

# Customizing Your Application, NO PROBLEM!

Chris Hogan  
Principal Systems Engineer  
Progress Software

Carole Miller  
Senior Systems Engineer  
Progress Software

**PROGRESS**  
**EXCHANGE** 2014

# Menu

---

## EAT IN / TAKE-OUT MENU

We now deliver to all 5 boroughs, as well as Hoboken and Jersey City. Click [HERE](#) to place your order online!

Allergy information can be found [HERE](#).

Dietary restriction information can be found [HERE](#).

---

<a href="#">Breakfast</a>	<a href="#">Starters &amp; Sides</a>	<a href="#">Sandwiches</a>
<a href="#">Soups</a>	<a href="#">Grill Favorites</a>	<a href="#">Noshes</a>
<a href="#">Salad Platters</a>	<a href="#">All Day Specials</a>	<a href="#">Beverages</a>
<a href="#">Salads</a>	<a href="#">Traditional Favorites</a>	<a href="#">Desserts</a>

### Breakfast Specials

(served till 11:30 A.M.)

Eggs or Pancakes or French Toast - \$7.95  
with Katz's Sausages - \$9.60  
(Potatoes, Orange Juice, Coffee or Tea Included)

### Bagel, Lox, Cream Cheese

New York on your tongue!  
platter 14.25 on bagel 12.50

Bagel and Cream Cheese 4.50

### All Day Omelets

## When Harry Met Sally – Apple Pie Order

---

"I'd like the chef salad please with the oil and vinegar on the side and the apple pie a la mode."

"Chef and apple a la mode."

"But I'd like the pie heated and I don't want the ice cream on top I want it on the side and I'd like strawberry instead of vanilla if you have it if not then no ice cream just whipped cream but only if it's real, if it's out of a can then nothing."

"Not even the pie?"

"No, just the pie, but then not heated."

– Sally Albright (Meg Ryan), confusing the waitress in a road side cafe because *"Well I just want it the way I want it."* as she states later on

# Sample OpenEdge Application Menu

---

- eCommerce module
  - Web
  - Mobile
  - GUI client
- Sales and Distribution module
- Bill of Materials Module
- Shipping Module
- Materials Planning Module
- Accounting Module

## My Order

- eCommerce Module
  - GUI client
  - Web
  - **NOTE: Need to customize pricing**
  - **NOTE: Need to change Discount Policy**
- Sales and Distribution

# Sample OpenEdge Application Menu

- eCommerce module
  - Web
  - Mobile
  - GUI client
- Sales and Distribution module
- Bill of Materials Module
- Shipping Module
- Materials Planning Module
- Accounting Module

The screenshot shows a menu titled "My Order" with a yellow overlay titled "2nd Order". The "My Order" menu lists "eCommerce" and "Sales and Distribution". The "2nd Order" overlay lists "eCommerce Module" (with sub-items "GUI client", "Mobile", and "NOTE: Need to customize pricing") and "Sales and Distribution" (with sub-item "Shipping Module").

**My Order**

- eCommerce
  - Web
  - Mobile
  - GUI client
- Sales and Distribution
  - Shipping Module

**2nd Order**

- **eCommerce Module**
  - **GUI client**
  - **Mobile**
  - **NOTE: Need to customize pricing**
- **Sales and Distribution**
  - **Shipping Module**

\*DiscountCalc.p

```
72
73
74 /* Total the balances for the customer using the item pricing and appropriate discount*/
75
76 DEFINE INPUT    PARAMETER pcSalesRep      AS CHARACTER    NO-UNDO.
77 DEFINE OUTPUT   PARAMETER pdTotalBalance AS DECIMAL      NO-UNDO.
78 DEFINE OUTPUT   PARAMETER pdAverageBalance AS DECIMAL      NO-UNDO.
79
80
81 FOR EACH Customer WHERE Customer.SalesRep = pcSalesRep:
82     ACCUMULATE Customer.Balance (TOTAL AVERAGE).
83 END.
84
85 /* Return the total balance and average balance. */
86
87 ASSIGN
88     pdTotalBalance = ACCUM TOTAL Customer.Balance
89     pdAverageBalance = ACCUM AVERAGE Customer.Balance.
```

\*DiscountCalc.p

72

73

\*DiscountCalc.p

75

76

77 74 /\* Total the balances for the customer using the item pricing and appropriate discount\*/

78 75

79 76 DEFINE INPUT     PARAMETER pcSalesRep         AS CHARACTER     NO-UNDO.

