PROGRESS EXCHANGE 3 DISCOVER. DEVELOP. DELIVER.

Corticon Business Rules for OpenEdge



Contents

Business Rules for OpenEdge	Error! Bookmark not defined.
Progress Exchange 2013	Error! Bookmark not defined.
Introduction	3
Exercise 1 – Model Corticon Decision	3
Step 1 – Create Vocabulary	3
Step 2 – Model Business Rules in Rulesheet Editor	
Step 3 – Create Tests	
Step 4 – Deploy Decision Service	23
Step 5 – Test Deployed Decision Service	
Exercise 2 – Integrate Corticon Decision Service	
Step 1 – Configure the Environment	
Step 2 – Add "Using" References	
Step 3 – Add Variable Definitions	
Step 4 – Add Decision Service Execution Code	
Step 5 – Run OpenEdge Application	40
Exercise 3 – Update Deployed Decision Service	43
Bonus activities	45
Exercise 1 – Enhance the Vocabulary	45
Exercise 1 – Configure Server Console for Monitoring	45
Exercise 2 – Use Code Macros	47
Exercise 2 – Externalize to Standalone.p file	
Exercise 3 – Add new Rulesheets to Ruleflow	
Exercise 3 – Deploy Multiple Versions	48
Exercise 3 – Deploy with Effective Dates	50



Introduction

This document outlines the exercises that you will complete as part of the workshop. It guides you step by step through building a simple Corticon decision service and integrating this with an OpenEdge application. No previous knowledge of Corticon is required. However, a basic understanding of OpenEdge ABL, Temp-Tables and DataSets is advisable.

A basic OpenEdge application is provided as a starting point. The end product of the workshop will be decision logic deployed and running in the Corticon Server and integrated with the OpenEdge application.

Exercise 1 – Model Corticon Decision

A Corticon decision model is comprised of a Vocabulary (data model), and sets of Rules against that model. The Vocabulary can be built from scratch inside Corticon Studio, but it can also be generated from existing artifacts. OpenEdge 11.3 provides the ability to create a Corticon Vocabulary from existing data structures like Temp-Tables and Pro DataSets. In this exercise we will create the Vocabulary from a Temp-Table within the provided application. Once we have the Vocabulary, we will model the business rules and use the analysis tools in Corticon Studio to ensure accuracy and completeness. If there are any defects, we will resolve them. Finally we will use test data to execute the rules, and see the outcome of a given decision.

When we are satisfied with the decision model, we will use the publish wizard to deploy the service to the Corticon Server. This will make the decision available as a callable service from any application via SOAP, as well as through the built in Corticon integration with OpenEdge 11.3.

Step 1 – Create Vocabulary

Launch Progress Developer Studio for OpenEdge from the desktop. If you are prompted to select a Workspace, you should use the following location C:\Progress\Corticon_5.3\OE_workspace. You will see the "Corticon Business Rules" project opened. Double click Underwriting.i to reveal the contents of a pre-defined temp-table (Figure 1)





Figure 1

Right mouse click on Underwriting.i and select Export/Business Rules Vocabulary Definition (Figure 2).





Figure 2

This will launch the export wizard where you can specify to temp-tables and datasets you wish to export, as well as the location of the exported file. (Figure 3)

You should choose the **Applicant Temp-Table** for export. You can choose any location and name for the export definition file. Be sure to remember your choices, as you will use this file to import into Corticon Studio in a later step.



Export Business Rules Vocabulary Definition							
Select ProDataSets and Temp-Tables							
Workspace Resources 🚊 🕀 📼 💌	ProDataSets and Temp-Tables						
type filter text	type filter text						
Corticon Business Rules Corticon Business Rules SampleApp.cls Content of Cont	Image: Name Image: Name Image: Name Image: Name						
Selected data structures:	*						
Definition file: C:\Users\chogan\Desktop	<pre>>/Underwriting.i) \export.brvd Browse</pre>						
?	Finish Cancel						



Now Launch Corticon Studio from the desktop. If you are prompted to select a Workspace, you should use the following location C:\Progress\Corticon_5.3\Corticon_workspace. You should see the project "Progress Exchange 2013" opened (Figure 4)



P Corticon Designer - Progress Developer Studio	
<u>F</u> ile <u>E</u> dit <u>N</u> avigate Se <u>a</u> rch <u>P</u> roject <u>R</u> un <u>W</u>	indow <u>H</u> elp
📬 🕶 🖻 🤷 🗸 🗸 🗸	🖹 🔣 Corticon Des
🕼 Rule Vocabulary 🛛 🔓 🍟 🗖	
Project Explo 🛛 🔊 Rule Operat 🖵 🗖	Image: Statements I
Progress Exchange 2013	
Documents	
 Kuleflow Rulesheet 	
Ruletest	
Vocabulary	
□ [◆] 0 items selected	

Figure 4

Right mouse click on the project and select Import (Figure 5). This will launch the import wizard. From here you should choose "Business Rule Vocabulary Definition" (Figure 6). Next you should select the file that you exported in the previous steps (Figure 7). And finally you will specify the name and location in the Corticon project for the imported vocabulary file. You should choose the **Vocabulary** folder, and name the file **Underwriting** (Figure 8). You should now see **Underwriting.ecore** in the Project Explorer. (Figure 9).





Figure 5



P Import	
Select Import Business Rules Vocabulary Definition to workspace.	Ľ
Select an import source:	
type filter text	
 General General Plug-in Development Progress Corticon Business Rules Vocabulary Definition Rule Modeling Run/Debug Team 	
Sack Next > Finish	Cancel

Figure 6





Page | 9

📔 Business Rules Vocabulary	Shared 💿 💌
Create new Vocabulary file Specify the file name and location of the new Vocabulary fil	e.
Enter or select the parent folder:	
Progress Exchange 2013/Vocabulary	
 Image: A state of the state of the	
File name: Underwriting Advanced >>	
Seck Next > Finish	Cancel



P Corticon Designer - Progress Developer Studio		
<u>F</u> ile <u>E</u> dit <u>N</u> avigate Se <u>a</u> rch <u>P</u> roject <u>R</u> un	<u>W</u> indow <u>H</u> elp	
📑 • 🖬 🖻 💁 • 🛷 •	II (II (II (II (II (II (II (II (II	Corticon Des
💭 Rule Vocabulary 🛛 🗧 🏹 🗖 🗖		
Project Expl 🛛 🥔 Rule Operat 🖓 🖬	📄 Rule Statements 🖾 Rule Messages 🔀 🔲 Properties 🖹 Problems	<u> </u>
Progress Exchange 2013		
> Documents		
🗁 Ruleflow		
➢ Rulesheet		
🔁 Ruletest		

Figure 9

Step 2 – Model Business Rules in Rulesheet Editor

To create a new Rulesheet, right mouse click on the project and select New/Rulesheet (Figure 10). Choose the "Rulesheet" folder and name it **RiskRating** (Figure 11). You are then prompted to specify the Vocabulary you want to use for this Rulesheet. Select the **Underwriting.ecore** file that you previously created (Figure 12). You are now presented with a blank Rulesheet, ready to use for modeling our rules (Figure 13).



