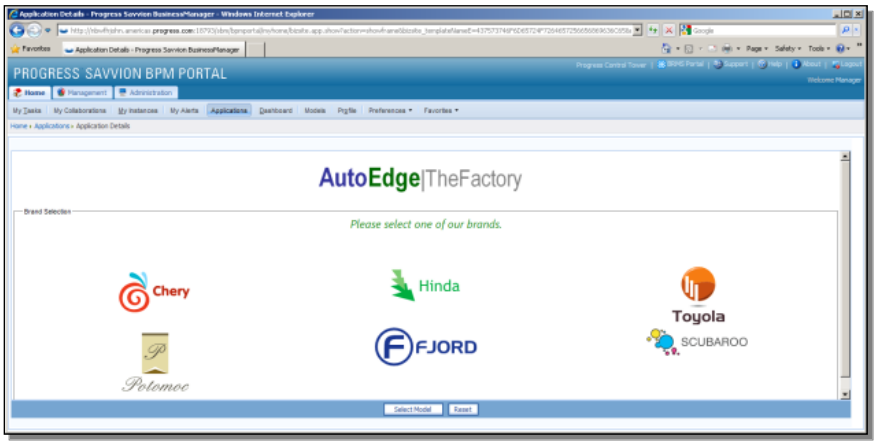


I've now shown you several of the ways that you can do process analysis at design time to aid you in creating a business process model. Next I'll show you how some of

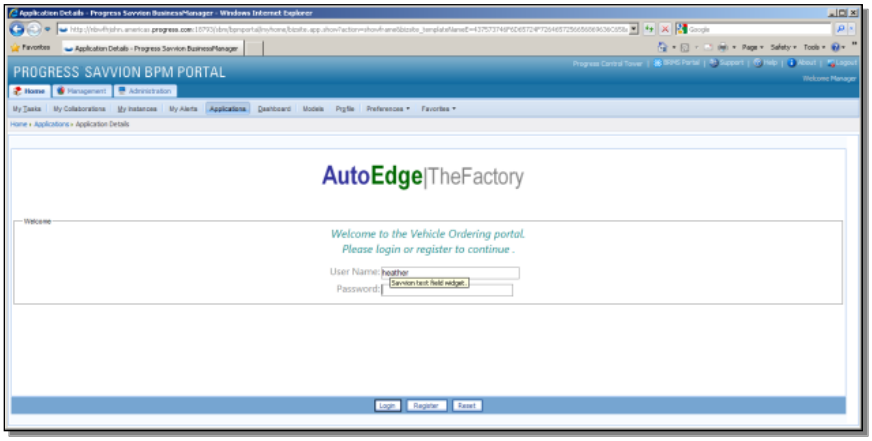
those things affect the runtime experience. Here I am back in the runtime environment, called the BPM Portal:



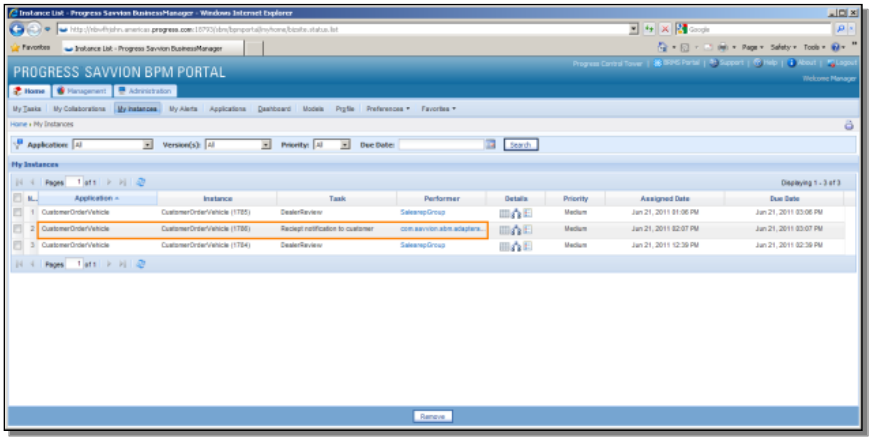
I can start an instance of the CustomerOrderVehicle application, and see the first form come up, where the customer picks a brand of car or truck. You've seen this before:



When the process gets to the login step, I enter a different username than I did before, one for which I've removed the email address, just to show some error checking at work.