80 77 DEFINE OUTPUT    PARAMETER pdTotalBalance     AS DECIMAL     NO-UNDO.

81 78 DEFINE OUTPUT    PARAMETER pdAverageBalance   AS DECIMAL     NO-UNDO.

82 79

83 80

84 81 FOR EACH Customer WHERE Customer.SalesRep = pcSalesRep:

85 82     ACCUMULATE Customer.Balance (TOTAL AVERAGE).

86 83     END.

87 84

88 85 /\* Return the total balance and average balance. \*/

89 86

87 87     ASSIGN

88     pdTotalBalance = ACCUM TOTAL Customer.Balance

89     pdAverageBalance = ACCUM AVERAGE Customer.Balance.

\*DiscountCalc.p

72

73

\*DiscountCalc.p

75 72

76 73

77 74 /\* Total the balances for the customer using the item pricing and appropriate discount\*/

78 75

\*DiscountCalc.p

80 77 72

81 78 73

82 79 74 /\* Total the balances for the customer using the item pricing and appropriate discount\*/

83 80 75

84 81 76 DEFINE INPUT     PARAMETER pcSalesRep         AS CHARACTER     NO-UNDO.

85 82 77 DEFINE OUTPUT    PARAMETER pdTotalBalance     AS DECIMAL       NO-UNDO.

86 83 78 DEFINE OUTPUT    PARAMETER pdAverageBalance    AS DECIMAL       NO-UNDO.

87 84 79

88 85 80

89 86 81 FOR EACH Customer WHERE Customer.SalesRep = pcSalesRep:

ACCUMULATE Customer.Balance (TOTAL AVERAGE).

88 83 END.

89 84

85 /\* Return the total balance and average balance. \*/

86

87 ASSIGN

88     pdTotalBalance = ACCUM TOTAL Customer.Balance

89     pdAverageBalance = ACCUM AVERAGE Customer.Balance.



\*DiscountCalc.p

72

73

\*DiscountCalc.p

75 72

76 73

77 74 /\* Total the balances for the customer using the item pricing and appropriate discount\*/

78 75

\*DiscountCalc.p

80 77 72

81 78 73

82 79 74 /\* Total the balances for the customer using the item pricing and appropriate discount\*/

83 80 75

\*DiscountCalc.p

84 81 76 72

85 82 77 73

86 83 78 74 /\* Total the balances for the customer using the item pricing and appropriate discount\*/

87 84 79 75

88 85 80 76 DEFINE INPUT

86 81 77 PARAMETER pcSalesRep AS CHARACTER NO-UNDO.

87 82 78 77 DEFINE OUTPUT PARAMETER pdTotalBalance AS DECIMAL NO-UNDO.

88 83 79 78 DEFINE OUTPUT PARAMETER pdAverageBalance AS DECIMAL NO-UNDO.

89 84 80 79

85 80 80

86 81 81 FOR EACH Customer WHERE Customer.SalesRep = pcSalesRep:

87 82 82 ACCUMULATE Customer.Balance (TOTAL AVERAGE).

88 83 83 END.

89 84 84

85 85 /\* Return the total balance and average balance. \*/

86 86

87 87 ASSIGN

88 88 pdTotalBalance = ACCUM TOTAL Customer.Balance

89 89 pdAverageBalance = ACCUM AVERAGE Customer.Balance.



*Manage  
my Pain*

Manage Your Pain by:  
Externalizing the Business Rules  
When Customizing Your Application  
NO PROBLEM!!!

**PROGRESS**  
**EXCHANGE** 2014

## Corticon enables organizations to make better, faster decisions by automating business rules

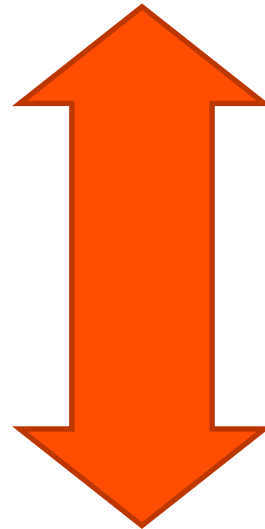
### Should Credit be Extended?