P Corticon Designer - Progress Developer Studio						
<u>F</u> ile <u>E</u> dit <u>N</u> avigate Se <u>a</u> rch <u>P</u> roject <u>R</u> un <u>W</u> in	idow <u>H</u> elp					
📬 🕶 🕼 🖕 🖋 🗸						😭 🌆 Corticon Des
💟 Rule Vocabulary 🛛 📲 🏹 🗖 🗖						- 8
🕒 Project Expl 🛛 🥔 Rule Operat 🖵 🗖 📋	Rule Statements	🖾 Ri	ile Messages 🖾 🚦	Properties	Problems	
Progress Extension 2012			1			
Docum New	+	Ľ	Project			
👂 🗁 Ruleflov 🛛 Go Into		F ♥	File			
Ruleshe	Ctrl+C		Folder			
A Car Vocabu	Ctrl+V	_				
🕼 Unc 💥 Delete	Delete	100 Ter1	Rule Project			
Move		1	Rule Vocabulary			
Rename	F2 [10	Ruleflow			
		50	Rulesheet			
Import		1	Ruletest			
Export		2	Other	Ctrl+N		
(a) Defrech	55	-				

Figure 10



P Create New Rulesheet		x
Rulesheet Create a new Rulesheet		
Enter or select the parent folder:		
Progress Exchange 2013/Rulesheet		
 Image: Image: Progress Exchange 2013 Image: Documents Image: Ruleflow Image: Rulesheet Image: Ruletest Image: Vocabulary 		
File name: RiskRating Advanced >>		
< Back Next > Finish	Cance	21

Figure 11



P Create New Rulesheet	
Rulesheet Select Vocabulary	
 Image: Image: Progress Exchange 2013 Image: Documents Image: Ruleflow Image: Rulesheet Image: Ruletest Image: Progress Exchange 2013 Image: Rulesheet Ima	
< Back Next > Finish	Cancel

Figure 12



P Corticon Designer - C:\Data\Projects\Progress	Exchar	ige 2013\Rule	esheet\RiskR	ating.ers - Progr	ess Developer Studio				- 0 <mark>- X -</mark>
<u>File Edit Navigate Search Project Run</u>	Rule <u>s</u> h	eet <u>W</u> indow	v <u>H</u> elp						
	E (🗞 ؼ 📚	000	0 🐁 🕀	Q 🔗 -			🖹 📑 🕼	Corticon Des
🕼 Rule Vocabulary 🛛 🗧 🗖 🗖	😼 R	iskRating.ers	23						- 8
🗊 Underwriting		Conditions			0	1	2	3	4 🔺
🔜 Applicant	a							1	
📑 age	b								
ing gender	d								
isSkydiver	e								
a name	f								-
iskBating		Actions							+ +
state		Post Messag	e(s)						
	A								
	C C								
	D								
	E								-
Project Expl 23 Project Expl				0	verrides				
	R	ule Statement	ts 🖾 🖂 R	ule Messages 🛽	Properties Problems				- 8
Progress Exchange 2013	Re	ef ID	Post	Alias	Text				*
Buleflow									E
Rulesheet									
RiskRating.ers									
🗁 Ruletest									
🗁 Vocabulary									
🖏 Underwriting.ecore									
				"					
							1		r

Figure 13

To model the rules, first type in the Rule Statements (Figure 14). Then you assign the conditions and actions that make up the two rules in this example. For the first rule, drag the "isSkydiver" attribute over into the first row of the Conditions area. Then drag "riskRating" to the first row of the Actions area (Figure 14). In Column 1, select "T" as the value for the Condition and type in "High Risk" as the value for the Action. (Figure 15).

For the second rule, drag "age" attribute to the second row in the Condition area. In Colum two type in "<35" for the Condition value, and type in "Low Risk" for the action value (Figure 16).

Finally, link together the Rule Statements with the declarative representation, but typing in the corresponding column number. You can also choose to post the Rule Statements as shown (Figure 17).



Projects\Pr	ogress Ex	change 2	013\Ruleshee	et\RiskRating.	ers - Progress D	eveloper Studio			
<u>File E</u> dit <u>N</u> avigate Se <u>a</u> rch <u>P</u> roject	<u>R</u> un Ru	ule <u>s</u> heet	Window H	lelp					
		1 🔊 🌢	Ø 😣 🛛 🖸	000	🗞 🕀 🛛 😋	⊾ - 🔗 -		📑 🐻 Ca	orticon Des
🔋 🖉 Rule Vocabulary 🛛 🔓 🍸 🗖 🗖	🐻 *Ris	kRating.er	rs 🔀						
💭 Underwriting	C	onditions			0	1	2		3 🔺
Applicant		pplicant.i	sSkydiver						
age .	В								
gender	d								
	e								
j policyPrice	A	ctions			< III				- F - A
📑 riskRating	Po	ost Messa	ge(s)						
🧾 state	A A	pplicant.r	iskRating		_				
	C								
	D								-
Project 🛛 🥔 Rule Op 🗖 🗖				Overri	les				
	🗋 Rule	Statemer	nts 🖾 🖾 R	ule Messages	Properties	🖹 Problems			
Progress Exchange 2013	Ref	ID	Post	Alias	Text				
🗁 Documents					Applicants w	ho skydive have a Hig oder 25 bave a Low Pi	gh Risk rating		E
🗁 Ruleflow					Applicants u	Idel 55 Have a Low Id	isk fatility		
🗁 Rulesheet									
Be RiskRating.ers									
🛛 😼 Underwriting.ecore									
	•	1	III						Þ
□ □ ◆									

Figure 14

Corticon Designer - C:\Data\Projects\Pro	ogress	Exchange	2013\Ruleshe	et\RiskRating.e	ers - Progress (Deve[\per S	itudio			×
<u>File Edit Navigate Search Project</u>	<u>R</u> un	Kule <u>s</u> heet	Window H	lelp						
📑 📬 🐨 🔚 🔞 👜 🔚 🥦 🚯 🔞 🖾 🕼 🐼 🍰 🌲 🛇 🛇 🛇 🛇 🔍 🖷 🧏 🔻 🔗 🔻 📑 📴 Corticon Des										
じ Rule Vocabulary 🛛 🏣 🏹 🗖 🗖	1	RiskRating.ers 🛛								
Underwriting		Condition	s		0		1	2	3	•
Applicant	a	Applicant.	isSkydiver				Т			
📑 age	b									
🧱 gender	d									
isSkydiver	e									
name		Actions			٠ III					+ +
rickBating		Post Mess	age(s)							
	Α	Applicant.	riskRating				'High Risk'			
	B									
					_					— . II
	Ĕ			Overrid	les l					
Project 🛛 💣 Rule Op							I			
	R	ule Stateme	ents 🛛 🔟 H	lule Messages	Properties	Probl	ems			
🔯 Progress Exchange 2013	Re	ef ID	Post	Alias	Text	L . L P	Laura I Patri Dia			
🗁 Documents					Applicants v	vno skydive inder 35 hav	e nave a High Kis ve a Low Risk rat	k rating		=
🔁 Ruleflow					, pp. comes c					
Rulesheet										
Bo RiskKating.ers										
Cabulary	<u> </u>									
Underwriting.ecore										-
	•	1								•



