USING APPLICATION DATA IN FORMS

John Sadd Fellow and OpenEdge Evangelist Document Version 1.0 July 2011

Building Business Process Applications Using OpenEdge BPM	BUSINESS
Using Application Data in Forms	MAKING
	PROGRESS
John Sadd	
Progress. OpenEdge.	
Progress. Savvion.	PROGRESS

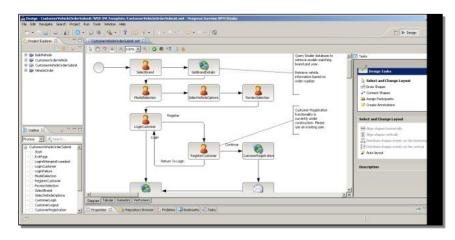


DISCLAIMER

Certain portions of this document contain information about Progress Software Corporation's plans for future product development and overall business strategies. Such information is proprietary and confidential to Progress Software Corporation and may be used by you solely in accordance with the terms and conditions specified in the PSDN Online (http://www.psdn.com) Terms of Use (http://psdn.progress.com/terms/index.ssp). Progress Software Corporation reserves the right, in its sole discretion, to modify or abandon without notice any of the plans described herein pertaining to future development and/or business development strategies. Any reference to third party software and/or features is intended for illustration purposes only. Progress Software Corporation does not endorse or sponsor such third parties or software.

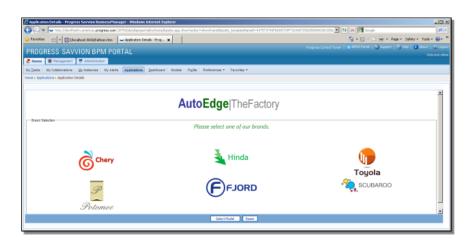
This document accompanies the two-part video which introduces you to the basics of how to populate fields in your forms with data retrieved from an application. The details on how to construct the actual calls out to the application are covered in a later session on using Web services. Here I just show how Savvion's controls can be assigned values at runtime, and how you can construct a flow that alternates between forms in the presentation and the calls that get data for them and pass the data from one form to the next.

The model below shows a sequence of steps in the Factory application that has been used before in videos and papers in this series:

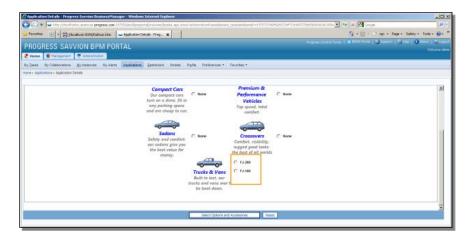


It's called a **Presentation Flow**, a special type of sub-process of a larger process that represents a series of forms that an individual user goes through to accomplish one of the major steps in the Savvion process, in this case, a customer placing an order for a vehicle.

Just to review how this looks when this part of the process runs, I can start up an instance of the **CustomerOrderVehicle** process, and in the first form that comes up, the customer selects a brand:



When the brand is selected, all the other brands are effectively grayed out. In the next step the user sees all the available models for the selected brand. (As you can see, there isn't much data in the sample database for some models yet.)



For truck models in this case I just have two choices. This is the key to what I'm covering in this paper: Where did these two values for Fjord truck models come from, and how did the process developer use them to assign the values for this radio set? That's the question this paper answers. Continuing through the Flow, the customer selects a model and proceeds, and goes on to the next step of selecting options, which are also dynamically populated from data retrieved through the OpenEdge application that is executing in the background:

	When here restalling home himster, and sho	w/actives/ws/awathints_terelations?ed	17573746/606572497264657256655866636365658 💌 🔩 🔀 Geogle	
Pavorites 💮 + 🔯 (locahod: 90%)/attor.htm 🖕			🔓 • 🖸 - 🗅 🛞 • Papa •	
			Progress Cantrol Tower 😹 5005 Partal 🎝 Support 🙃	
ROGRESS SAVVION BPM PORT	AL			Webser
Home 🔮 Hanagement 💂 Administration				
Tasks My Collaborations My Instances My Alerta	Applications Dashboard Models	Prgfie Preferences * Favorites *		
e + Applications + Application Details				
		AutoEdge TheF	actory	
		-uto-uge men	actory	
Maonroof				
Maorraof		your vehicle by selecting addit		
Baterior Options				
Interior Options Exterior Options	Please customise ;	your vehicle by selecting addit	ional options and accessories	
Saterior Options Cristic Options	Please customise j	your vehicle by selecting addit	ional options and accessories	
Interior Options Enterior Options Seal Mathal C Plash & Comfort Microfiber	Please customise ; Tim Car C Lig	your vehicle by selecting addit	Ional options and accessories	
Batterior Options Coloriar Options	Please customise ; Tim Car C Lig	your vehicle by selecting addit	ional options and accessories	
Interior Options Extension Options Set Marcal Plant & Connort Microliber	Please customise ; Tim Car C Lig	your vehicle by selecting addit ter pr Gray a State Gray	Accessive Accessive Tensive Tensive System Tensive Prenam Social System	
Interior Options Extension Options Set Marcal Plant & Connort Microliber	Please customise ; Tim Car C Lig C Ber	your vehicle by selecting addit use pe Gray n Base Gray	Accessive Accessive Tensive Tensive System Tensive Prenam Social System	
Interior Options Extension Options Set Marcal Plant & Connort Microliber	Please customise ;	your vehicle by selecting addit use pe Gray n Base Gray	Accessive Accessive Tensive Tensive System Tensive Prenam Social System	
Interior Options Extension Options Set Marcal Plant & Connort Microliber	Please customise ;	your vehicle by selecting addit use pe Gray n Base Gray	Accessive Accessive Tensive Tensive System Tensive Prenam Social System	
Interior Options Extension Options Set Marial Plant & Connort Microfiber	Please customise ;	your vehicle by selecting addit use pe Gray n Base Gray	Accessive Accessive Tensive Tensive System Tensive Prenam Social System	

