

PROGRESS EXCHANGE 2013

DISCOVER. DEVELOP. DELIVER.

Application Modernization Labs

An introductory guide to Modernizing OpenEdge Applications using OpenEdge 11.3

Executive Summary

Using a Sample Legacy Application as the basis, we'll take you through an entertaining guide to let you modernize your application step-by-step and seamlessly use the new features within your existing OpenEdge Application.

In this workshop, you'll learn how to:

- > **Assess** – Carefully assess business and technical state of your application AND your future needs and wishes.
- > **Restructure** – As the basis for any Modernization you first need to make split the layers. No big bang but a phased approach is the only right way.
- > **Insert new technology features** – OpenEdge 11.3 offers a wide range of new features that your application will benefit from. Modernize while you go. Again, no big bang, and your existing legacy application needs to run seamlessly with the newly developed features like OE-Mobile, OE-Corticon and OE-BPM.

Table of Contents

Executive Summary.....	2
Table of Contents	3
Preface.....	4
Introduction.....	5
Chapter 1 – Setting up your environment.....	6
Chapter 2 – Seperating Business Logic	10
Chapter 3 – Implementing OE-Corticon.....	25
Chapter 4 – Implementing OE-Mobile screen.....	51
Chapter 5 – Implementing OE-BPM	75
Appendix A – Getting a Progress login.....	98
Appendix B – Gui Application Manual	100

Preface

This guide is designed to quickly introduce you to the basics of Modernizing Business Applications using the power of OpenEdge 11.3. We'll only cover some of the basic aspects in this guide while showing the true power the newest features can bring.

HOW TO USE THIS GUIDE

This guide is designed to flow from one lesson to the next, with each lesson building on the previous. Starting from a legacy application sample, we will then extract your OpenEdge business logic from the legacy application and afterwards re-use that separated Business Logic in OE-Mobile, OE-Corticon and OE-BPM.

To best leverage this guide, it is recommended you perform the operations as you read them, taking breaks between lessons as required. After you've completed all the lessons, the guide will make for a practical reference that you can dive into at any point to refresh yourself on the concepts.

CONVENTIONS USED IN THIS GUIDE

This document uses the following conventions to distinguish elements of text:

- > **Bold** – Indicates new terms and titles of commands
- > ***Bold italics*** – Notes and tips that alert you to important information.
- > **Bold Courier font** – Text that you should type.
- > *Footnotes* – Expand on details and used to define terminology

Introduction

DESCRIPTION OF THE SAMPLE LEGACY APPLICATION

For Legacy Application Manual see Appendix B.

Chapter 1 – Setting up your environment

Getting started with Progress Arcade

WHAT IS PROGRESS ARCADE?

Progress® Arcade™ is a web portal where you can deploy and manage Progress applications in a Cloud-based environment. You can use Arcade to migrate existing Progress applications to a public Cloud environment, run and test the application in the Cloud, and then deploy the application. Deployment includes the ability to host demos or to publish applications that are available to subscribers.

Progress Arcade is also a site where you can interact with other Arcade users, and find out about services offered by various Progress Technology Partners.

HOW DO I ACCESS THE FEATURES OF PROGRESS ARCADE?

The features of Progress Arcade are grouped into functional areas, represented by six panels on the Arcade home page. The panels include:

- > **Stage & Test** – Gain experience in running your application in the public cloud.
- > **Deploy** - Deploy your application into full production for customer use directly from the public cloud (not currently available).
- > **Demo** – Publish your application for demonstration purposes using the public cloud.
- > **Expo** – Search for information on complementary products and services offered by the Progress community.
- > **Community Café** – Access Progress product resources and network with others in the Progress community.
- > **Product Showroom** – Learn about the features and benefits of Progress products. Specify what technology you are interested in, and within minutes get a public cloud-based machine dedicated to your exclusive use.