Page | 16

🔋 Corticon Designer - C:\Data\Projects\Progress Exchange 2013\Rulesheet\RiskRating.ers - Progress Developer Studio									×	
<u>F</u> ile <u>E</u> dit <u>N</u> avigate Se <u>a</u> rch <u>P</u> roject	Eile <u>E</u> dit <u>N</u> avigate Se <u>a</u> rch <u>P</u> roject <u>R</u> un Rule <u>s</u> heet <u>W</u> indow <u>H</u> elp ^나 장									
📫 📬 🖛 📾 🔚 🖼 💀 🐼 😥 🖾 🖉 🖉 🖉 🖉 🖉 🖉 🖉 🖉 🐨 🖓 🐨 🖆 🔛 🚱 🔛 🖾 🖾 🖾 🖾 🖾 🖾 🖉										
🕼 Rule Vocabulary 🛛 🔚 🍸 🗖 🗖	🐻 *Ris	*RiskRating.ers 🛛 🗖 🗖								
Underwriting	0	Conditions			0	1	2	3	•	
📑 Applicant	a A	Applicant.is	sSkydiver			Т				
📑 age		Applicant.a	ge		_		< 35		- 11	
🧱 gender	d				-					
isSkydiver	e				_				-	
name	4	Actions			٠ III	1			+ +	
riskRating	F	ost Messa	ge(s)							
state	A A	Applicant.r	iskRating			'High Risk'	'Low Risk'		_	
	B								- 11	
	D				_					
				Overrid	es				_	
Project 🛛 💣 Rule Op		- 0		de Marcana	Descetion 🛛	P. Desklaure				
E 🕏 🎽	Kul	e statemer		ule iviessages						
🔯 Progress Exchange 2013	Ket	ID	Post	Alias	Lext Applicants who	shudiwe have a High Rig	sk rating		- âl	
🔁 Documents					Applicants unde	er 35 have a Low Risk ra	ting			
Ruleflow							-			
RiskRating ers									_	
Ruletest		_								
🦢 Vocabulary										
Vnderwriting.ecore										
	•		III						F	
□\$										

Figure 16







Now we can check for any flaws in the rule model, such as conflicts and incompleteness. This means there are rules that overlap, or there a gaps in the logic and not enough rules to cover all possible scenarios. Conflicts can be resolved by using overrides, and incompleteness is handled by adding additional rules as suggested.



🐻 R	iskRating.ers 🛛					
	Conditions	0	1	2	3	•
а	Applicant.isSkydiver		Т	-		
b	Applicant.age		-	< 35		
с						
d		6	Continue Ch		x	
e		P	rogress Corticon Stu			
t						
	Actions	٠ III	— —	Constant and the		
	Post Message(s)			flicts detected: 1		
Α	Applicant.riskRating					
В						
C				OK		
D				OK		
E						
F	Override	s				

Page | 18



💀 R	iskRating.ers 🛛						
	Conditions	0	1	2	3	4	
а	Applicant.isSkydiver		Т	-	F		
b	Applicant.age		-	< 35	>= 35		
с							
d							
e							
f							
- 1							•
	Actions	▲ III					- F - A
	Post Message(s)						
Α	Applicant.riskRating		'High Risk'	'Low Risk'	'Medium Risk'		
В							
С							
D							
E							
F							T
	Overrides		2				

Step 3 – Create Tests

Right click on the Project and select New/Ruletest. Name it **Applicant** and save it in the Ruletest folder. You will then choose the Rulesheet that you want to test against. Select the RiskRating Rulesheet you have just created (Figure 18).

You can now drag the Applicant from the Vocabulary area and drop into the Input section. Double click on the age attribute and type in 25. The double click isSkydiver and choose "false". You can provide a name to identify this applicant, although it is not necessary for the decision to execute. You may also delete the other attributes for clarity's sake if desired (Figure 19).

Once complete, press the "Execute" button to run the test and see the outcome of the decision, as well as the corresponding Rule Statements for every rule that fired (Figure 20, 21).



P Create New Ruletest	
Ruletest	
Select Test Subject	
Local	
a 🔯 Progress Exchange 2013	
🗁 Documents	
🗁 Ruleflow	=
🔺 🗁 Rulesheet	
RiskRating.ers	
🗁 Ruletest	-
A Manakada a	
Remote Servers	
http://localhost:8082/axis	
Update List	
Permete Decision Services	
< Back Next > Finish	Cancel

Figure 18



🔋 Corticon Designer - Progress Exchange 2013/Ruletest/Applicant 🔭 - Progress Developer Studio										
<u>File Edit N</u> avigate Se <u>a</u> rch <u>P</u> roject	<u>F</u> ile <u>E</u> dit <u>N</u> avigate Se <u>a</u> rch <u>P</u> roject <u>R</u> un Rule <u>t</u> est <u>W</u> indow <u>H</u> elp									
📅 🕶 🔄 🕼 🗁 🕼 🕼 🖉 🖓 🔐 🖏 👘 👘 👘 🖓 🖓 🖓 🖓 🗛 🖓 💀 👘 👘 🖓 🖓 🖓 🖓 👘 👘 👘 🖓 👘										
じ Rule Vocabulary 🛛 🏪 🍸 🗖 🗖	Not the second s	Applicant.ert 🕅 🗖 🗖								
Underwriting	untitled_1									
a Applicant	file:/C:/Data/Projects/Progress Exchange 20	13/Rulesheet/RiskRating.ers								
age	Input	Output	Expected							
isSkydiver	Applicant [1]									
	🔳 age [25]									
policyPrice	📑 gender									
🤜 riskRating	isSkydiver [false]									
📑 state	name									
	riskRating									
	state									
Project 🛛 🥔 Rule Op 🗖 🗖										
□ 🔄 🏹	📄 Rule Statements 🖾 Rule Messages 🛛	🔲 Properties 🚼 Problems	- 8							
🔯 Progress Exchange 2013	Severity Message		Entity							
🗁 Documents										
🗁 Ruleflow										
RickRating erc										
Prostatinglers										
Applicant.ert										
🗁 Vocabulary										
🔯 Underwriting.ecore										
			1							

Figure 19



P Corticon Designer - Progress Exchange 2013/Ru	lletest/Applicant.er	t - Progress Developer	Studio	Shared 📃 🗖 🗾 🔀
File Edit Navigate Search Project Run	Ruletest Window	r Help		
📑 🗝 🗟 🙆 🔊 🖉 😭 🗞	à 🖻 🖉 🖓	S 🖗 🗞 🔛 🗲	🗟 🕨 🥌 💁 🖌 🛷 🔻	😭 🔣 Corticon Des
💭 Rule Vocabulary 🛛 🧯 🍟 🗖	🔞 Applicant.ert	×		- 8
Underwriting	📄 untitled_1			
📑 Applicant	file:/C:/Data/Pro	jects/Progress Exchang	ge 2013/Rulesheet/RiskRating.ers	
		Input	Output	Expected
	📃 Applicar	it [1]		
	🔳 age [25]		
	📑 isSky	diver [false] • [Mika]		
🎦 🎦 Project Expl 🔀 🧬 Rule Operat 🖵 🗖		e [IVIIKE]		
□ 🔄 🍒 🏹				
🕼 Progress Exchange 2013				
➢ Documents				
Bulesheet	Rule Statemer	nts 🕅 Rule Messages	😚 🔲 Properties 💽 Problems	
Auletest				
🗁 Vocabulary	Severity	Message		Entity
4 III +				
□◆				

Figure 20

🏠 Applicant.ert	X					
untitled_1						
file:/C:/Data/Proje	ects/Progress Exchange/	Rulesheet/RiskRating.ers				
	Input	Output	Expected			
Applicant age [2	: [1] 5] liver [false] [Mike]	 Applicant [1] age [25] isSkydiver [false] name [Mike] riskRating [Low Risk] 				
Rule Statemen	ts 🖾 Rule Messages 🛽	3 🔲 Properties				
Severity	Message		Entity			
Info	Applicants under 35 have a Low Risk rating Applicant[1]					

Figure 21



Step 4 – Deploy Decision Service

A decision service is a collection of one or more Rulesheets orchestrated in a Ruleflow. To deploy a decision service, first create a new Ruleflow by right mouse clicking on the project and selecting New/Ruleflow. Name the Ruleflow **NewPolicyProcessing (**Figure 21). Next select the Vocabulary to use for this Ruleflow. Choose the Underwriting.ecore that you created earlier (Figure 23). Finally, drag the RiskRating.ers Rulesheet from the Project Explorer on to the Ruleflow canvas.