That's as far as I need to go in the Presentation Flow. Let me drill down into a few of the design details. The first step that has a visible form is called **SelectBrand**. I can open the form for that step, and take a look at the images in the form. You can see that there are actually two images for each brand, one color, and one black and white:

🚠 Design - SelectBrand - Progress S	arriton 1874 Studio			
File Edit Insert Navigate Search i				
	• 🔝 @] 💁 •] 🎖] 😅 🛷 •] 🥹 • 🕅			🗈 🕨 Design
Project Explorer 21 📃 🗆	Custoner/ehicleOrderSubmit.ovt	8		° 0)
😑 😫 🐨 🏹	Tahona ¥9 ¥ ¥ 7 5	i 🗹 🖾 🖬 📥 🔍 🗅		
Song Dudivinde Song Dudivinde Song Dudivinde Song Dudivinde OrderSubnit Song DudivindeOrderSubnit Song VehideOrder		🛓 Hinda		
	P Polomoc		SCUBAROO	Controls Controls Prion Controls Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Prion Pr
BE cutine 12	Potomoc		<u> </u>	Leynot A madar One-Clasm Table Theo-Clasm Table
	L			Custon Table
	1	Select Model Record		Field Sat
				v Tasia Eventa
	Ports Nakiation Script			
	🗖 Properties 🕄 🚺 Repesitory Browser 其 Problem	s 💷 Beolmanis 🕢 Tasks		e
] D *				j.

If I select the color image for Fjord, I can drag the Properties view back up to see what's defined for it:

	· 🔝 (8) 9₂ -] 9 ⊕ - 4 -] 5 • *Cattorie WeldeOrderSubsit.ovt		E Design
😑 🎕 👘 🔻			
Se Buddvinde Boddvinde Boddvinde Se DostowerOrderVeticle Se DostowerVeticle Se DostowerVeticleOrderSubnit Se Se VetideOrder	Chery	🛓 Hinda	Form Tasks
	P Potomoc	CUBAROO	Scarbab Scarbab Scarbab Scarbab Scarbab Scarbab Scarbab Scarbab Free Programmes Leynext
🔠 Outline 🕄 👘 🐨 🖬 🖬 An outline is not available.	Porm Waldation Script	U	* Taks Events
	Properties 21 Le Repository Browser		e***0
	Control Properties (Image) - ImgFjordLi	a)	
	Colors Description Events Events URL: [mageut/fund_cope Access ance	re 🕑	<u>*</u>
] *		Watts 10 Height 60 Prove	

The **ID** is the name of the image control in the form. The **URL** shows the path to the actual file name for the image. What's new here is that there is an event defined for the image, so the image is really functioning as a button control. Selecting the **Events** tab in the properties, you see that there's an event defined for **onMouseDown**, the click event:

B Design - SelectBrand - Progress S. File Edit Insert Navigate Search P						
😁 - 🖾 🛆 🕋 🖉 🕓	12010	94 • J 🕈 🛛 😅 🛷 • J 😒 • 🕅	1		E 🕨	Design
Project Explorer 21 -	Customer's	rehideCivderSuberit sivit	Rand II			° (
😑 😫 🖗 🍸	Tahona	¥9 ¥ 6 7				
Brukhvinde Gostoner OrderVehicle Gostoner OrderVehicle Gostoner WehldeOrderSubnit Software WehldeOrder		GChery	🛓 Hinda		Tools Trom Tasks Torm Tasks	*
		P	FJORD	SCUBAROO	Controls Controls Controls Controls Prove Programmes	
() Cutine 23 () 전 프 An outine is not evolution	Pors Raldatio	Potomoc	FJORD	SCOBARGO	Lapent TBHS Courds	* _
	Properties	🕄 🚺 Repository Browser 🛣 Prabi	Jens Doolmaris 🕢 Tasks			- e ° ° °
	Control Prop	perties (Image) - imgFjordLogo	,			
	General Choices Description Events	Description The Events subtra enables you to d Events subtra enables you to d Events oriflur oriflur oriflur controls contro	Biffee events, functions, and actions that are associated with	ra field in the form.		Add Remove New Action

Before looking at the code for that event, there is another interesting element on the form, a text field under the footer:

💑 Design - SelectBrand - Progress S						
File Edit Insert Navigate Search F						
			- 8 0 -			Design
Presiect Explorer 21 -	Customerve	ideOrderSubait.ovt	ectilizand [3]			° 0)
😑 😩 🐨	Tabona					
18: Webdwinde 응 Webdwinde Rew CostonerOrderVehicle Rew CostonerWebderOrderSubmit Re WebdeOrder		P Potomoc	FJORD	's V.	Z Todo Todo Term Taks Cantal Cantal Contal Conta	*
👔 Cutine 🕄 🚽 🐨 🖽		electred vehicles and	Select Model Reset		E Port Prograets	* <u>*</u>
An outine is not evalable.	Forta Validatio	Line			v Taska Events	
	Properties 3		Problems 💷 Beckmania 🕢 Taska			8 V ° D
	Control Prop	rties (Image) - ingFjordLo	aga			
	General	* Description				Renome
	Choices	The Events subtab enables you t	to define events, functions, and actions that are associated	with a field in the form.		Resource
	Description	Events Name				New Action
	Events	onflux onflux onfluxeBown ingFardLops_on onfluxeBown onfluxeOver onfluxeOver onfluxeOver onfluxeOver	Plastone			
] 19					1	