Do not provide credit to delinquent accounts

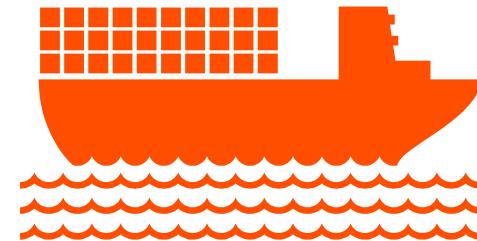
Senior officer approval required for loan amounts greater than \$100,000

### DECISIONS



### RULES

### How to Plan a Shipment?



Hazardous materials must be shipped in double hull tankers

Class 7 super tankers require a minimum berthing distance of 300 ft

# Why Externalize Rules Into Progress Corticon?

---

Model-driven  
Development

Automate rules  
without coding

Ease of use

Guaranteed Rules  
Integrity

Ensure safe changes

Reliable

DeTI (Design Time  
Inferencing) Engine

Market-best performance  
and scalability

Fast

**Design-time and Run-time Integration with OpenEdge**

# Application Scenario

**PROGRESS**  
**EXCHANGE** 2014

# Scenario

---

 **PROGRESS** OpenEdge

OpenEdge  
Application

- Order Processing Application
- Complex Discount Rules for each user

 **PROGRESS** Corticon

Business Rules  
Externalized

- Rules customized in Corticon per customer
- Business Users can define rules

1

One Set of Source  
Code to Manage

- Customizations easy to make
- Upgrades easy to deploy

# Scenario

---

 **PROGRESS** OpenEdge®

OpenEdge  
Application

- Order Processing Application
- Complex Discount Rules for each user

 **PROGRESS** Corticon®

Business Rules  
Externalized

- Rules customized in Corticon per customer
- Business Users can define rules

1

One Set of Source  
Code to Manage

- Customizations easy to make
- Upgrades easy to deploy

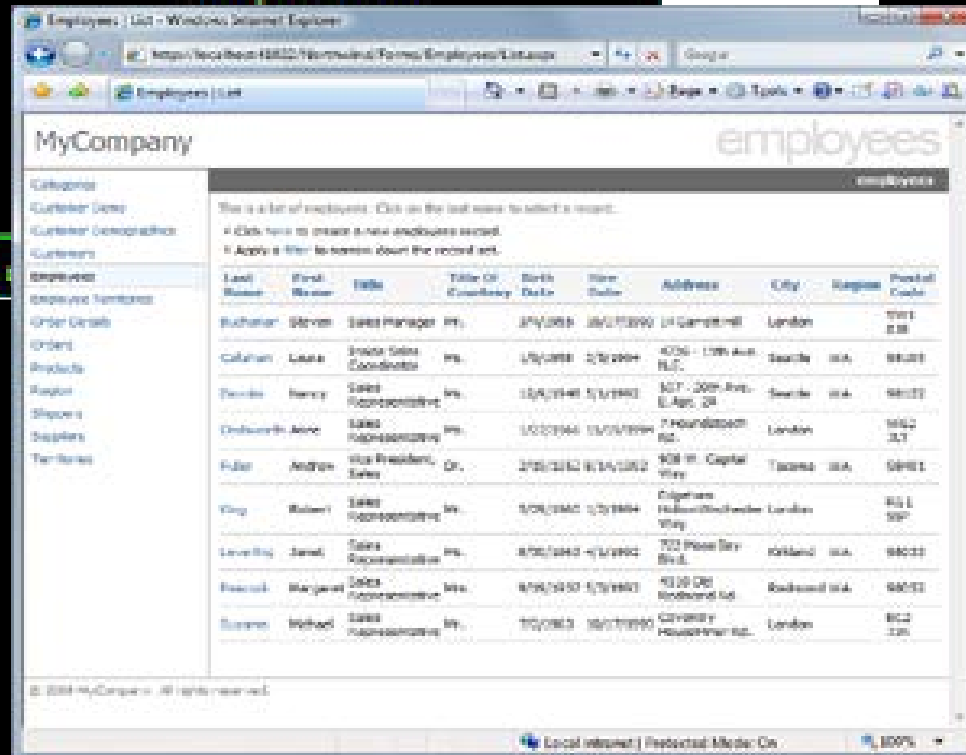




```

DISPLAY CUSTOMER INFORMATION      Area: 010003
-----
Credit Limit: 0      Finance Charge: Y      Area:      Debt Collect: 0
      Billing
Name: A CLEAN WELL LIGHTED PLACE FOR SHOPPING
Address: 401 VAN NESS AVENUE      Address: 401 VAN NESS AVENUE
      :
City: SAN FRANCISCO
State: CA
Zip: 94109
Country: U.S.A
Phone:
  