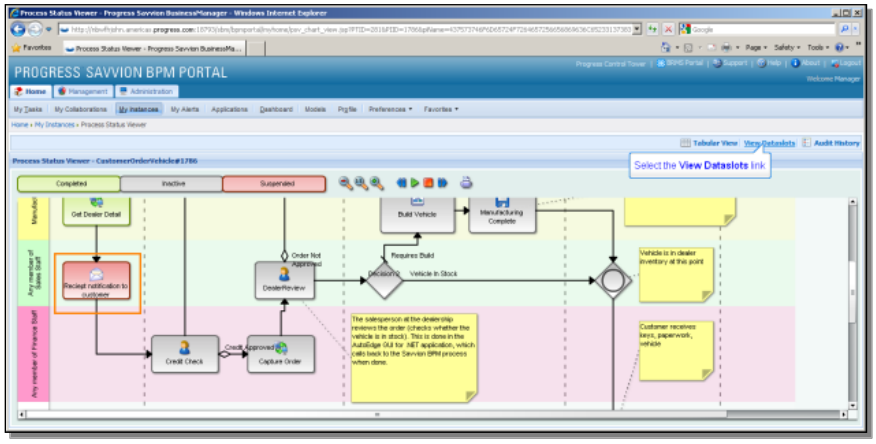


I select the **My Instances** tab to see the application instance I just started, and here it is along with a couple of others that are running:

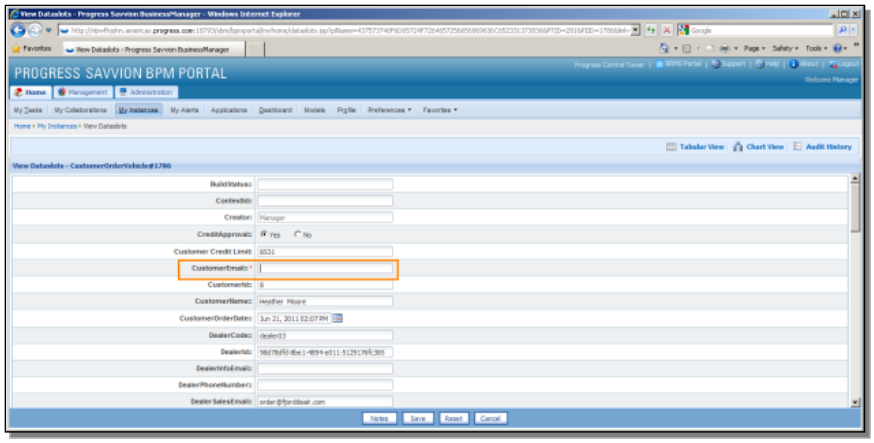


The one I just started is at the task step that sends an order receipt notification to the customer. I can check what's going on with this instance; one reason to do that would be that if I let it sit long enough it will flag that it's exceeded its expected duration, because it's not moving forward. Let's see why, by selecting what is called the Flow view. This is the middle icon under the section titled **Details**.

Below you see the process diagram in a runtime mode, with all the steps color coded to identify their status. Red means that the workstep has been suspended for some reason, and hasn't run to completion.

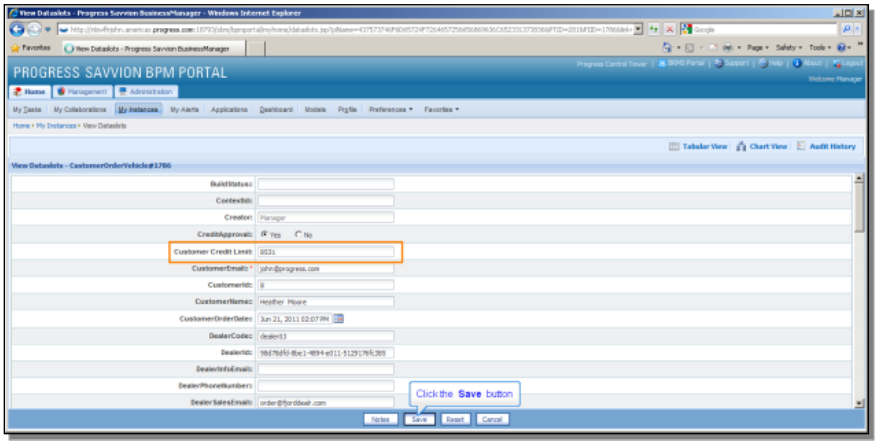


You can see the **Receipt Notification** step flagged in red. The process is hung at that point. What's the problem? One way to look into it is to select the **View Dataslots** link to the right. Again, the Dataslots are all the data values that are assigned and shared throughout the process.