It is bound to the Dataslot called **SelectedVehicleBrand**. The field didn't actually appear before when I ran the process. If I select the field and look at its properties, you can see that it's called **txtBrand**, and it's a **String** field. You can also see that its **Visible** property is off:

🔒 Design - SelectBrand - Progress :		
File Edit Insert Navigate Search		
🗂 • 🔛 🗠 🗎 🔠 🕄	• [🔐 @] 💁 •] 🦹 [😅 🛷 •] () = 11 + 11 + 12 + 14 + 14 + 14 + 14 + 14 +	📑 🕨 Design
Project Explorer 31 📃 🔍 🗆	* *Cutoner/WhideOrderSubart.out	- 0
😑 😫 👘 🔨	Tahana 🗶 9 🕷 🖉 🖞 🗉 🖬 🍓 🖻	
18 월 buddheds 日월 Octowershite 18 월 Octowershite 18 월 OctowershiteOrderSubmit 운 월 WeldeOrder	Potomoe	2 Table 2 From Lasks 2 Controls 4, PAm 5 GASServers 1 table Events 1 table Events
🐮 Outine 💠 👘 🖬 🖬		
An outline is not available.	Control Properties (Text Field) - totBrand	
	General ID: bithrend	
	Colose Electrition Tak Order Parsatz 2000	2
	Apparation Longity 30 Max. Longity 226	
	- Reference	
 **		l.

This is a hidden field, set in the background to hold a value that gets passed from form to form, which the user doesn't see. Let's look at the code for the form now by selecting the **Script** tab. The code that gets run when the form is first loaded is the **onLoad** event.

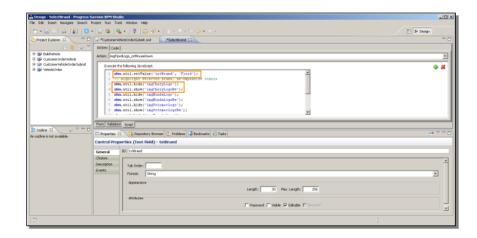
🚋 Design - SelectBrand - Progress Si			_0×
File Edit Insert Navigate Search P			
		2 •] ?] G ≪ •] 2 • # • # • # • + •	📑 🕨 Design
		WideOrderSubait.out	° 0)
🖻 😫 🐨	Actions] Cade	1	
8 W Duid/whide 8 W OustowerOrder/teliscie	Action: form	orizad	*
18 🐷 OustomerVehideOrderSubmit	Ereoter	the following JavaScript	× ×
R 🕼 VehicleCider	1 // 2 de 3 de 4 de 5 de 6 de	<pre>setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setLister(staffset); setL</pre>	
	-		<u> </u>
An outine is not evaluate.	Perm Kalidata	01. <u>30907</u> 21. <mark>(]),</mark> Repository Brower []2, Publiess []] Bodmadu []2 Tada	
	Control Pro	perties (Text Held) - txtBrand	
	General	Ex (billiand	
	Choices Description Events	Tak Oder Fenal: 2mg	*
		Length 20 PierLength 26	
] 0*			×

You can see that the code – and this is JavaScript – is executing a **hide** function that's in a library called **sbm.util**:

<pre>// highlight all brands</pre>
<pre>sbm.util.hide('imgCheryLogoBw');</pre>
<pre>sbm.util.hide('imgHindaLogoBw');</pre>
<pre>sbm.util.hide('imgPotomocLogoBw');</pre>
<pre>sbm.util.hide('imgToyolaLogoBw');</pre>
<pre>sbm.util.hide('imgScubarooLogoBw');</pre>
<pre>sbm.util.hide('imgFjordLogoBw');</pre>

There's a color and a black and white image for each brand, and this little block of code is hiding the black and white images when the form first comes up. Now look at the code that gets run when the customer clicks on one of the visible color images, like the Fjord image:

```
sbm.util.setValue('txtBrand', 'Fjord');
// highlight selected brand, de-emphasise others
sbm.util.hide('imgCheryLogoBw');
sbm.util.show('imgGheryLogoBw');
sbm.util.show('imgHindaLogoBw');
sbm.util.show('imgPotomocLogo');
sbm.util.show('imgPotomocLogoBw');
sbm.util.show('imgToyolaLogoBw');
sbm.util.show('imgToyolaLogoBw');
sbm.util.show('imgFjordLogoBw');
sbm.util.show('imgFjordLogoBw');
sbm.util.hide('imgFjordLogoBw');
sbm.util.hide('imgScubarooLogoBw');
```

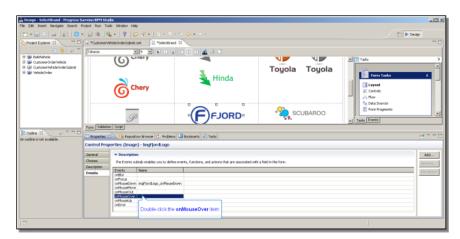


Remember that the form has a hidden field at the bottom named txtBrand. The first highlighted line is setting that field to **Fjord**, in preparation for passing that value on to the next step in the process. Then it hides all the color images for the other brands and shows their black and white images instead. Even though the design window for the form shows the color and black and white images stacked above one another, at runtime the form layout is done dynamically based on what's visible, so the black and white images just appear in place of the color ones, and there's no flashing or jumping around of the display.