```

Enter F to skip



All Order Lines

Line Num: 1

Discount: 0%

Price: 4.86

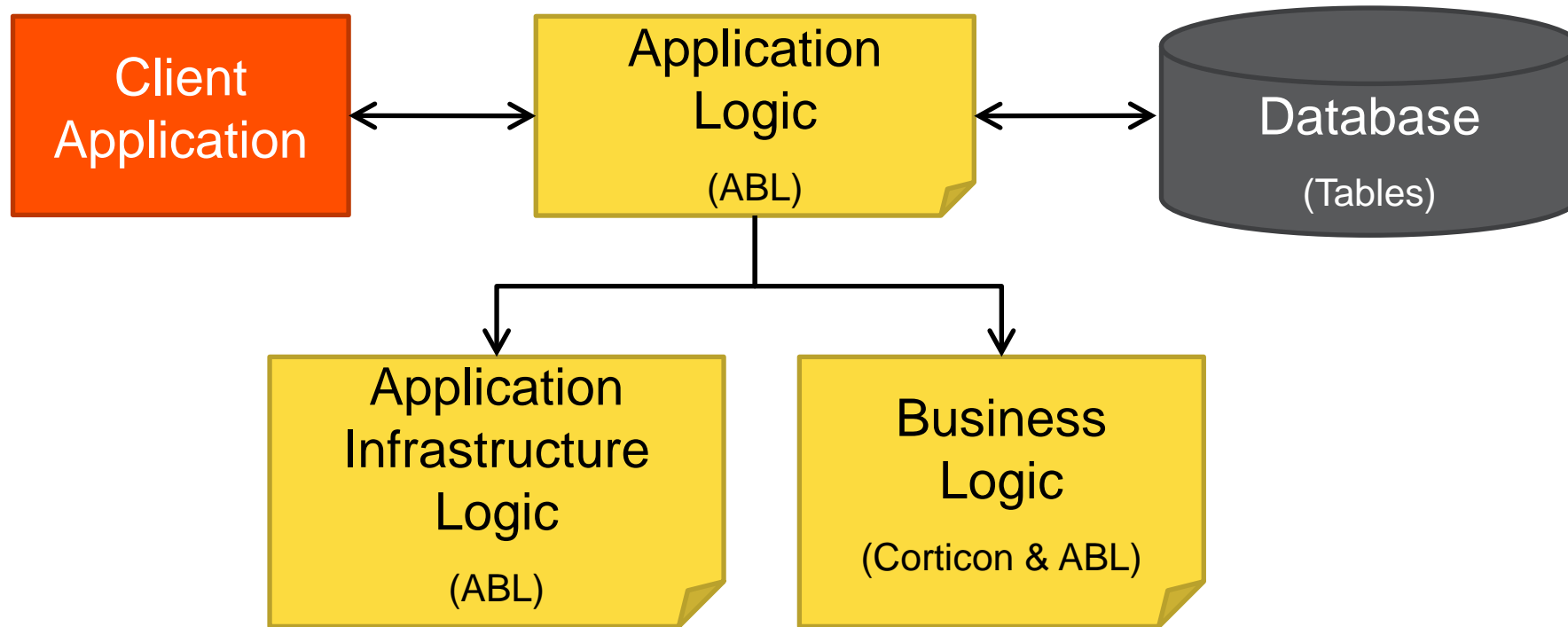
Qty: 30

Do Promo

Navigation buttons: Home, Previous, Next, End

# The Architecture

---



# Progress OpenEdge Application Demo

**PROGRESS**  
**EXCHANGE** 2014

# Scenario

---