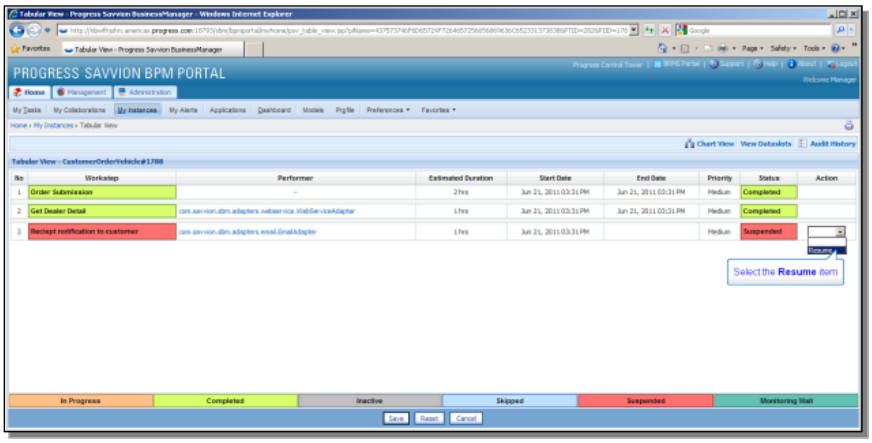


Here you see the problem. Because the customer data retrieved from the database through the Web service didn't include an email address, it didn't get set, and the email can't get sent without a To address in it. I can fix a running process instance on the fly by entering or changing a dataslot value, so I do that.

While we're here, take a look at Heather's credit limit, part of the data that was retrieved from the OpenEdge application. It's only 8000 or so, less than the 10000 minimum set up by the alert we looked at. Keep that in mind as we continue on. I save the email address change to continue:

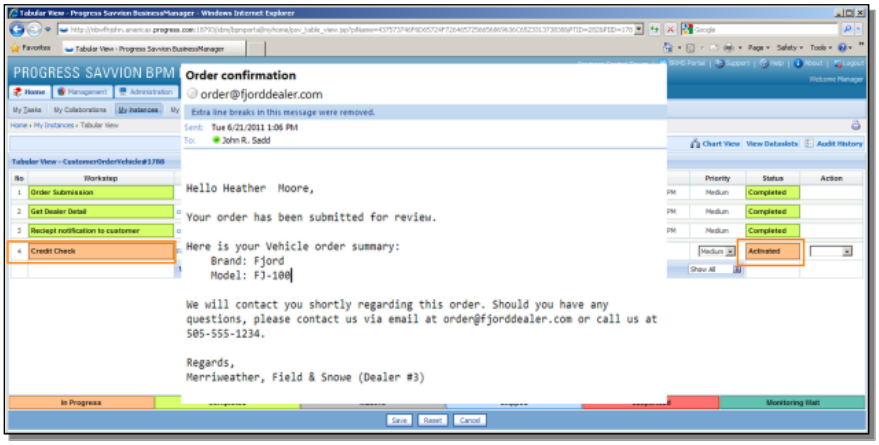


I could go back to the diagram, called the **Chart view**, and see how things are progressing, but I'll show another option, which is a runtime **Tabular View**. This also shows all the steps and color-codes them, but as a table rather than a diagram:

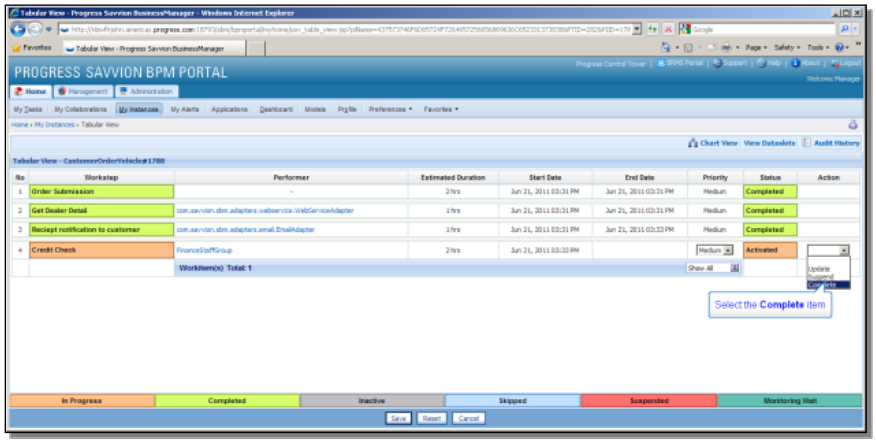


I can tell Savvion to **Resume** execution of the process now that I've changed its values, and Save that command by clicking on the **Save** button in the footer.