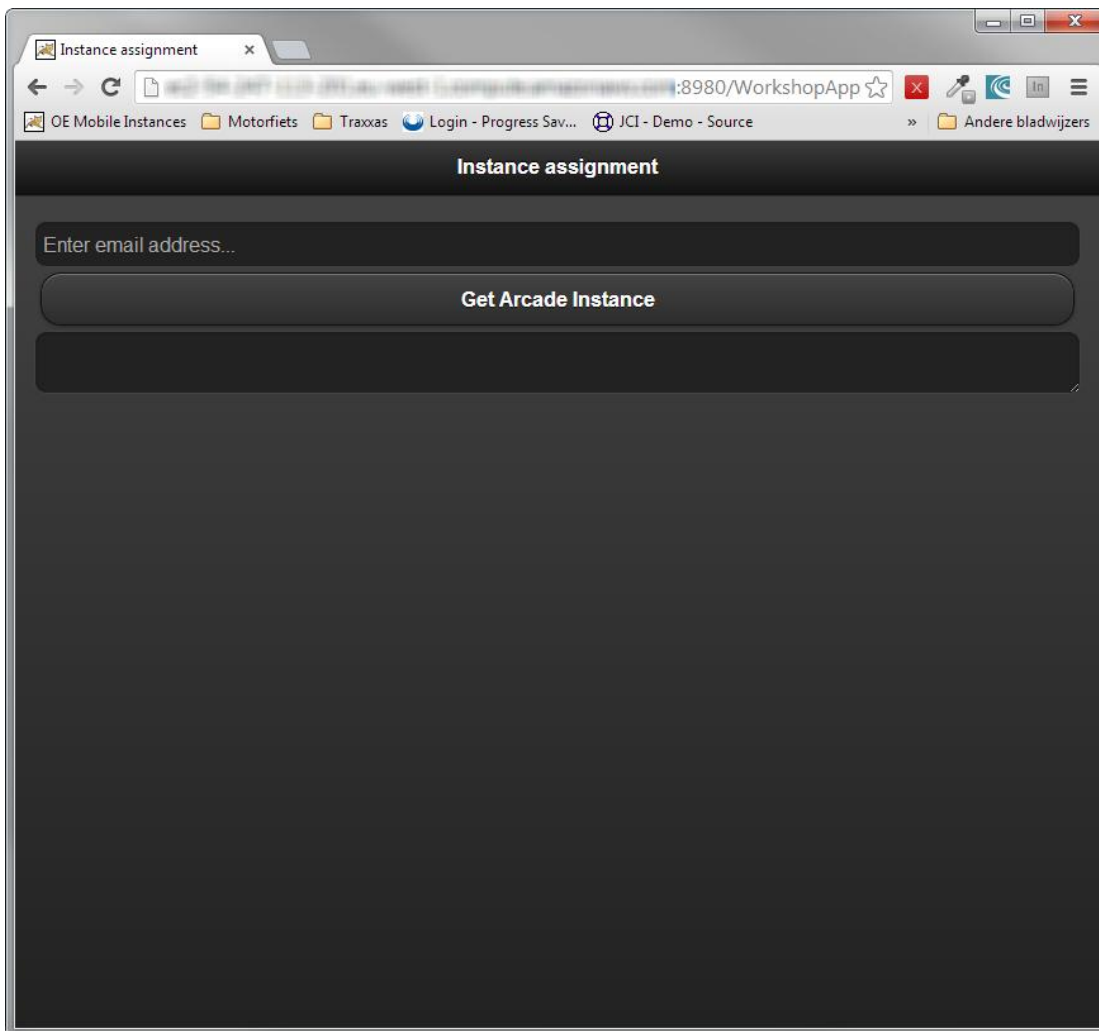
PROGRESS LOGIN CREDENTIALS

In order to be able to do this workshop, you'll need a Progress Login (PSDN, Communities, ESD,...). If you don't have one yet, don't worry. Just go to Appendix A – Getting a Progress login and follow the described steps.

GET AN ARCADE INSTANCE

- Open a browser (Google Chrome, Mozilla Firefox or Apple Safari)
- Go to <http://54.226.102.130:8980/WorkshopApp>
- Enter your email address
- Press the button: Get Arcade Instance

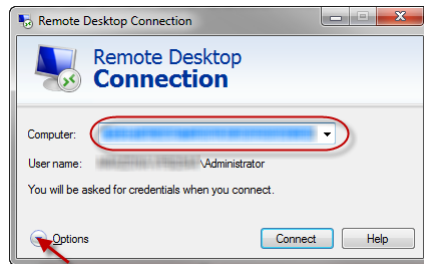
This will give you a DNS for a running Arcade instance, which you can use to connect an RDP session.



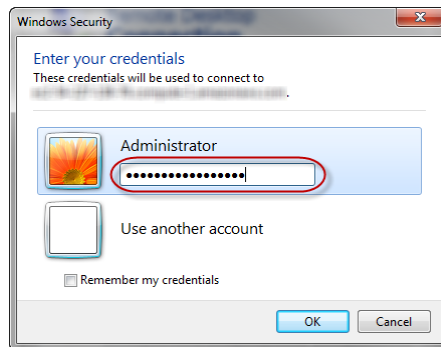
LAUNCHING YOUR MODERNIZATION DEMO MACHINE ON PROGRESS ARCADE

To launch your Modernization Demo machine:

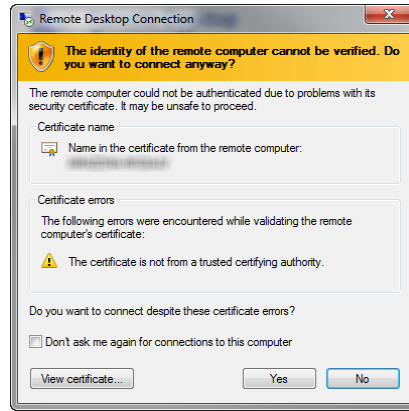
1. Several machines have been started for this workshop. Each of you will be given a unique DNS address by the workshop host at this point which points to a virtual windows server running on Progress Arcade.
2. On your laptop, choose **Start| Accessories| Remote Desktop Connection** and paste your unique DNS into the **Computer** field. Click the **Connect** button. You may need to click the button left of Options to change (erase) the domain name.