 **PROGRESS** OpenEdge<sup>®</sup>  
OpenEdge  
Application

- Order Processing Application
- Complex Discount Rules for each user

 **PROGRESS** Corticon<sup>®</sup>  
Business Rules  
Externalized

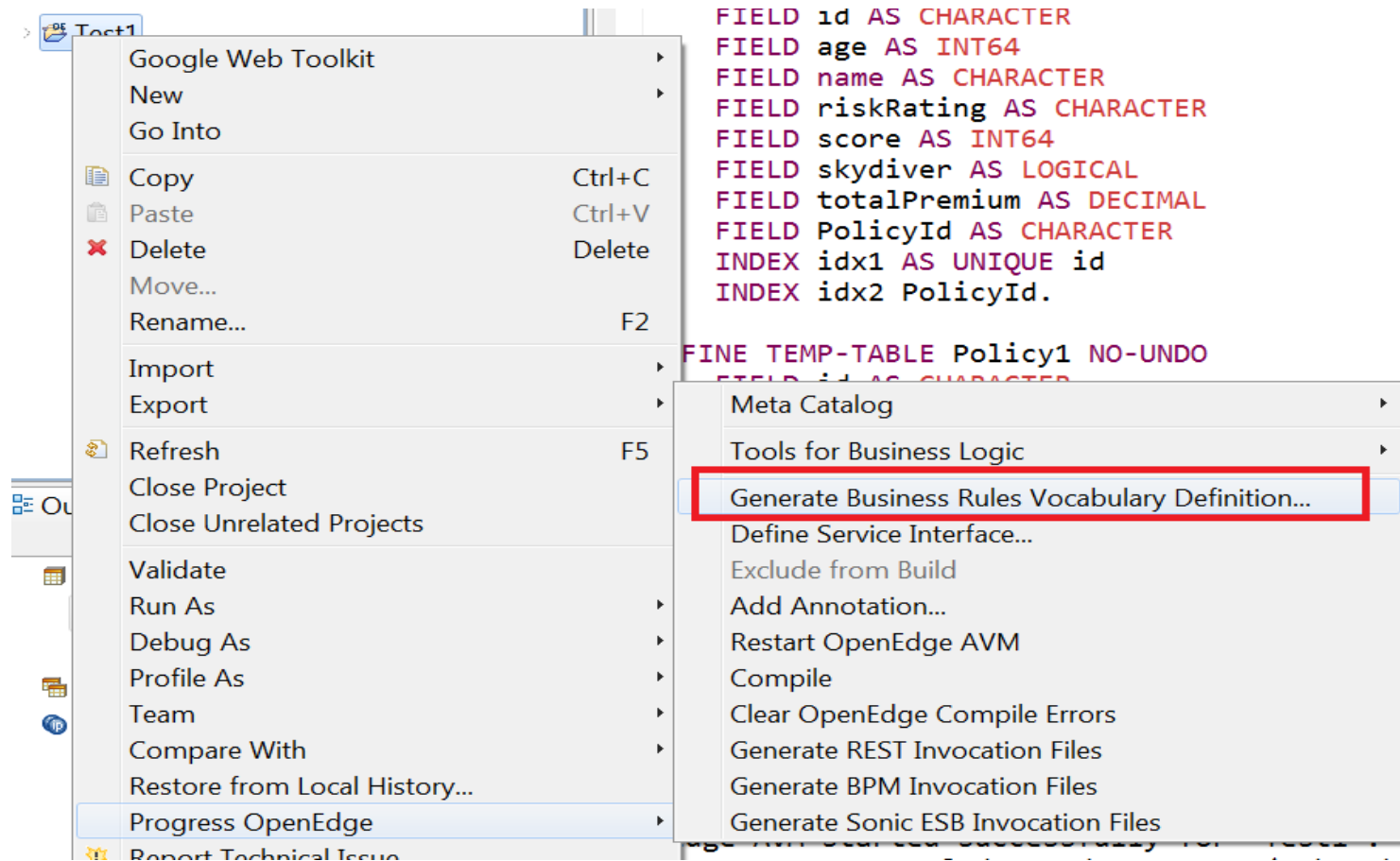
- Rules customized in Corticon per customer
- Business Users can define rules