P Create New Rule Flow	
Rule Flow Create a new Rule Flow	
Enter or select the parent folder:	
Progress Exchange 2013/Ruleflow	
 Image: Image: Progress Exchange 2013 Image: Documents Image: Documents Image: Documents Image: NewPolicyProcessing 	
Advanced >> < Back Next > Finish	Cancel

Figure 22





Figure 23



P Corticon Designer - Progress Exchange	2013/Ruleflow/NewPolicyProcessing.erf - Progress Developer Studio	
<u>File Edit N</u> avigate Se <u>a</u> rch <u>P</u> roject	Rule <u>f</u> low <u>R</u> un <u>W</u> indow <u>H</u> elp	
100	% 👻 🗛 👻	😭 🔣 Corticon Des
🕼 Rule Vocabulary 🛛 🏭 🏹 🗖 🗖	🔞 NewPolicyProcessing.erf 🛛	- 8
 Underwriting Applicant Project 23 Rule Op □ Se Progress Exchange 2013 Documents Ruleflow Ruleflow Rulesheet RiskRating.ers Ruletest Applicant.ert Vocabulary Underwriting.ecore 	RiskRating RiskRating Rule Statements Rule Messages Properties Problems	 Palette Palette Q → Connection Rulesheet Iterative Subflow Service Call-out

Figure 24

The decision service is completed and ready to be deployed to the Corticon Server. Start the Corticon Server from the desktop, and ensure that the application server is running (Figure 25, 26). In Corticon Studio, right mouse click on the project and select "Publish". Provide the connection details to the Corticon Server you want to publish to (Figure 28).

URL: http://localhost:8082/axis

User: admin

Password: admin

Select the "NewPolicyProcessing" flow and Finish (Figure 29).





Figure 25



Figure 26





📔 Corticon Design	ier - I	Progress Exchange 2013/Ru	leshe	et/RiskRating.ers	- Progres	s Develop	per Studio				Sh	ared	
File Edit Navig	File Edit Navigate Search Project Run Rulesheet Window Help												
1 - 8 6	📫 🕆 🔛 🗟 🔚 🗔 🚯 🔞 🐼 🔕 🖉 🔕 🕼 🛇 🛇 🛇 🕲 🗄 🗛 🔻 🔗 🕶												
💭 Rule Vocabulai	🕼 Rule Vocabulary 🛛 🗧 🎽 🌆 🏧 🖓 Applicant.ecore 🔤 RiskRating.ers 🖾 👘 🖓 🖓												
🔰 Applicant				Conditions			1		2	3	4	5	
📑 Applica	ant		a	Applicant.isSky	diver		T		-	{F, null}			
			b	Applicant.age			-		< 35	{>= 35, null}			
			d										_
			e										
			-	Actions			•		III				
				Post Message(s	;)			1					
Broject Evel	52	🥔 Rula Operat 🛛 🗖 🗖	A	Applicant.riskR	ating		'High	Risk'	'Low Risk'	'Medium Risk'			
C Project Expl	~ [C										
<u> </u>			D		_								
Progress E		New		+	0	Overrides	2						
🕞 Rulefic		Go Into			3 🖂 F	Rule Mess	ages 🔲 P	ropertie	🕄 Problems				
> 🗁 Rulesh		Conv		Ctrl+C	st	Alias	-2	Text					
D > > Rulete:	曲	Paste		Ctrl+V	0	Appli	cant	Applic	ants who skydive	have a High Risk ratir	ng		
👂 🗁 Vocabi	~	Delete		Delete	0	Appli	cant	Applic	ants under 35 hav	e a Low Risk rating	diana minta		_
	^	Move		Delete	0	Арри	cant	Аррис	ants over 30 who	don tiskydive are iviei	alum Kisk		-
		Panama		E2									
		Netrattie		12	-	_		_					
	è	Import			-								
	4	Export											
	8	Refresh		F5	-	_							
		Close Project											
		Close Unrelated Projects											
		Dublish			-	_		_					
	-	Publish											-
<		Pup Ac				1							۱.
📑 🗘 🔂 F		Null As											
		Debug As		•									

Publish Decisio	n Services	
Server Connecti Specify the Serve		
Server		
Server URL:	http://localhost:8082/axis	-
Username:	admin	
Password:	*****	
		Test Connection
	< Back Next > Finish	Cancel

Figure 28



P Publish Decision Services								
Decision Services Select the decision services to be published. To update existing decision services, check Republish.								
Select All Deselect All					🔲 Republish			
Decision Service Name	Version	Status	Database Mode	Return Entities	Ruleflow			
NewPolicyProcessing	1.0	New	None	All	file:/C:/Data/Pro			
					Ţ			
		< Bacl	c Next >	Finish	Cancel			

Figure 29

To access the Corticon Server Console and confirm the service is deployed, launch the web browser from the desktop and go to the URL: <u>http://localhost:8082/axis</u>

User: admin

Password: admin

Once logged in you can access the deployed decision services (Figure 30), and see a list of all decisions and various details. You should see v1.0 of NewPolicyProcessing, along with some other default decision services (Figure 31).









Firefox 🔻										• ×
Corticon Technologies		+								
🗲 🔶 🏫 國 nbwfhchog	an :8082/a	xis/			☆ ⊽ C ^e	8 - Goog	gle 🔎			¥-∻-
Progress. C	ortico	n.					The Po	wer o	f Deci	sions™
Business Rules Server C Deployed Decision Serv	onsole ices	-> Home								
Service Name Version	Live	Effective	e Expires	5 Deploye	ed from CDD	Dynamic Reload	: I Execution:	1 5 (Avg Time ms)	Clear Stats
AllocateTrade	1.14	\checkmark			Yes	Yes	0		0	Clear
Candidates	1.14	\checkmark			Yes	Yes	0		0	Clear
NewPolicyProcessing	1.0	V			No/Remove	No	0		0	Clear
ProcessOrder	1.10	\checkmark			Yes	No	0		0	Clear
						8	0	%	M	#



Step 5 – Test Deployed Decision Service

Corticon Studio allows you to test against remote decision services. To do see, return to the Ruletest you created, and update the tested asset. Double click on the file path of the RiskRating.ers file (Figure 32), choose "Remote Servers", click the "Update List" button, and select NewPolicyProcessing in the list (Figure 33).

Now when you execute the test, Corticon Studio will create a SOAP message, post it to the remote URL and display the result payload from the deployed decision service (Figure 34).