3. Enter **Administrator** for the Username, and **Exchange2013** as the Password.



4. Click the **Yes** button to say you trust this remote connection.



5. You should now be connected to the demonstration machine running in Progress Arcade.

Note: The machine instance has may have a different keyboard setting than what you are using. If your laptop has a different keyboard, you may need to set this up yourself on the machine instance through **Control Panel| Region and Language** settings.

Chapter 2 – Separating Business Logic

What Business Logic to select

It's not needed to split up your entire legacy application in order to modernize only a part.

Your Work environment



Startup existing ERP system



Startup Development Environment

Test the current ERP

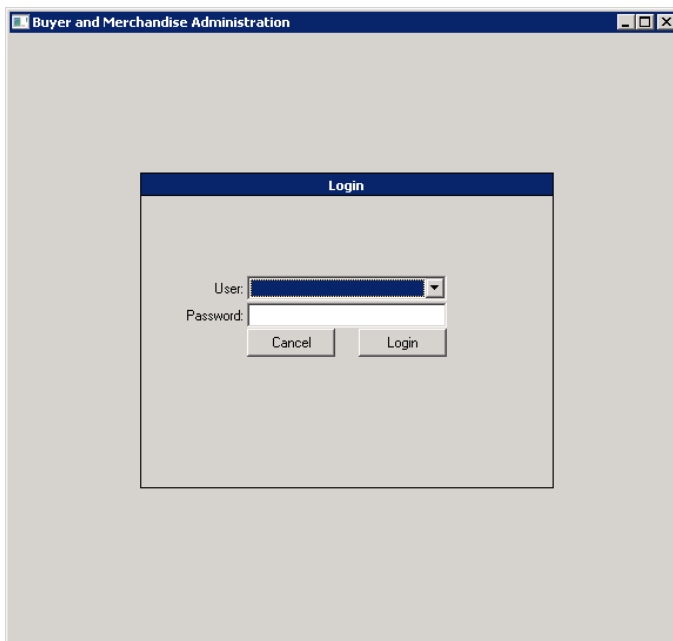
Start the current ERP system with the 'ERP' icon on your desktop. This icon also can be used during the workshop to test your changes.

Login can be done with one of the existing users.

Each user behaves different in the application. In general, 'Buyer' can only add articles. 'Supplier' can only update 'Supplier tab' and 'Merchandising' can only update 'Merchandising tab'.

Password can be left blank for all users.

After login you will get the main screen.



For the total functional description look in appendix “B”.

Article Number	Article Description	Supplier Name	Status
1	Cabernet Sauvignon	Wines International	Administere
10	Hele Droge Wijn	Wines International	Administere
2	Special Spirits	Spirits Co.	new
3	White stuff above 80°	Moonshine Inc	Administere
4	Alco pop Red	Spirits Co.	Administere
7	dddd	Wines International	New
8	magic stuff	Wines International	New
9	Gummy Ducks	Spirits Co.	Administere

Article Number: 1
Supplier Code: WINE001 Wines International
Supplier Reference: aa
Description: Cabernet Sauvignon

Predict Shipping Cost: 0.00
New Save Cancel

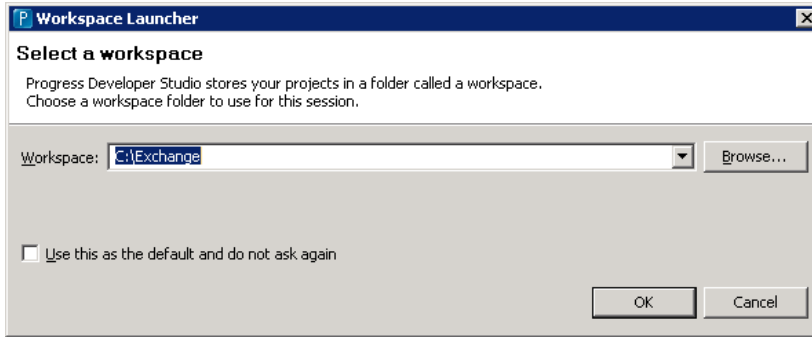
Some general notes:

- Data access is done from the UI.
- Some security rules are removed to make it easier for testing.