1  
One Set of Source  
Code to Manage

- Customizations easy to make
- Upgrades easy to deploy

# Progress Developer Studio Integration

- Shared Definition of Vocabulary that rules operate on: import Schema definitions, ProDataSets & Temp tables as Corticon vocabulary Entities



Export Temp Table  
or Prodataset  
to create a  
Corticon vocabulary  
from Progress  
Developer  
Studio

# Model-Driven Rules Development

- Intuitive no-coding approach
  - Empowers Business Analysts
  - Accelerate IT Professionals
- Patented Excel-like modeling tool
  - No programming
  - Business friendly
  - Natural text
- Simple deployment

The screenshot shows a software interface for developing rules. It includes a 'Rule Vocabulary' tree on the left, a central table for defining rule conditions and actions, and a 'Rule Statements' table at the bottom. Three orange callout boxes provide additional context:

- Rules are constructed with drag-and-drop ease**: Points to the central table where conditions and actions are defined.
- Vocabulary consists of your business objects/terms**: Points to the 'Rule Vocabulary' tree on the left.
- Comprehensive, extensible library of operators (akin to Excel functions)**: Points to the 'Rule Operators' tree at the bottom left.
- Rule Statements provide rule documentation and traceability**: Points to the 'Rule Statements' table at the bottom right.

		1	2	3	4	5
a	Patient.risk	'high'	'low'	-		
b	Patient.recommendation	'treat immediately'	'observe'	-		
c	Patient.a1_diagnosis_test	-	-	T		
d	Patient.blood_pressure	-	-	> 130		
e	Patient.cholesterol_level	-	-	> 5		
f						
g						
h						
i						
j						
k						
l						
m						

Actions					
Post Message(s)					
A	Patient.treatment_plan	'low dosage thizide diuretics'	'low dosage beta blockers'	'immediate hospitalisation'	

Alias	Text
Patient	A low dosage of thizide diuretics is required for a patient who is high risk and who needs to be treated immediately.
Patient	A low dosage of beta blockers is required for a patient who is low risk and who needs to be reevaluated over the next two weeks.
Patient	Patient needs to be hospitalised immediately if diagnosis test A1 is positive, blood pressure is over 130 and cholesterol level exceeds 5.

**Proven with the most sophisticated business problems**

# Guaranteed Rule Integrity

Logical loops

Conflicts

Completeness

Progress Corticon Studio

Total conflicts detected: 3

Ref	ID	Type	Source	Description
1	Violation	LoanApplication	Only applicants aged 50 and above can borrow money to purchase a Ferrari.	
2	Info	LoanApplication	Applicants named Carlos are always approved to borrow money to purchase a Ferrari, regardless of any other factors.	
3	Violation	LoanApplication	Applicants may only borrow up to \$2000 for each year of their age.	
4	Violation	LoanApplication	Collateral assets must cover at least 10% of the requested loan amount.	

**Built-in integrity checks ensure creation of accurate, complete and robust rules**

# Add the Progress OpenEdge Business Rules API to the Application

---

```
METHOD PRIVATE VOID button1_Click( INPUT sender AS System.Object, INPUT e AS System.EventArgs ):
```

```
oOptions = NEW JsonObject().  
oOptions.Add ('URL', 'http://localhost:8980').  
oParams = NEW RulesServerConnectionParameters(oOptions).  
oConnection = NEW RulesServerConnection(oParams).
```

```
cServiceName = 'RiskRatingOE'.  
decisionService = NEW DecisionService(oConnection, cServiceName).
```

Establish connection  
with decision service

```
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).
```

Call decision service

```
FIND LAST Applicant.  
THIS-OBJECT:txtRiskRating:Text = Applicant.riskRating.  
/* get the messages */
```

```
decisionService:GetMessages(OUTPUT table RulesMessage).  
listBox1:Items:Clear().
```

Work with the rule  
execution messages

```
FOR EACH RulesMessage:  
listBox1:Items:Add(RulesMessage.MessageText).  
END.
```