🔯 Applicant.ert 🙁		
untitled_1		
file:/C:/Data/Projects/Progress Exchang	ge/Rulesheet/RiskRating.ers	
Input	Output	Expected
 Applicant [1] age [25] isSkydiver [false] name [Mike] 	 Applicant [1] age [25] isSkydiver [false] name [Mike] riskRating [Low Risk] 	



P Select Test Subject		×
Local		
🕼 Progress Exchange 2013		•
🗁 Documents		
🔁 Ruleflow		=
🗁 Rulesheet		
RiskRating.ers		
🕞 Ruletest		-
Remote Servers		
http://localhost:8082/axis		
Update List		
Remote Decision Services		
AllocateTrade		*
Candidates		=
NewPolicyProcessing		-
	OK Canc	el

Figure 33



🔯 Applicant.ert 🛛		
untitled_1		
http://localhost:8082/axis?NewPolicy	Processing	
Input	Output	Expected
 Applicant [1] age [25] isSkydiver [false] name [Mike] 	 Applicant [1] age [25] isSkydiver [false] name [Mike] riskRating [Low Risk] 	



Exercise 2 – Integrate Corticon Decision Service

In this exercise we will use the new features of OpenEdge 11.3 to integrate the deployed decision service with an existing application. Because the decision service vocabulary was created from existing Temp-Tables, these can be seamlessly used as the data payload, with no mapping or transformation necessary.

Step 1 – Configure the Environment

Launch Developer Studio for OpenEdge from the desktop, if it is not already running. To leverage the built in connectivity with Corticon in OE 11.3 you will need to add one External Directory, and one External Library to your project.

Right mouse click on the "Corticon Business Rules" project and select Properties. Navigate to Progress OpenEdge\PROPATH .

Click "Add External Library" and select "C:\Progress\OpenEdge\gui\rules\OpenEdge.BusinessRule.pl" (Figure 35)

Click "Add External Directory" and select "C:\Progress\OpenEdge\gui\rules" (Figure 36)

You should now see the 2 components in the Properties view (Figure 37).



Figure 35



PR	OPATH Directory Selection	x
	Select a directory to add to the PROPATH.	
	⊳ 퉲 adm2	^
	🛛 🔐 prodict	
	퉬 prohelp	
	🍌 protools	
	⊿ 🔑 rules	
	4 퉬 OpenEdge	
	🌗 BusinessRules	=
	🖻 퉲 web	
	▷ 퉲 web2	
	🌗 webutil	
	🛛 퉲 Help	
	🌗 include	
	D 퉲 install	
	D 🔝 iava	-
	Folder: rules	
	Make New Folder OK Canc	el

Figure 36





Step 2 – Add "Using" References

In the Project Explorer, double click on SampleApp.cls to bring up the visual editor for the application form (Figure 38). To view and edit the code associated with the form, right mouse click on the form and select "View Source", or press F9 (Figure 39).

ampleApp.cls (Design) 🛛 🖸 SampleApp	o.cls		
_		Toolbox	
🚽 Progress Exchange 2013		🖭 Microsoft Contro	ols
		E OpenEdge Cont	rols
Name		🗄 OpenEdge Ultra	Controls
Age			
Skydiver			
B ' 1 B 4			
Risk Rating			
	Ĩ		
Execute			
listBox1			
	b		

Figure 38



P OpenEdge Visual Designer - Corticon Busine	ss Rules/SampleApp.cls - Progress Developer Studio	
<u>File Edit Source Navigate Search Proje</u>	ct <u>R</u> un <u>O</u> penEdge <u>W</u> indow <u>H</u> elp	
📑 🕶 🔛 🗟 🚔 🗞 🕶 🗛 🕶 🚱	▼ 🔗 ▼ 🗈 🛱 OpenEdge Vi	鑙 OpenEdge E
🎦 Project Explorer 🛛 🗖 🖏 *Sa	ampleApp.cls (Design)	- 8
		*
🚰 Corticon Business Rules	File · FirstForm	
Referenced Assemblies	Purpose :	
Procedure Libraries	Syntax :	
X assemblies.xml	Description :	E
C SampleApp.cls	Created : Mon Apr 08 13:12:08 EDT 2013	
X SampleApp.resx	Notes :	
i Underwriting.i	*/	
	JSING Progress.Windows.Form. CLASS SampleApp INHERITS Form: DEFINE PRIVATE VARIABLE chkSkydiver AS System.Windows.Forms.CheckBox NG DEFINE PRIVATE VARIABLE listBox1 AS System.Windows.Forms.ListBox NG DEFINE PRIVATE VARIABLE button1 AS System.Windows.Forms.Button NG DEFINE PRIVATE VARIABLE components AS System.Windows.Forms.Label NG DEFINE PRIVATE VARIABLE label3 AS System.Windows.Forms.Label NG DEFINE PRIVATE VARIABLE label1 AS System.Windows.Forms.Label NG DEFINE PRIVATE VARIABLE label1 AS System.Windows.Forms.Label NG DEFINE PRIVATE VARIABLE txRiskRating AS System.Windows.Forms.TextBox NG DEFINE PRIVATE VARIABLE txtName AS System.Windows.Forms.TextBox NG DEFINE PRIVATE VARIABLE txtName AS System.Windows.Forms.TextBox NG DEFINE PRIVATE VARIABLE txtBale AS System.Windows.Forms.TextBox NG DEFINE PRIVATE VARIABLE txtBale AS System.Windows.Forms.TextBox NG DEFINE PRIVATE VARIABLE txtBale AS System.Windows.Forms.TextBox NG DEFINE VARIABLE Options AS JsonObject NG	D-UNDO. D-UNDO. D-UNDO. D-UNDO. D-UNDO. D-UNDO. D-UNDO. D-UNDO. D-UNDO. D-UNDO. D-UNDO. D-UNDO.
	DEFINE VARIABLE OF A A RULESSERVERCONNECTION AMERICEN NO DEFINE VARIABLE OCONNECTION AS RulesServerConnection NO	D-UNDO.
	DEFINE VARIABLE decisionService AS DecisionService NG	D-UNDO.
	DEFINE VARIABLE dVersion AS DECIMAL NO	D-UNDO. 👻
	<	
	Writable Insert 15:31 Preprng on	

Figure 39

The first step is to add the USING references as shown in Figure 40. The code snippet is provided so you can copy and paste into your project.

```
/*Begin code snippet*/
USING OpenEdge.BusinessRules.DecisionService.
USING OpenEdge.BusinessRules.RulesServerConnectionParameters.
USING OpenEdge.BusinessRules.RulesServerConnection.
USING Progress.Json.ObjectModel.JsonObject.
USING Progress.Lang.AppError.
/*End code snippet*/
```









Step 3 – Add Variable Definitions

Next you will add the reference to the Include files and Variable definitions as show in Figure 41. The code snippet is provided so you can copy and paste into your project.

```
/*Begin code snippet*/
{Underwriting.i}
{OpenEdge/BusinessRules/ttRulesMessage.i}
```

DEFINE	VARIABLE	oOptions	AS	JsonObject	NO-UNDO.
DEFINE	VARIABLE	oParams	AS	RulesServerConnectionParameters	NO-UNDO.
DEFINE	VARIABLE	oConnection	AS	RulesServerConnection	NO-UNDO.
DEFINE	VARIABLE	decisionService	AS	DecisionService	NO-UNDO.
DEFINE	VARIABLE	dVersion	AS	DECIMAL	NO-UNDO.
DEFINE	VARIABLE	cServiceName	AS	CHARACTER	NO-UNDO.
/*End code	snippet*/				