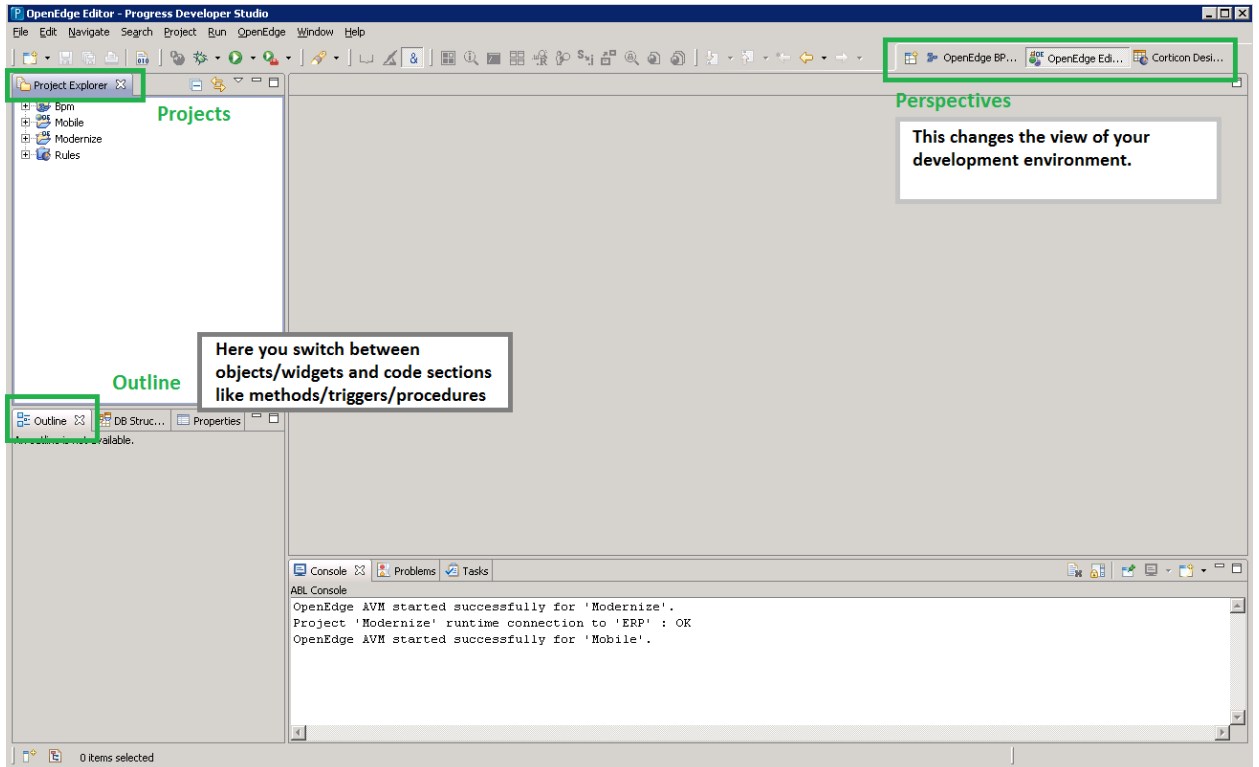
- Try to create a new article.
- Try to update an article as Supplier.
- Try to update an article as Merchandising.
- Try the Predict Shipping Cost.

Development Environment

Open your development environment by double-clicking the ‘Developer icon’ on the desktop.
As workspace select “c:\Exchange” and press Ok.



You now will see a development environment with at the left side (Project Explorer) already 3 projects set-up.



OEBpm : Here we will define our business process and design the webscreen for the Suppliers.

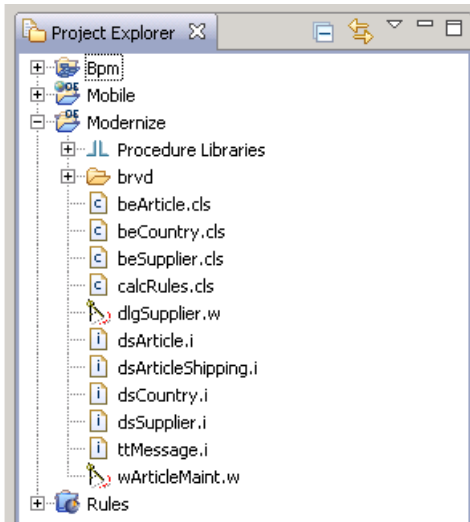
Modernize : This holds all the files from the current application and some already prepared files.

Rules : Here we will define our External Rules for the Shipment calculation.

Mobile : You will create later yourself.