Progress OpenEdge  
and Progress Corticon  
Development Demo


**PROGRESS**  
**EXCHANGE** 2014

# Scenario

---

 **PROGRESS** OpenEdge<sup>®</sup>  
OpenEdge  
Application

- Order Processing Application
- Complex Discount Rules for each user

 **PROGRESS** Corticon<sup>®</sup>  
Business Rules  
Externalized

- Rules customized in Corticon per customer
- Business Users can define rules

**1**  
One Set of Source  
Code to Manage

- Customizations easy to make
- Upgrades easy to deploy

# REMEMBER THIS?

\*DiscountCalc.p

72

73

\*DiscountCalc.p

75

76

77 74 /\* Total the balances for the customer using the item pricing and appropriate discount\*/

78 75

79 76

\*DiscountCalc.p

80 77

81 78

82 79 74 /\* Total the balances for the customer using the item pricing and appropriate discount\*/

83 80

84 81

\*DiscountCalc.p

85 82

86 83

87 84 79 74 /\* Total the balances for the customer using the item pricing and appropriate discount\*/

88 85

89 86

76 DEFINE INPUT PARAMETER pcSalesRep AS CHARACTER NO-UNDO.

77 DEFINE OUTPUT PARAMETER pdTotalBalance AS DECIMAL NO-UNDO.

78 DEFINE OUTPUT PARAMETER pdAverageBalance AS DECIMAL NO-UNDO.

79

80

81 FOR EACH Customer WHERE Customer.SalesRep = pcSalesRep:

82 ACCUMULATE Customer.Balance (TOTAL AVERAGE).

83 END.

84

85 /\* Return the total balance and average balance. \*/

86

87 ASSIGN

88 pdTotalBalance = ACCUM TOTAL Customer.Balance

89 pdAverageBalance = ACCUM AVERAGE Customer.Balance.



New Pricing Promotion

New Version of Software

New Government Regulations

New Discounting Rules

Always Customizing  
the App

Customizing... AGAIN!

**PROGRESS**  
**EXCHANGE** 2014

Demo

# Demo Highlights

---

- One OpenEdge Application
- 2 Customers WITH different business Rules for Discounting
- We customized the Rules, not the Application Code
- We upgraded the OpenEdge Application without **losing** the Rules



# Scenario

---

 **PROGRESS** OpenEdge

OpenEdge  
Application

- Order Processing Application
- Complex Discount Rules for each user

 **PROGRESS** Corticon

Business Rules  
Externalized

- Rules customized in Corticon per customer
- Business Users can define rules

1

One Set of Source  
Code to Manage

- Customizations easy to make
- Upgrades easy to deploy

# When Can OpenEdge Applications Benefit From Corticon?

- Is business logic enabling relatively **complex** decisions?
- Is business logic **customized** or **personalized** for each end user?
- Are **changes** frequently required to the application's business logic?
- Is **accuracy** / integrity of logic critical?
- Do you require more **transparency** into how decisions are made?



The more “yes” answers, the greater the benefit from applying Corticon!

# NCAA Case Study: Proven Productivity Benefits of Corticon

Sample ROI Analysis\* – Automation of single, decision-making task

	1 <sup>st</sup> Generation	Corticon	
<b>Initial Development</b>			
Analyze	20 hours	20 hours	
Prepare – Design	5 hours	N/A	
Prepare – Implement	200 hours	2 hours	
<b>Total</b>	<b>225 hours</b>	<b>22 hours</b>	<b>10x Improvement</b>
<b>Change Cycle</b>			
Analyze	15 hours	0.5 hours	
Prepare – Design	7 hours	N/A	
Prepare – Implement	28 hours	1.5 hours	
<b>Total</b>	<b>50 hours</b>	<b>2 hours</b>	<b>25x Improvement</b>



\*Per NCAA customer testimonial

# **PROGRESS EXCHANGE<sup>2014</sup>**

## **Visit the Resource Portal**

- **Get session details & presentation downloads**
- **Complete a survey**
- **Access the latest Progress product literature**

**[www.progress.com/exchange2014](http://www.progress.com/exchange2014)**



**PROGRESS**