Below is a screenshot of the **BPM Studio Users Guide** in the documentation for Savvion. Just to make sure you know where to look to find all of the built-in function libraries,I can select the **Util** package down in the appendices at the end of the book, and there you will find a summary of what all the **sbm.util** functions do. All of this information is available for you when you start to write event handlers of your own:

🖏 🛧 🔷 💷 J 550 🖲 🖲 [100% *	E Rnd •		
okmarka 🗉	Util Package (sb	m.util)	
)• 📑	Table 125: presents	the APIs available in	n the sbm.util library.
Defining Workstep Presentation	Table 125: Util Pa	ckage APIs	
Format Using Alerts	Function	Parameter(s)	Description
Using Form Editor	addOptions()		Allows user to add items in the specified list in the form.
Configuring Dashboard Widgets		aList	The name of the list control on the form.
Working with Service Designer tool		anitem	An array of primitive elements.
Working with User Management tool	getField()		Returns a reference in the domain object model.
Working with Business Calendar Managing Messages and Channels		id	Returns the value of a reference in the domain object model.
Managing Business Objects	getText()		Returns the text of a current option.
Using Managed Adapter Browser		id	Returns the value of the text.
Defining Rules with Rule Wizards	getValue()		Returns an HTML element.
Customizing Rules using Rule Editor		id	Returns the value of the field.
Managing Changes in Processes		aList	The name of the list control on the form.
BizLogic examples	getValues()		Returns the list (as an array) of all items in specified list.
Effect Package (sbm.effect) Validate Package (sbm.validate)		aList	The name of the list control on the form.
Util Package (ibm.util)	hide()		Hides the specified control.
Adaptet Package (sbm.adaptet)		control_name	The name of the control that is to be hidden.
Using the License Update Tool	hide()	-	Hides the specified label.
Glossary		label name	The name of the label that is to be hidden.
Index	RemoveAllOptions		Removes all items from the specified list.

Back in the design window for the form, I want to give you a quick example of how you can put event handlers like this together yourself. Using the color Fjord icon again, I go back into the events for it, and this time I define a **mouseOver** event. I do that by double-clicking on the event:



I get a default event handler name that I can accept or rename as I wish. Then I press the **View Action** button. Since there's no event handler code for this event yet, I just see a message that says, "**Click here to add action**", so I do that. Immediately you can see that there are many types of actions that you can define handlers for. The **Select Expression** dialog shown below is there to help you define common actions in event handlers without having to write any code at all. In addition to **Data Operations** like setting or copying a field value, there are expressions you can define for the effects supported for a field, for a widget, and so forth. I select **Field Effects**, and from the different effects an object like the image might support, I select **Hide**.

Design SelectBrond Progress File Edit Insert Navigate Search		ois Window He							
11 • El 🍝 🗎 🕹 🖉								E 🕨 De	
Contract Explorer 21			set 🗄 "Sections 🖬						- (
😑 😫 😒 1	Actions] Cade	1			Select D	quession.	×		
3: 🐲 BuildVehicle 3: 🐲 CustomerOrderVehicle	Action: IngF)	ordLogo_onMouse	Over		Cabegory:	Field Effects	-		*
8 📴 CustomerVehideOrderSubmit						SvechOff Pult	-		÷ х
8 🐨 WehideOrder					Expressions:	Pulcate			
						Hde	-		
					Make a field	p	-		
					mare a neis	mode	- 1		
							- 1		
							- 1		
	Fam Taldati	00. Control			-				
E Outre St St T		-				OK Cancel	ł		
An outline is not available.			tory Browser 🔝 Problems 斗 t	soolimarks 💰 Tasks					
	Control Pro	perties (Imag	e) - imgFjordLogo						
	General	* Descriptio	an i						Resare
	Choices	The Events a	ubtab enables you to define event	s, functions, and actions	that are assoc	aked with a field in the form.			Remove
	Events	Events	Name						Vew Action
		onFocus							
		onMouseMove	n ingfjordLogs_onHouseDove						
		orMouseOut orMouseOver	indfierd.cos onNouseOver						
		onMouseLip onError	and in proceeding on some of the						
		0.070							

Now the event handler UI prompts me to pick a control, in this case any of the controls on the form, so I select the black and white Hinda Logo.

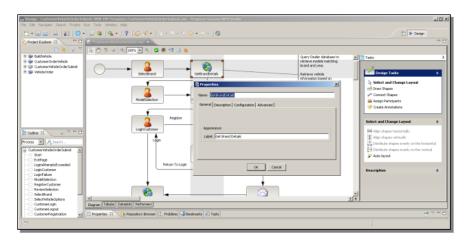
B Design - SelectBrand - Progress S File Edit Inseit Navigate Search F				
			4. 15. 4. 4. 6	📑 🕨 Design
Project Explorer 11 -	Castoner%	RideOrderSubmit.	A Selectional D	° 0
🖻 😫 🐨	Addons Cade	1		
R 😻 Buldvihide R 🐨 OustomerOrderVehicle	Action: impfijo	rdLago_onMouseO	e de la constante de	
8 🕼 CustomerVehideOrderSubmit 8 🐲 VehideOrder	Hide field	mg4ETR.ogo	×	x e
	Farm Waldatio	Isbrand Isbrand ImgChenyLoppBiv ImgChenyLopp ImgHindaLopp		
🐮 Outine 💠 👘 🚥 🖬			v thomase) [Problems) 🗳 Biologuadus) 🕢 Tacks)	
An outline is not available.			nomen (, monen) 4 nomen () none)	
	General	• Description	- mblo orolo	
	Choices		ab enables you to define events, functions, and actions that are associated with a field in the form.	Retare
	Description Events	Events ontilue	Vane	Menr Action
		onPacus onMouseDevin onMouseMove onMouseDut	ng perlang perlandaran ng perlang perlandaran	
17				

The result is that when I select the Fjord brand, and its color logo is displayed and all the black and white logos are shown for other brands, and then I mouse over the Fjord logo, the Hinda logo will disappear.