DEFINEPRIVATEVARIABLEchkSkydiver chkSkydiverASSystem.Windows.Forms.CheckBoxNO-UNDO.DEFINEPRIVATEVARIABLElistBox1ASSystem.Windows.Forms.ListBoxNO-UNDO.DEFINEPRIVATEVARIABLEbutton1ASSystem.Windows.Forms.ButtonNO-UNDO.DEFINEPRIVATEVARIABLEcomponentsASSystem.ComponentModel.IContainerNO-UNDO.DEFINEPRIVATEVARIABLElabel3ASSystem.Windows.Forms.LabelNO-UNDO.DEFINEPRIVATEVARIABLElabel2ASSystem.Windows.Forms.LabelNO-UNDO.DEFINEPRIVATEVARIABLElabel1ASSystem.Windows.Forms.LabelNO-UNDO.DEFINEPRIVATEVARIABLEtxRiskRatingASSystem.Windows.Forms.TextBoxNO-UNDO.DEFINEPRIVATEVARIABLEtxRiskRatingASSystem.Windows.Forms.TextBoxNO-UNDO.DEFINEPRIVATEVARIABLEcoptionsASSystem.Windows.Forms.TextBoxNO-UNDO.DEFINEVARIABLEoOptionsASSystem.Windows.Forms.TextBoxNO-UNDO.DEFINEVARIABLEoOptionsASSonObjectNO-UNDO.DEFINEVARIABLEoOptionsASRulesServerConnectionParametersNO-UNDO.DEFINEVARIABLEoConnectionASRulesServerConnectionNO-UNDO.	{Underwriting.i} {OpenEdge/BusinessRules/ttRulesMessage.i}	}		
DEFINE VARIABLE oOptions AS JsonObject NO-UNDO. DEFINE VARIABLE oParams AS RulesServerConnectionParameters NO-UNDO. DEFINE VARIABLE oConnection AS RulesServerConnection NO-UNDO.	DEFINE PRIVATE VARIABLE chkSkydiver DEFINE PRIVATE VARIABLE listBox1 DEFINE PRIVATE VARIABLE listBox1 DEFINE PRIVATE VARIABLE button1 DEFINE PRIVATE VARIABLE components DEFINE PRIVATE VARIABLE label3 DEFINE PRIVATE VARIABLE label2 DEFINE PRIVATE VARIABLE label1 DEFINE PRIVATE VARIABLE txtRiskRating DEFINE PRIVATE VARIABLE txtName DEFINE PRIVATE VARIABLE txtAge A	AS System.Windows.Forms.CheckBox AS System.Windows.Forms.ListBox AS System.Windows.Forms.Button AS System.ComponentModel.IContainer AS System.Windows.Forms.Label AS System.Windows.Forms.Label AS System.Windows.Forms.TextBox AS System.Windows.Forms.TextBox AS System.Windows.Forms.TextBox AS System.Windows.Forms.TextBox	NO-UNDO. NO-UNDO. NO-UNDO. NO-UNDO. NO-UNDO. NO-UNDO. NO-UNDO. NO-UNDO. NO-UNDO.	
DEFINE VARIABLE decisionService AS DecisionService NO-UNDO. DEFINE VARIABLE dVersion AS DECIMAL NO-UNDO. DEFINE VARIABLE cServiceName AS CHARACTER NO-UNDO.	DEFINE VARIABLE oOptions A DEFINE VARIABLE oParams A DEFINE VARIABLE oConnection A DEFINE VARIABLE decisionService A DEFINE VARIABLE dVersion A DEFINE VARIABLE dVersion A	AS JsonObject AS RulesServerConnectionParameters AS RulesServerConnection AS DecisionService AS DECIMAL AS CHARACTER	NO-UNDO. NO-UNDO. NO-UNDO. NO-UNDO. NO-UNDO. NO-UNDO.	

Step 4 – Add Decision Service Execution Code

In this step you will add the code to execute the decision service when the "Execute" button is clicked. The code will create a connection to the Corticon Server. It will create a record in the temp-table based on the values entered in the user interface. It will execute the decision service and the update the form with the results of the decision (Figure 42).

The code snippet is provided so you can copy and paste into your project.

```
/*Begin code snippet*/
oOptions = NEW JsonObject().
oOptions:Add ('URL', 'http://localhost:8082').
oParams = NEW RulesServerConnectionParameters(oOptions).
oConnection = NEW RulesServerConnection(oParams).
cServiceName = 'NewPolicyProcessing'.
decisionService = NEW DecisionService(oConnection, cServiceName).
EMPTY TEMP-TABLE Applicant.
CREATE Applicant.
Applicant.age = INTEGER(THIS-OBJECT:txtAge:Text).
Applicant.isSkydiver = LOGICAL(THIS-OBJECT:chkSkydiver:Checked).
decisionService:InvokeService(INPUT-OUTPUT TABLE Applicant BY-REFERENCE).
```



```
FIND LAST Applicant.
THIS-OBJECT:txtRiskRating:Text = Applicant.riskRating.
/* get the messages */
decisionService:GetMessages(OUTPUT table RulesMessage).
```

```
listBox1:Items:Clear().
FOR EACH RulesMessage:
    listBox1:Items:Add(RulesMessage.MessageText).
END.
/*End code snippet*/
```

```
🎳 SampleApp.cls (Design) 🖾 🖻 SampleApp.cls 🔀
                                                                                                 ۰
        Purpose:
        Notes:
                            -----*/
       WisualDesigner.
       METHOD PRIVATE VOID button1_Click( INPUT sender AS System.Object. INPUT e AS System.EventArgs
           oOptions = NEW JsonObject().
           oOptions:Add ('URL', 'http://localhost:8082').
           oParams = NEW RulesServerConnectionParameters(oOptions).
           oConnection = NEW RulesServerConnection(oParams).
           cServiceName = 'NewPolicyProcessing'.
           decisionService = NEW DecisionService(oConnection, cServiceName).
           CREATE Applicant.
           Applicant.age = INTEGER(THIS-OBJECT:txtAge:Text).
                                                                                                 Ξ
            Applicant.isSkydiver = LOGICAL(THIS-OBJECT:chkSkydiver:Checked).
           decisionService:InvokeService(INPUT-OUTPUT TABLE Applicant BY-REFERENCE).
           FIND LAST Applicant.
           THIS-OBJECT:txtRiskRating:Text = Applicant.riskRating.
           RETURN.
    ₹ .
                                               111
```

Figure 42

Step 5 – Run OpenEdge Application

Now we will run the application. From Developer Studio, click the Run button and select "Run As\Progress OpenEdge Application" (Figure 43). This will execute your app at the form will be shown (Figure 44).

Input values for Name, Age, Skydiver and click the Execute button. Based up the values you provided you will see output similar to Figure 45. You can change the values and Execute again to get various answers from the decision.



P OpenEdge Visual Designer - Corticon	Business Rules/SampleApp.cls - Progress Developer Studio	
<u>File Edit N</u> avigate Se <u>a</u> rch <u>P</u> roject	<u>R</u> un <u>O</u> penEdge <u>D</u> esign <u>W</u> indow <u>H</u> elp	
🔁 🕶 🗟 🖆 🗞 🔻 📀	- 🔽 - 🖉 - 🗈 🖽 🖾	penEdge Vi 🗳 OpenEdge E
Project Explorer 🛛 🗖	(no launch history)	
□ <u></u> □ <u></u>	Run As I Progress Mobile Application	olbox 🕂
😤 Corticon Business Rules	Run Configurations 2 Progress Open dge Application Alt+Shift+X, 0	G Aicrosoft Controls
Referenced Assemblies	Organize Favorites	OpenEdge Controls
x assemblies.xml	Name	OpenEdge Ultra Controls
SampleApp.cls	Age	
Underwriting.i	Skydiver Risk Rating Execute IistBox1	
[◆]]	Writable	

Figure 43

Name Age Skydiver Risk Rating
Age Skydiver
Age Skydiver
Skydiver Risk Rating
Risk Rating
Execute





 Progress Exchan	ge 2013	
Name	Mike Jones	
Age	34	
	Skydiver	
Risk Rating	Low Risk	
	Execute	
Applicants under Version 1.0 of [er 35 have a Low Risk rating Decision Service "NewPolicyProces	sing" was invoked

Figure 45



Exercise 3 – Update Deployed Decision Service

In this exercise, while leaving the OpenEdge application running, you make a change the decision service you previously modeled. Once you re-deploy, you will see that for the same input values into the OpenEdge application you will get a different answer.