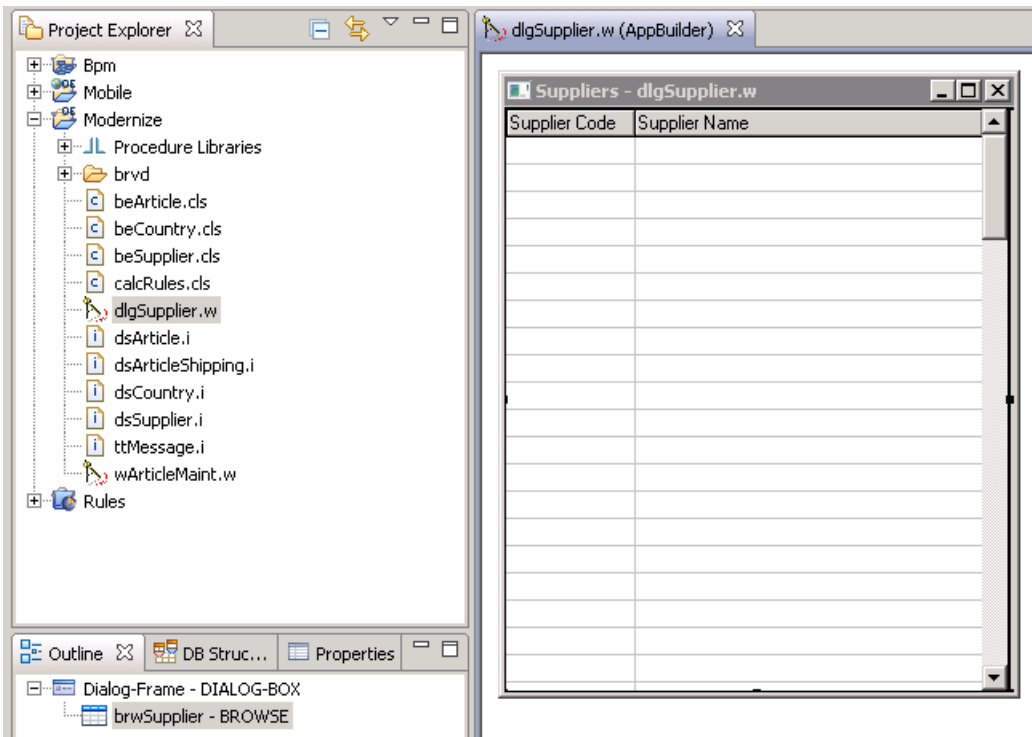
Extract Business Logic from GUI

Expand The 'Modernize' project



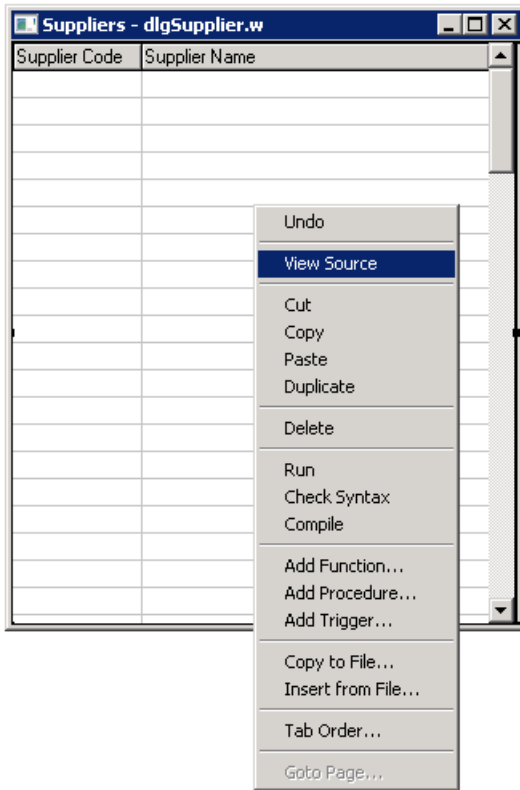
Extract getSupplier code from Supplier dialog

Open dlgSupplier.w

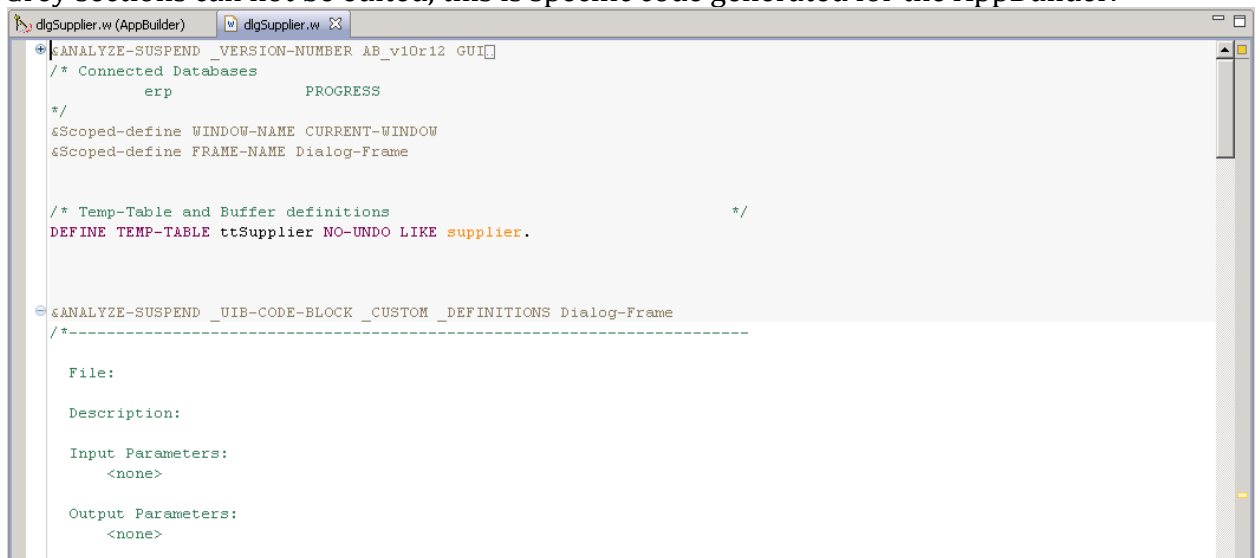


Go to the Section Editor.