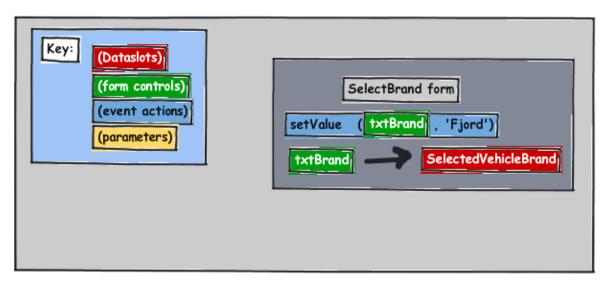
Now to make it come back, I define a similar handler for **mouseOut** of the Fjord image. Instead of Hide, this time I select the **Show** action, so when I mouse out of the Fjord image, the Hinda logo will appear again.

Now I can go back to the form, and close it to save the changes I just made, and then save the Flow that it's a part of. As I mentioned before, the flow is a special type of subprocess that gets deployed to the server on its own. I click the **Deployment** button to deploy the flow. Once I do that and start up another application instance, I can show the effect of the new event handlers I just defined. Once I get to the Select Brand form, and select the Fjord button, mousing over that image makes the Hinda logo disappear and then reappear.

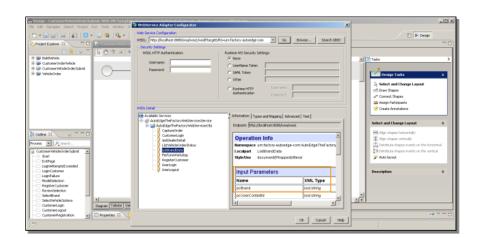
This quick demo of how you can define event handlers of your own shows you that the design interface provides a lot of guidance for you so that in many cases you don't actually have to write any code at all. Now take a look at the next step in the flow, called **GetBrandDetails**. This one doesn't have an image of a person on it in the process model diagram, so it's not a step executed by a person. In its properties, I see the **Configuration** tab; clicking on that will tell me what kind of step it is.



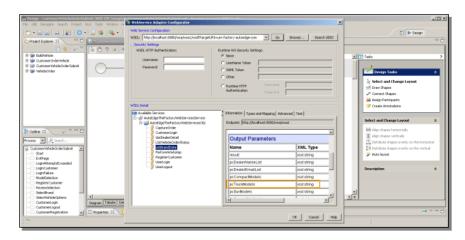
First let me review how I got here. The first form in the flow has an action that sets a hidden field called **txtBrand** to the value Fjord if I click on the Fjord image. Then it binds that variable to the Dataslot **SelectedVehicleBrand**:



Now let's look at the **Configuration** tab to see what this step does. Clicking the **Configure** button, I see that this step is defined as a Web service adapter, in this case, one that calls out to an entry point in an OpenEdge application. One of the input parameters is called **pcBrand**:



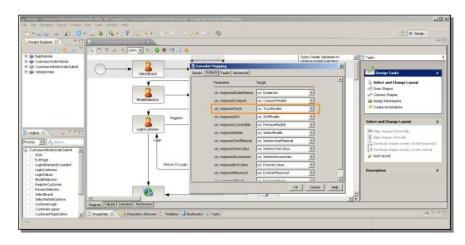
If I look at the output parameters, I see a number of parameters including one called **pcTruckModels**, which is where the list of Fjord truck models like **FJ-100** comes from:



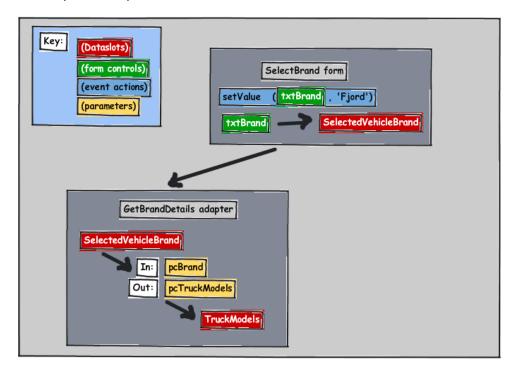
When I OK this dialog, I'm placed into the **Dataslot mapping**. This is where I'm able to pass values from one form to this step through variables and dataslots, and from here on to the next step as well:

💑 Design - Eustomer#ehicleDrderSu	ubmit/WEB_INF/template/CustomerVehicleOrderSubmit.awt -	- Program Sovvicei BPM Studie	
File Edit Navigate Search Project	Run Taols Window Help		
	• 💆 🚳 🛛 💁 🖉 🖉 🖓 🖉 🖉 • 🖉 • 🖉 • 👘 •	¢•••↓	📑 🕽 Design
Project Explorer 33	🗈 *Castones Vehide Order Subait, aut 🕷		° D)
😑 😫 🖉 🗸	R 🖱 🤈 🗉 🔌 100% 💌 🔍 🥥 🌒 💐 🗟 🤹		
18 🐲 BuildVehide		Ouery Dealer database to	7 Taoka
🗄 🍅 CustomerOrderVehicle		settieve podek paidding	
8 @ CustomerVehideOrderSubmit 8 @ VehideOrder		🖹 Datadat Mapping 🔣	-2
12 W VENDEL/DEF	Selectional	Posts Outputs Faults Advanced	🖉 Design Tasks 🛛 🕹
		Source Parameter	Select and Change Layout
		er Selected/violefrand w ein pefeged	🖬 Draw Shapes
	ModelSelection		
	ModelSelection	an Cantentid 💌 🖮 public Contentid	🚑 Ansign Participants
			🕫 Creake Annotatione
	Register		
			Select and Change Loyout ±
BE Outre 23 St T == []	LoginCustomer		High shapes horizontally
0-000 F 10 0	· · · · · · · · · · · · · · · · · · ·		Algo shapes vertically
Process P P Search	Lógin		To Distribute shapes evenly on the horizontal
CustomerVehideOrderSubmit			Distribute shapes evenly on the vertical
-0 Start 			🌮 Auto layout
 Extrage LoginAttemptsEccended 	Return To Login		
- LoginCustomer			Description 2
- LoginFalure - NodelSelection			
RegisterCustomer	•	for the set of the set	
- ReviewSelection		OK Cancel Help	
CustomeLogin	Diagram Tabular Dukadots Performent	2161	
- CustomeLogout			
CustomeRegistration	🗖 Properties 🕄 🚺 Repestory Browser 🚺 Problems 🗳 B	eolmans 🐔 Tasis	e
1.02			