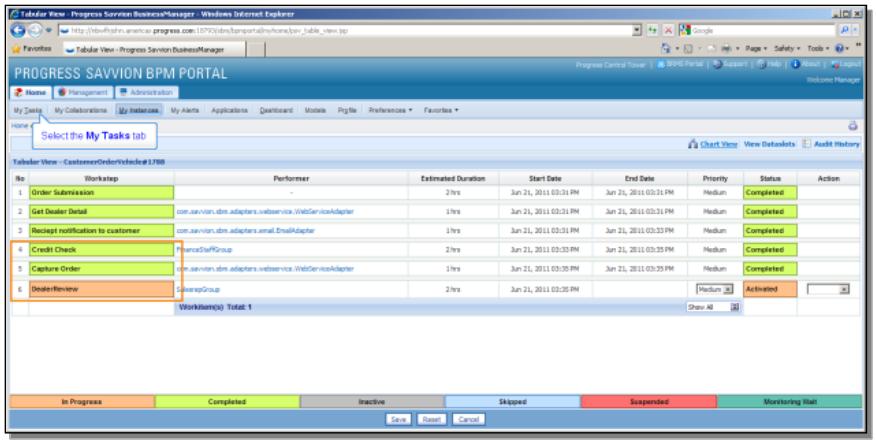
Below you can see that the process has moved on to the next step, and now an orange color (according to a revised color map at the top of the chart) indicates that the step has been activated. And I get an email message, the output from the receipt notification step, to confirm that it completed successfully. The **Activated** workstep is now the **Credit Check** step for the financials person. If credit is approved, there isn't anything that actually needs to be filled in for that step:



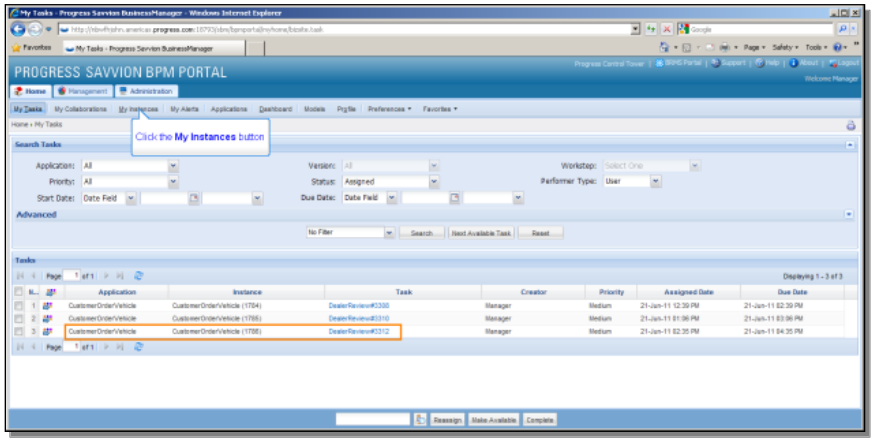
I can just tell the tabular view to complete the step and keep going:



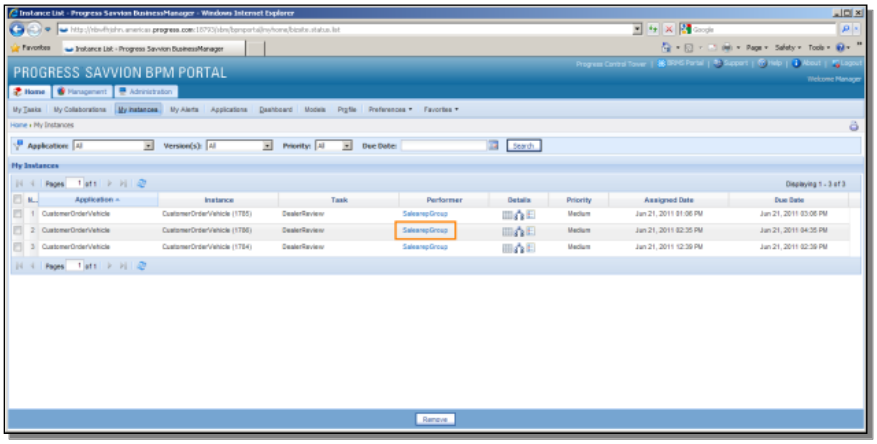
CreditCheck and the following workstep, **CaptureOrder**, have both completed, and the process has moved on to the **DealerReview** step, where someone at the car dealership reviews the order. Now I select the **My Tasks** tab:



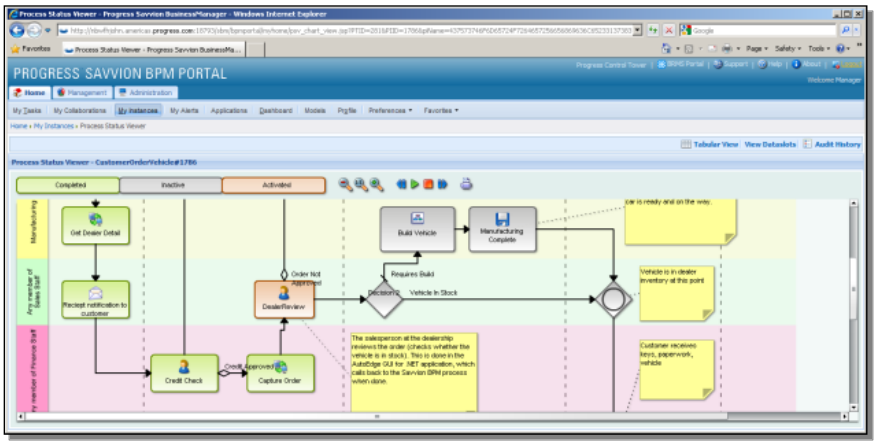
I see here as well that this instance, like the others that haven't completed, is at the **Dealer Review** step:



Going back to the instances, I see the performer for that step is the **Salesrepgroup**.

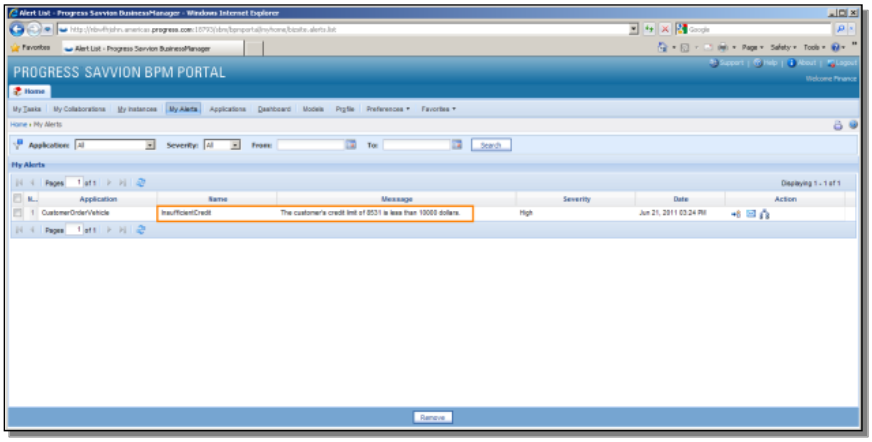


One last look at the diagram in **Flow View** shows me as well that nothing is suspended:



The **CreditCheck** and **CaptureOrder** steps have completed, and are marked in green. But wait: Heather's credit limit was only 8000 and some, so the alert should have fired to tell the finance people that, even though someone approved her credit, her order should be flagged for another look. To find out, I log out of this BPM Portal

session, and log back in as a financials person. Here I can click on **My Alerts**, and see the alert where the value of the **CustomerCreditLimit** dataslot has been plugged into the message:



This session has shown you just a few of the many capabilities that Progress Savvion provides to help you analyze and optimize a model as you're creating it, to set up alerts to tell you about important conditions when the application runs, and to track down problems at runtime and even fix them on the fly. The next series of videos will walk you through the process of creating and testing an entire new Savvion process from scratch.