The first step is to launch Corticon Studio, if it is not already running. Update the rules to use 30 as the threshold age. Make sure to update the Rule Statements as well (Figure 46). Next, right mouse click on the Project and launch the Publish wizard again. Make sure to check the "Republish" option to overwrite an existing version (Figure 47).

Finally, click the execute button on your OpenEdge app and see the new results (Figure 48).

R R	iskRating.ers	83						
	Conditions				0	1	2	3
а	Applicant.is	Skydiver				Т	-	F
b	Applicant.a	ge				-	< 30	>= 30
c								
d								
e								
t								
	Actions			•				
	Post Messa	ge(s)						
Α	Applicant.r	iskRating				'Medium Risk'		
B								
C								
E		0				2		
		000	errides			2		
R	ule Statemer	nts 🖾 🖾 Ri	ule Mes	sages	Properties			
Re	ef ID	Post	Alias		Text			
1		Info	Appli	cant	Applicants w	ho skydive have	a High Risk rating	
2		Info	Appli	cant	Applicants ur	nder 30 have a Lo	ow Risk rating	
3		Info	Appli	cant	Applicants 30) and over who d	on't skydive are M	edium Risk

Figure 46



Publish Decision Services							x
Decision Services Select the decision services to be pu	ıblished. To	update exist	ing decision services	, check Republish	ı.		
Select All Deselect All]					🔽 Republis	sh
Decision Service Name	Version	Status	Database Mode	Return Entities	Ruleflow		
NewPolicyProcessing	1.0	Update	None	All	file:/C:/Dat	a/Projects/	-
							4
			Back Nex	t > Fi	nish	Cancel	

🖳 Progress Exchan	ge 2013	
Name	Mike Jones	
Age	34	
	Skydiver	
Risk Rating	Medium Risk	
	Execute	
Applicants 30 a Version 1.0 of [nd over who don't skydive are M Decision Service "NewPolicyProc	ledium Risk cessing'' was invoked

Figure 48



Bonus activities

If you complete the basic lab exercises before the allotted time, you may wish to look at some addition examples of functionality.

Exercise 1 – Enhance the Vocabulary

In developer Studio for OE, update Underwriting.i to add new fields the temp-table. You can also create an additional temp-table and define a relationship with a Dataset. Export the .brvd file and re-import into Corticon Studio. You should see the updated vocabulary. If you used a Dataset, you will see an association between the Vocabulary entities

Exercise 1 – Configure Server Console for Monitoring

Click "Configure Rules Server" icon from toolbar. Ensure all options are "Yes" under Decision Service Options (Figure 49). Go to list of Decision Services and select NewPolicyProcessing. Drill down to Service Configuration and add Applicant.riskRating as a Monitored Attribute (Figure 50). For all future executions of this service, Corticon will capture statistics for the attribute that you can review (figure 51, 52).



Figure 49



Firefox 🔻							• X
Corticon Technologies	+						
🗲 🔶 🏦 🛞 nbwfhchogan:808	2/axis/	⊽ C'	<mark>8</mark> ▼ Google		۹ م		**** *
🕑 Progress. Cortic	con.				The Power	of Dec	isions™
Business Rules Server Conso Decision Service Details	le -> Home -> Decisior	n Service	25				
Create New Version	Modify Current Version	Downl	load Rule Models				
Overview	Service Configura	tion	Rules Report	Test	Execution		
Current Rule Asset URL: Rule Asset URL: Minimum Pool Size: Maximum Pool Size: XML Message Style: Update No Monitored Attributes	Registered		./C11379618921149/NewPoli BrowseNo file selected. 11 Auto-detect ▼	cyProcessing_1	/0_11.eds		
Monitored Attribute ApplicantriskRating Add		Analysis	<u>s Bucket</u>				E
				8 🕑	H 3	s 🗠	1 #

Figure 50

Firefox *									
Conticon Technologies	an 8082/avis/	Ŧ		<i></i> √	Cal 🛛 🛪 Google			0 [
Progress C	orticon.			~				The Power of	Decisions™
Business Rules Server (Deployed Decision Serv	Console -> rices	Home							
Service Name	/ersion	Live	Effective	Expires	Deployed from CDD	Dynamic Reload	Executions	Avg Time (ms)	Clear Stats
AllocateTrade	1.14	1			Yes	Yes	0	0	Clear
Candidates	1.14	1			Yes	Yes	0	0	Clear
NewPolicyProcessing	1.0	1			No/Remove	No	%	1	Clear
ProcessOrder	1.10	\checkmark			Yes	No	827	0	Clear
							× 🕑	÷1 %	딸 🛖



Firefox 🔻						N			
Corticon Technologies		+				2			
🗲 🔶 🏦 🎯 nbwfhch	iogan:8082/axis/		7 7 C ⁱ	<mark>8</mark> ▼ Google			۹ 🖡		╞╼╶╔╝
Progress.	Corticon.					The	ower	of Decis	ions™
Business Rules Server Decision Service Perf	r Console -> H ormance & Res	lome -> Decision Ser ults Distribution	vices						
Decision Service: New Version: 1.0	wPolicyProcessi	ing							
Performance		Distribution Chart							:
Applicant.riskRating									
Value	Quantity	Percentage					1		
High Risk	3	37.5		<nul></nul>					
Low Risk	3	37.5		Medium		High			
Medium Risk	1	12.5		Risk		Risk			1
<null></null>	1	12.5							
Total Change	5 8			Low Risk					
					2	🕑 🐳	*	<u>_</u>	#



Exercise 2 – Use Code Macros

There are 3 default macros provided to assist with code insertion.

BR-CONNECT BR-GETMSG BR-INVOKE

The templates for these are editable and found in System Preferences (Figure 53). You can experiment using these macros to automatically insert the code that you copy/pasted in the previous exercise.



oe filter text	Templates				⇔ • ⇒ •
Install/Update	Create, edit or remove	e templates:			
Java			D. T. C.		New
Java EE	Name	Context	Description		INEW
JavaScript	✓ &FR	OpenEdge Editor	Frame name		Edit
Plug-in Development	✓ &GL	OpenEdge Editor	Global define		
Progress Customer Sup	✓ &SC	OpenEdge Editor	Scoped define		Remove
Progress Databases	BR-CONNECT	OpenEdge Editor	Business Rules:		
Progress OpenEdge	BR-GETMSG	OpenEdge Editor	Business Rules:		Restore Remove
AnnBuilder	BR-INVOKE	OpenEdge Editor	Business Rules: I		Deverte Defer
RPM	CATCH	OpenEdge Editor	Catch block		Revert to Derau
Business Rules	CMT	OpenEdge Editor	Comment		Transit
Database Connectio	DED	OpenEdge Editor	Event with deleg	-	Import
Debug	1	m	Event with delegin	F I	Export
Editor	Denviewe				
Annotations	Preview:				
Assistance					
Build					
Colors					
Templates (maci					
Meta Catalog	•				Þ
Mobile App Builder	Use code formatte	r			
Server +					
4 11			Restore D	efaults	s Apply

Figure 53

Exercise 2 – Externalize to Standalone.p file

Instead of putting all of the decision service execution code in the Form definition, you can place it in a standalone.p file and then execute that .p file from the Form code. This could be helpful if you want to re-use this code from many locations.