Remember that the **SelectBrand** step has an **onChoose** event that places the brand represented by an image into the hidden field called **txtBrand**. You also saw that txtBrand is bound to a dataslot called **SelectedVehicleBrand**. This is where the process is using that value. In this step, it takes the Dataslot value, which is available to all steps in the process, and maps that to the input parameter **pcBrand** that goes out to the web service call:



For reasons I won't go into in this paper, the output parameters from the service come back with names beginning with **response**, so for example, **responseTruck** is the output from the OpenEdge application call with all the available truck models for the selected brand that was passed in. In the screenshot above you can see that is is mapped in turn to a Target dataslot called **TruckModels**. This diagram will help review the steps in the process:

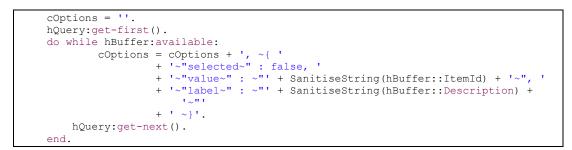


The **SelectBrand** form set the **SelectedVehicleBrand** Dataslot. Now the Web service adapter is using that value to set the input parameter to a call out to

OpenEdge that returns a list of truck models, among other things, and that in turn sets another Dataslot called **TruckModels**. Take a look at just a bit of the actual ABL code that's being called here. On the OpenEdge side, it's a procedure called **service_branddata.p**, which is part of the Web service proxy that provides access to the ABL procedures from Savvion. You can see the **pcBrand** input parameter along with the **contextID**, as well as all the output parameters with various kinds of information for whatever brand was passed in:

```
define input parameter pcBrand as character no-undo.
define input parameter pcUserContextId as longchar no-undo.
define output parameter pcDealerNameList as longchar no-undo.
define output parameter pcCompactModels as longchar no-undo.
define output parameter pcTruckModels as longchar no-undo.
define output parameter pcSuvModels as longchar no-undo.
```

Later in the same procedure is code where the data in the ProDataSet that holds all the values for the brand is converted into a JSON object with **selected** and **value** and **label** properties, which is what a Savvion widget like the radio set expects:



The **value** property in the JSON example below (truncated from its full length) represents the unique character string key that the OpenEdge application uses to identify each row in the database:

```
<json>
[ { "selected" : false, "value" : "fd1603...", "label" : "FJ-200" },
        { "selected" : false, "value" : "fd1604...", "label" : "FJ-100" }
]
</json>
```

To continue, I open the next user-driven step in the process, called **ModelSelection**, and open its form. This is the form where model names are displayed, and here under Truck Models, take a look at the control. It is bound to the **TruckModels** dataslot, which was populated by output values from the Web service in the previous step.

Design - ModelSelection - Progres	Sarvin HM Stala	
File Edit Insert Nanigate Search F		
] 😁 • 🖾 📥 🛯 🏔 🛛 🔝 • 😁 •	🔐 @ Q. • 9 😅 🖋 • 5 - 8 - 9 - 9 • 9 •	📑 🖻 Design
Project Explorer 23 📃 🔍 🗖	*Castone-WhideOrderSubait.out	° 0
😑 😫 🐨 🍸	Tahona M 9 M 6 M U S M L A R h	
Se Se Dubdvihide Se Se Costoner Order Vehicle Se Se Costoner Vehicle Order Vehicle Se Se Costoner Vehicle Order Submit Se Se Vehicle Order	Sinders	2 Talo >
	bilding and exceptor have considered in the second	Controls Controls Notestand Train T
()E oudine 53 → □ = □ An oudine is not evaluate.	Track A tase But to but, as tracks and exerce sent 1 to but, as tracks and	Errord 8 Pader Groediaan Table Teocham Table Groen Table Groen Table
	Select Options and Accounting Facest	Table Devets
	Form Nakdation Script	
	Teoperties 83 💦 🖪 Repository Browser 🕄 Phablens 🔍 Bookmarks 🕢 Taols	e**=0
] 5 9		