You can do this by a right-click on `dlgSupplier.w` and selecting "Open With/Openedge ABL Editor". You can also select the grid in the Appbuilder, right-click on it and select "View Source" or press F9.



Grey sections can not be edited, this is specific code generated for the AppBuilder.



First we will add a dataset definition to the window definition section because we want all our data which travels between front and backend using datasets. This will make it easier and more standardized to reuse our backend code later on.

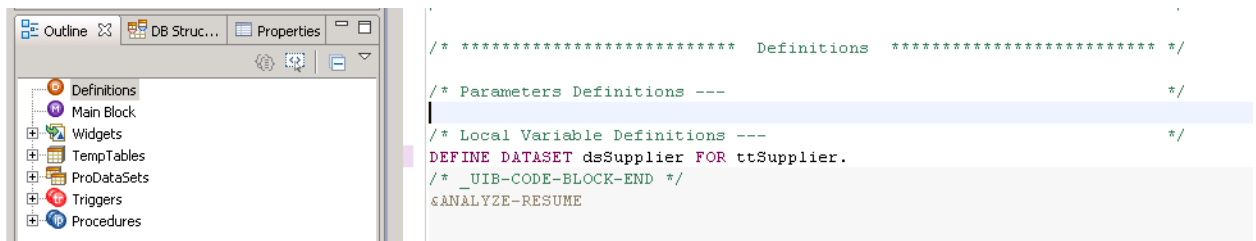
```

/* Local Variable Definitions ---
DEFINE DATASET dsSupplier FOR ttSupplier.
*/

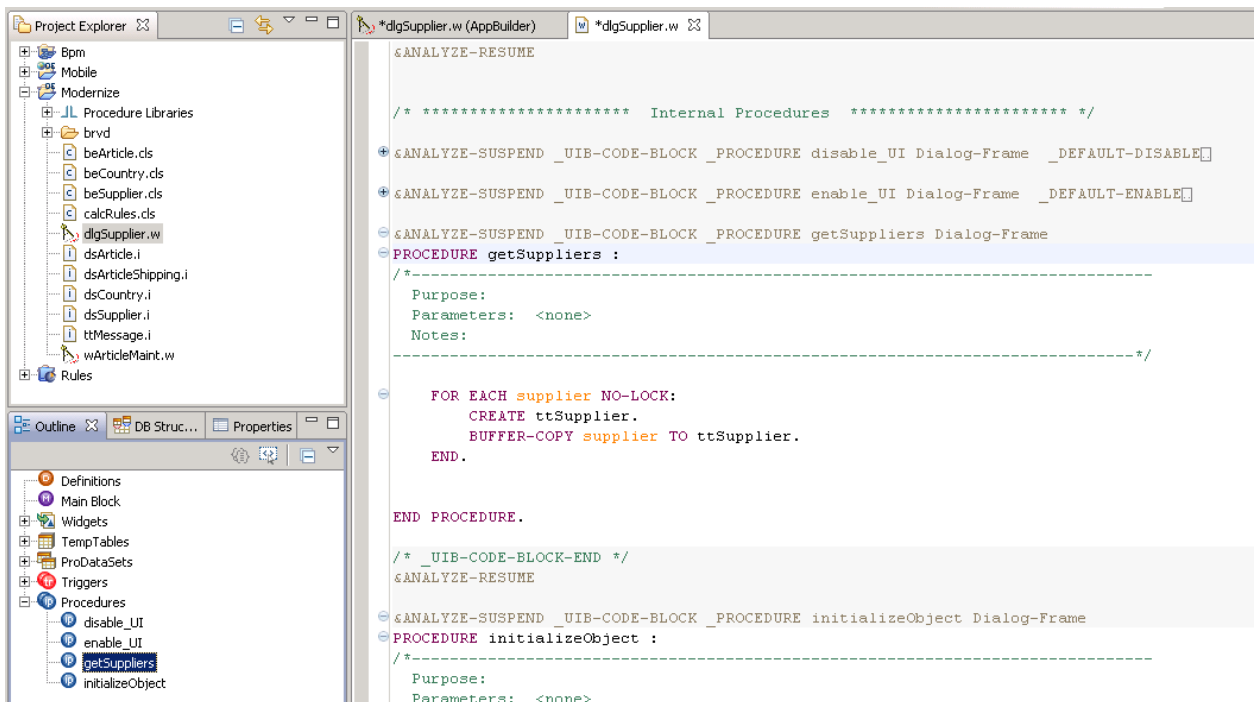
```

You can easily jump to code sections via the outline view.