Exercise 3 – Add new Rulesheets to Ruleflow

The current Ruleflow has only one Rulesheet. Real world decisions have many Rulesheets. In this step, create a new Rulesheet called PolicyPricing. Add 3 new rules that set a policy price amount for the three different risk levels High Risk, Low Risk, Medium Risk. You can then add this Rulesheet to your existing NewPolicyProcessing Ruleflow and link the steps together. Now your decision service not only calculates a risk rating, but it also determines the policy price as well.

Exercise 3 – Deploy Multiple Versions

Corticon supports deployment of multiple versions of the same decision service. To create new version, navigate to the Properties panel for the Ruleflow (Figure 52). Update the Major Version to 2 and save the Ruleflow. Now when you publish the Ruleflow you will see to versions deployed on the Corticon Server (Figure 55).



P Corticon Designer - Progress Exchange	2013/Ruleflow/Ne	wPolicyProcessing.erf - Prog	gress Developer Studio		
<u>File Edit Navigate Search Project</u>	Rule <u>f</u> low <u>R</u> un	<u>W</u> indow <u>H</u> elp			
📑 🗝 🗟 🖶 🖺 100)% 🔹	Q 🔗 -			😭 職 Corticon Des
💟 Rule Vocabulary 🛛 🍋 🔽 🗖	NewPolicyPro	cessing.erf 🛛			- 8
Applicant age gender isSkydiver name policyPrice risKRating state		RiskRating	La Ca		Image: Subflow
Project X Project C				- 4	Service Call-out
🔯 Progress Exchange 2013	Rule Stateme	nts 🔛 Rule Messages 🛄	Properties 🛛 🛣 Problems		E * - L
Documents	Ruleflow	Rule Vocabulary:	file:/C:/Data/Projects/Progress	s Exchange 2013/Vocabul	ary/Underwriting.ecoi Browse
NewPolicyProcessing.erf	Comments	Work Document Entity:	-		
Rulesheet RickPating or	Rulers & Grid	Major Version:	2		
➢ Ruletest		Minor Version:	0		
Applicant.ert		Version Label:			
Underwriting.ecore		Effective Date:	//	Time: 0 🌩 0 🗧	0 AM V Clear
		Expiration Date:		Time: 0 🚔 0 🗧	h 0 ▲ AM ▼ Clear
		Total Number of Rules:	3		

Figure 54

Firefox 🔻									- 0 X
Corticon Technologies		+							
🗲 🔶 🏦 🎯 nbwfhchogan:80	82/axis/			⊂ ⊄	<mark>8</mark> ▼ Google			P 🖡 🛛	⊡ - * -≉`
🕑 Progress Corti	con.						т	he Power of	Decisions™
Business Rules Server Conse Deployed Decision Services	ole ->H	Home							
Service Name Version	n Liv	ve E	ffective	Expires	Deployed from CDD	Dynamic Reload	Executions	Avg Time (ms)	Clear Stats
AllocateTrade	.14	\checkmark			Yes	Yes	0	0	Clear
Candidates 1	.14	\checkmark			Yes	Yes	0	0	Clear
NewPolicyProcessing	1.0	\checkmark			No/Remove	No	8	1	Clear
NewPolicyProcessing	2.0	\checkmark			No/Remove	No	0	0	Clear
ProcessOrder 1	.10	\checkmark			Yes	No	0	0	Clear
							23	*1 %	





Exercise 3 – Deploy with Effective Dates

Corticon also support effective dates for versions. This means you can specify what date ranges that a certain version is applicable. Navigate to the properties for the Ruleflow and specify a start and stop date/time (Figure 56). Re-publish the decision service, and you will see that there are now effective dates for the decision service (Figure 57).



Figure 56



Corticon Technologies +									Firefox 🔻
Image: Service Name Version Live Effective Express O Avg Time (ms) Allocate Trade 1.14 ✓ Yes Yes 0 0 NewPolicyProcessing 1.0 ✓ Jan 1, Mar 31, 2014 No/Remove No 0 ProcessOrder 1.10 ✓ Yes No 0 0							+		Corticon Technologies
Progress Corticon. Business Rules Server Console -> Home Deployed Decision Services Service Name Version Live Effective Expires Deployed from CDD Dynamic Reload Avg Time (ms) AllocateTrade 1.14 // Yes Yes 0 0 NewPolicyProcessing 1.0 // No/Remove No 8 1 NewPolicyProcessing 2.0 Jan 1, Mar 31, 2014 No/Remove No 0 0 ProcessOrder 1.10 // Yes No 0 0	• ₽ 🖸 • ★ • 🖗 •	۹ ج		Google	″ C" <mark>8</mark> -	☆ ⊽	xis/	gan: 8082/a	🗲 🔶 🏦 🛞 nbwfhchog
Business Rules Server Console -> Home Deployed Decision Services Live Effective Expires Deployed from CDD Dynamic Reload Avg Time (ms) AllocateTrade 1.14 // Yes Yes 0 0 Candidates 1.14 // Yes Yes 0 0 NewPolicyProcessing 1.0 // No/Remove No 8 1 NewPolicyProcessing 2.0 Jan 1, Mar 31, 2014 No/Remove No 0 0 ProcessOrder 1.10 // Yes No 0 0	ower of Decisions™	The Power					1.	ortico	🥑 Progress. C
Service NameVersionLiveEffectiveExpiresDeployed from CDDDynamic ReloadAvg Time (ms)AllocateTrade1.14IImage: Service Servic							- > Home	Console vices	Business Rules Server (Deployed Decision Serv
AllocateTrade1.14Image: constraint of the systemYesYes00Candidates1.14Image: constraint of the systemYesYesYes00NewPolicyProcessing1.0Image: constraint of the systemNo/RemoveNo81NewPolicyProcessing2.0Image: constraint of the systemNo/RemoveNo00ProcessOrder1.10Image: constraint of the systemYesNo00	vg Time Clear (ms) Stats	mic Avg Tir load Executions (m	Dynamic Reload	loyed from CDD	ires Depl	ffective Exp	Live	sion	Service Name Ver
Candidates1.14VYesYes00NewPolicyProcessing1.0VNo/RemoveNo81NewPolicyProcessing2.0VJan 1, Mar 31, 2014No/RemoveNo00ProcessOrder1.10VYesNo00	0 Clear	es 0	Yes	Yes			\checkmark	1.14	AllocateTrade
NewPolicyProcessing1.0Image: Constraint of the second secon	0 Clear	es 0	Yes	Yes			\checkmark	1.14	Candidates
NewPolicyProcessing 2.0 Jan 1, Mar 31, 2014 No/Remove No 0 0 ProcessOrder 1.10 Image: Constraint of the second s	1 Clear	No 8	No	No/Remove			\checkmark	1.0	NewPolicyProcessing
ProcessOrder 1.10 Ves No 0 0	0 Clear	No O	No	No/Remove	Mar 31, 2014	Jan 1, 2014	\checkmark	2.0	NewPolicyProcessing
	0 Clear	No 0	No	Yes			\checkmark	1.10	ProcessOrder
2 🕒 🖬 🐝 🖂	š. ⊠ ▲	2 0 1 %							

Figure 57