Looking at its properties, you see that its name is **uxTruckModels**, and it's a radio set widget:

a Design - ModelSelection - Progres	a Service BPH Star	dio				
File Edit Insert Navigate Search i						
] 😁 • 🖾 📥 🛯 🔝 🛛 🔘 • 😁 -	- @_] Q	•] 🤻] 😂 🛷 •] 🛬 • 🕸 •	- \$			📑 🕨 Design
Project Explorer 21 📃 🗖	CastomerVehid	ieOrderSubmit.svit 🕞 *ModelSelecti	on 23			° 0
😑 😫 👘 🔻	Tahona					[
 B to But Markels B to But Markels B to Customer Volkrichels B to B to Customer Volkrichels B to B Volksbröcher B to B Volksbröcher 		Sedens Sedens Sedens sar sedens give noner,	¢	Consider, the consider, relability, regard good looks the best of all meetids	Tools Term Task Control Control	*
	Fors Nalidation	Soript				
🚼 Outine 🛙 👘 🐨 🗖 🗖	Properties 23	L Repestory Browser	Deolemania 🕢 Taolia			e ~ = 0
An outline is not available.	Control Propert	ties (Radio) - uxTruckModels				
	General	* 10				
	Atributes	ux/Trud/Modele				
	Cata Dinding Validation	* Tab Order				
	Eventa	a 10				
	Effecta	* Description				
		Service Radio vidget.				×
		Jan multivano moger.				
						x
] 📬						

Under **Data Binding**, you can see that the radio set values are bound to the **TruckModels** Dataslot. When this form is realized, this radio set is populated with the values from TruckModels, which in turn was bound to an output parameter from the Web service call:

Design - ModelSelection - Progres												l ×
File Edit Insert Navigate Search F				- 6 - 4 -							> Design	
Project Explorer 23		ideOrderSubmit.s	avt 📄 *ModelSelectio	n 21							9	0
😑 😫 🖉 🔨	Tahona	2	9	🗹 🗹 🛋 🔍 🗅								
18 딸 Dublithede 19 딸 Outcomer Vahrlehet 18 딸 Outcomer Vahrlehete 18 딸 Outcomer Vahrleheter Vahrlahet 18 딸 Vahrlehoter		sedans gi	Sedens and comfort: ear we you the best value for money.	0	Crossovers Comfort, visibility, rugged good looks: the best of all warfds	c				Form Tasks Layout Controls A Plow Data Sources Prom Progreets	\$	~
	Fors Neldetion		- NT N		1			1	w Task	2 Events		
An outine is not available.			ry browser 🔝 Problems	Bookmans STasks								-
	General	· Choken	- annoce-soaces				• Select	ion Source/Target				
	Attributes Data Binding Validation	Q Service				Citizer,	DetaSkt	TruckModels -Supported Data Types List, String			Gear	
	Eventa Effecta	© DetaSkt	TruckModels Supported Data Types Loli, Map, String			 dear						
		O Adaplet	Properties Name:			re- Kalue						-
] 📬										1		

If I now look at the control's **Events** tab, there's an **onChange** event:

Design - ModelSelection - Progra Re Edit Insert Navigate Search					10
		•] ?] @ # •] > • * • • • • •		📑 🕨 Dec	ign .
		NdeOxderSubsit.svt			- 0
😑 😫 🗇 🏹	Tahona		<u> </u>		
18 등을 Dadhhidd 등을 CastonerOderVelkit R Gr CastonerHeidelCoderSalmit 문 등 Velkelonder		Setions Setions due you we he set relate for ensary	Constant Constant, registring, registring, registring, search of eff wards	Z tolo Term Table Term Table Term Table Constance Trans Table Torm Table Torm Table Torm Table Torm Table Tabl	*
	Fors Kalidation				
E Outine IX 🔐 🖓 🚥 🗖		🚹 Repestory browser 👔 Problems 🗳 Bookmarks 🗧	Tasks		- e v -
	General Advibubes Deta Dinding Validation Events Diffects	ortics (Readio) - unfruckéhodels	, and actions that are associated with a field in the form.		Add Remove Vew Action
				I	

Selecting one of the radio set values fires this event. Selecting **View Action** again shows two actions defined:

🛔 Design - ModelSelection - Progress			_IO ×
File Edit Insert Nanigate Search P			
		•] ?] @ ? •] 5 • 5 • 5 • 6 • • •	🖹 🕨 Design
		ideCiderSubalt out 🕥 "ModelSelection (2)	° 0)
😑 😫 🕼 🝸	Actions Code		
8 W DuidWhide 8 W OustomerOrderVehicle	Actions acTruck	Kodels_on/Dange	
8 Gr CustomerVehicleOvderSubmit 8 WehdeOrder	Copy the va	Aur of widget (unTrudHodels) to (totHodel)	- × •
	Execute the	Following JavaScript:	÷ #
		OtherSelections('ucTruckEodels')J	
	2 68950	pHodelHame('urTruckHodels');	
	-		
	Fam Taldation		
	Properties 33	Le Repestory Browser 🔝 Problems 🗳 Bookmarks 🕢 Tasks	e = = 0
An outline is not available.	Control Prope	rties (Radio) - unTruckModels	
	General	* Description	Resare
	Atributes	The Events subtab enables you to define events, functions, and actions that are associated with a field in the form.	Ramova
	Data Binding Validation	Eventa Name	Vew Action
	Events	onChange unTruckPlodels_enChange anClub.	
	Difecto	arFoos arBy	
0.			

The first, highlighted as "Copy the value of widget...", defined through the kind of dropdowns and expression choices I showed you earlier, tells Savvion to copy the value of the **uxTruckModels** control to another variable called **txtModel**.