Just click on 'Definitions' and Definitions section will be viewed on the screen.
Add your dataset definition here.



Now we will rebuild the internal procedures to get our supplier list.
It should call a class now which runs on the backend instead of having a direct db connection from the GUI.
Select Procedures -> getSuppliers in the Outline View. In the section editor we now see the code from the getSuppliers procedure.
You see that this is looping through a database table and fills a temp-table.
Filling this temp-table should happen at the backend.



Comment or remove the old code and add the code which instantiates a Supplier class and invokes the ReadbeSupplier method.
The method expects a filter as input and return a dataset as output.
Filter can stay blank because we want all Suppliers.

```

/*-----
Purpose:
Parameters: <none>
Notes:
-----*/

/* FOR EACH supplier NO-LOCK:      */
/*   CREATE ttSupplier.            */
/*   BUFFER-COPY supplier TO ttSupplier. */
/*   END.                          */

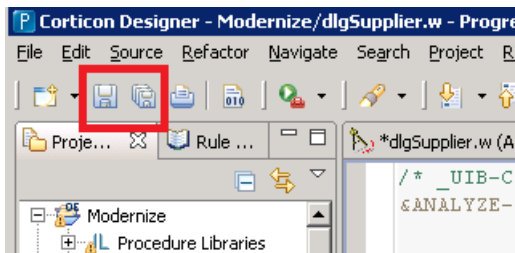
DEFINE VARIABLE objSupplier AS CLASS beSupplier.
objSupplier = NEW beSupplier().
objSupplier:ReadbeSupplier(INPUT "",
                           OUTPUT DATASET dsSupplier).

END PROCEDURE.

```

Save your project

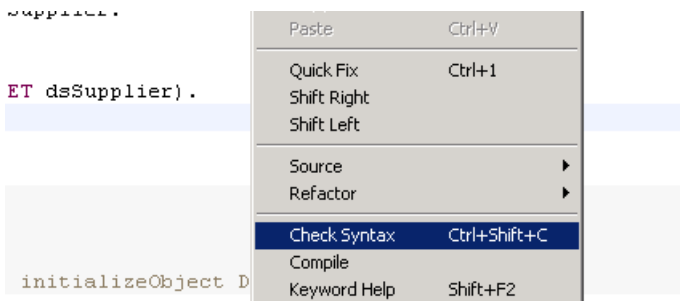
You can click the save or save-all buttons, or ctrl-shift-S will also work.



You can also check your syntax from time tot time.

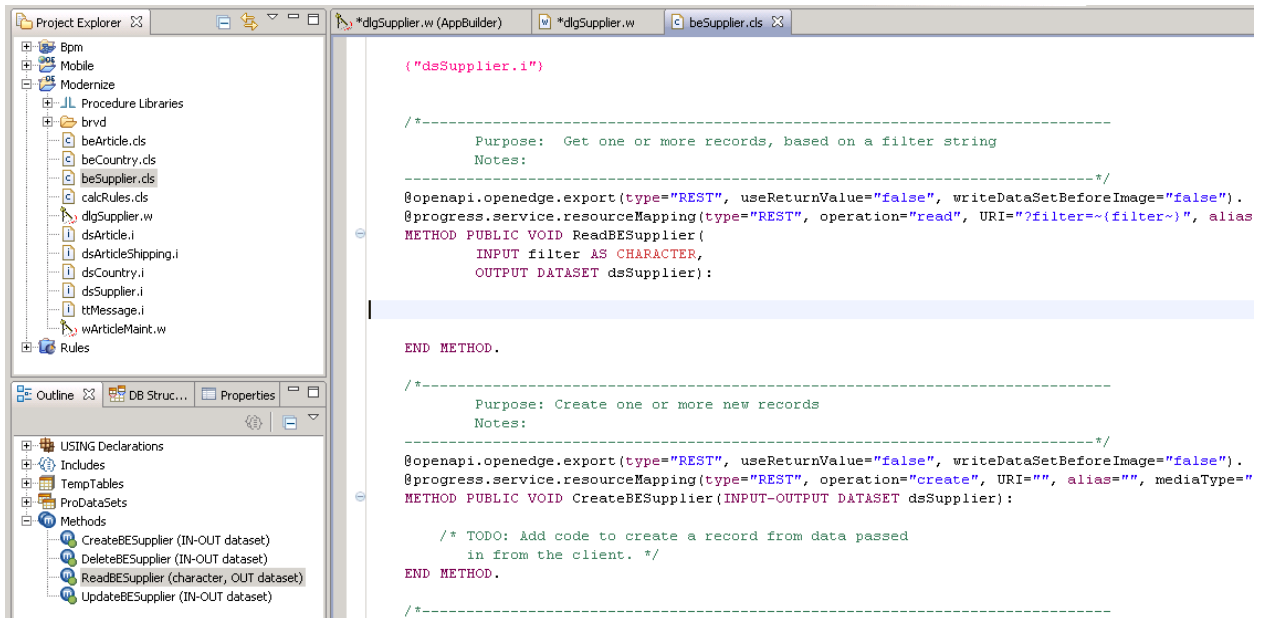
Ctrl-shift-C

Or right-click code and select 'Check Syntax'.