The second action is actually a bit of handwritten JavaScript code, executing two user-defined functions, both of which take the chosen radio set value as input:

<pre>clearOtherSelections('uxTruckModels');</pre>	
assignModelName('uxTruckModels');	

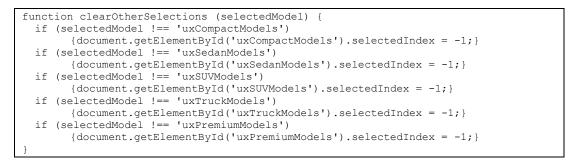
If I select the **Code** tab I can actually look at the code for those functions. Here's the first, **assignModelName**:

```
// Sets the value of the ModelName dataslot via a field
function assignModelName (srcModel) {
  var s = sbm.widgets.getValue(srcModel);
  var obj = eval(s);
  for(var i = 0; i < obj.length; i++) {
    var e = obj[i];
    if (e.selected) {sbm.widgets.setValue('txtModelName', e.label);}
  }
}</pre>
```

<pre>interligies is in the second sec</pre>	Design - ModelSelection - Progre			치미
<pre>characterization constrained on the first and provide on the first</pre>	File Edit Insert Navigate Search	Project Run Tools Wi	dow Help	
Intel Legen 1: Inter the second control of the formation of th	🔁 • 🖽 🛆 🛱 👪 😆	• 🐷 🛯 💁 • 🛛	7 0 4 • 1 2 • 4 • • • • • •	📑 🕨 Design
<pre>Updated by Updated by Update</pre>	🏷 Project Explorer 🕄 📃 📟 🗆	Customer Vehide Cr	Sersiubanit ave 👔 👔 ModelSelection (2)	-
<pre>iii Outcome derivation iii Outcome deriv</pre>	😑 😫 👘 🖤	Actions Cade		
Served Attributed Description Description Participation * Description Percent Participation Description Description Participation New Enclose Enclose Enclose Enclose New Enclose Enclose Enclose New Enclose Enclose Enclose New Enclose Enclose Enclose New Enclose Enclose Enclose New Enclose Enclose Enclose New Enclose Enclose Enclose New Enclose Enclose Enclose Enclose New Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose Enclose En	은 UPC Contrainer Order Melicie 또 UPC Contrainer Melicie Order Submit 문 UPC Melicie Order	Function and vor s = ob vor s = ob to (vor i if (c.s) if (c.s	<pre>ignbotizes (reflect) (wid(r); wid(r); wid(r); vis(r); vis(r);</pre>	
		Attributes T Data Binding Bin Validation Bin Events or Diffects of	er Dverbandska kradikan poulos define eventa, functiona, end actions that are associated with a field in the form. ender in there define unbrackfielded, prochaege Colo Forat	Ensure

The object-oriented view of the data makes this look a bit complicated, but basically the code has to walk through all the possible values of the radio set control as a JSON object like the example superimposed on the screenshot above, and find the one whose selected property is set. That choice's **label** property in turn is written into a simple string variable called **txtModelName**. Note that this code is using another Savvion library called **sbm.widgets**, which you can find in the same appendix I showed you earlier in the **BPM Studio Users Guide**.

The other function clears any previous selection that was made for another model type, such as Sedan or Crossover, by setting its selectedIndex property to minus 1.



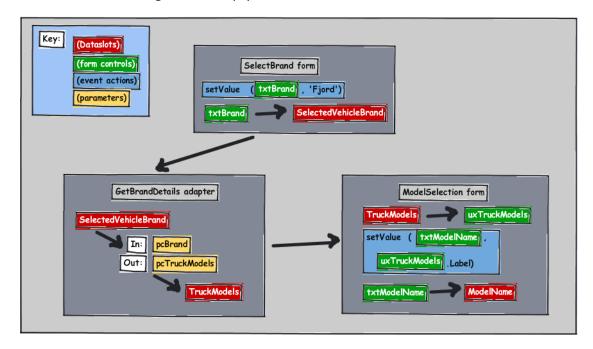
This is in case the customer changed his mind and selected one model type and then another. You can see that the code uses the standard HTML convention of **document.getElementById** to locate each radio set control within the form.

So overall, looking at the event handlers for just these steps has introduced you to several of the different ways you can define event handlers for Savvion forms, sometimes just using predefined categories of expression types that are offered to you, sometimes by writing your own JavaScript code.

Finally, the screenshot below shows the two hidden fields that this form uses. Doubleclick on the first one, which is bound to the **SelectedVehicleModel** dataslot, you see that this is the one named **txtModel**, referenced in the code we just looked at:

🗈 - 🖂 🛆 🔛 🔝 - 💭 🚳 I	🖕 +] 🦹 🖕 🖉 +] 5 + 3 + 4 - 4 - 4 - 4	📑 🕨 Design
	WildeOxderSizbait.put	
e giji Dukthede Gjiji OutowerOrdertelicle Giji OutowerOrdertelicle Giji OutowerOrdertelicle Giji VehicleOrder	Sofer and complete soften for management of the soften for management of t	X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X X
c outire 53 == = =	Tracks African	Lannak A Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Texture Te
a	Statut dynamical fields	trais
Form Valida	on Script	

The second one, bound to the **ModelName** dataslot -- which in turn is passed on to a later step in the process -- is the one named **txtModelName**.



One final look at a diagram to help you understand the data flow here:

This user form **ModelSelection** starts with the **TruckModels** dataslot value, a JSON object holding a list of all the Fjord trucks, and uses it to set the radio set for the new form. It then sets the hidden field **txtModelName** to the label of the selected model, which in turn is mapped to the dataslot **ModelName**, which is used in a later step in the flow.

My goal in this paper has been to show you just enough about the very powerful ability to pass data values from one step in a process to another, and to define flexible event handlers for control events, to get you started. As with all of these videos and papers, you should take what you've been introduced to here and follow up on your own in your investigation of all the kinds of ways you can use data retrieved from your application to populate and control the steps in your Savvion process. This can enable you to build an application process model that at runtime is

closely tied to all the data and logic that's already built into your OpenEdge ABL application.