Now we need to add code at the backend to fill the supplier dataset. We can use the same code which was in the screen, but it would be better to use dataset methods. This gives a better performance compared to a “for each” and is more flexible in complex situations.

Open the ‘beSupplier.cls’ by double-clicking this file in the project Explorer. Then select the ‘ReadbeSupplier’ Method in the outline view. Now we see the ‘ReadbeSupplier’ Method in the section editor.



Add following code in the ReadbeSupplier methods to fill the Supplier Dataset.

```
METHOD PUBLIC VOID ReadBESupplier(
    INPUT filter AS CHARACTER,
    OUTPUT DATASET dsSupplier):
```

```
    DEFINE DATA-SOURCE srcSupplier FOR supplier.
```

```
    DATASET dsSupplier:empty-dataset().
```

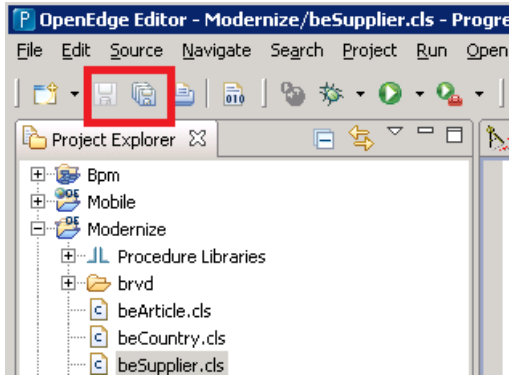
```
    BUFFER ttSupplier:ATTACH-DATA-SOURCE(DATA-SOURCE srcSupplier:HANDLE).
```

```
    DATASET dsSupplier:FILL().
```

```
    BUFFER ttSupplier:DETACH-DATA-SOURCE().
```

```
END METHOD.
```

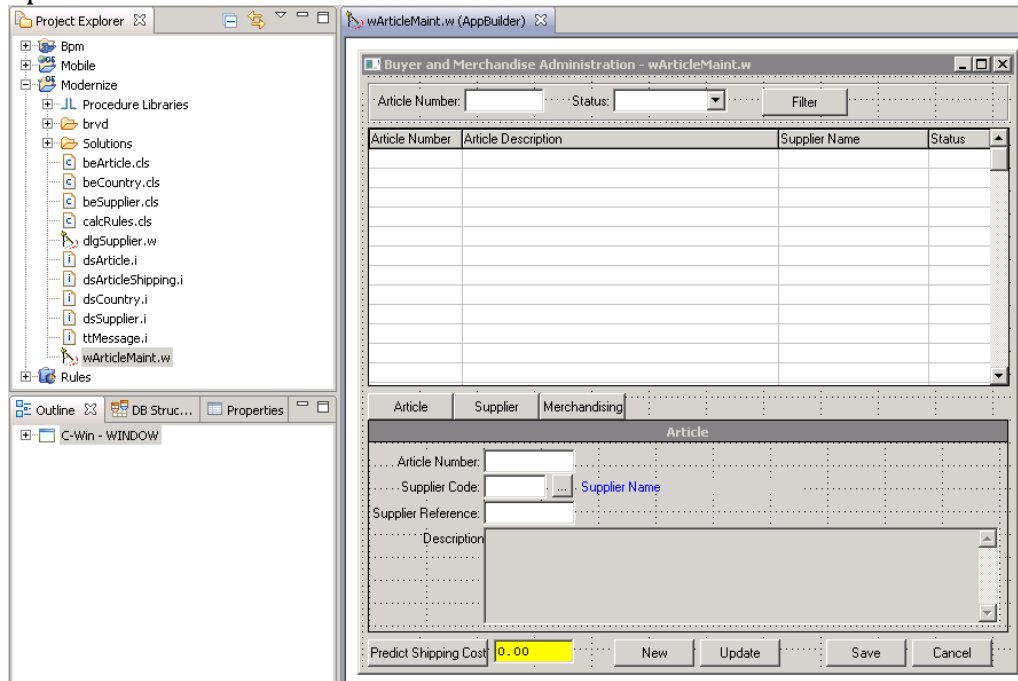
Save all your work. With the save-all button or pressing ctrl-shift-S.



Test the application and check that the supplier dialog is filled with all the Suppliers. To verify that the code you just added gets executed you can add a message to the ReadBESupplier method: `MESSAGE PROGRAM-NAME (1) VIEW-AS ALERT-BOX.`

Extract createArticle code from wArticleMaint

Open wArticleMaint.



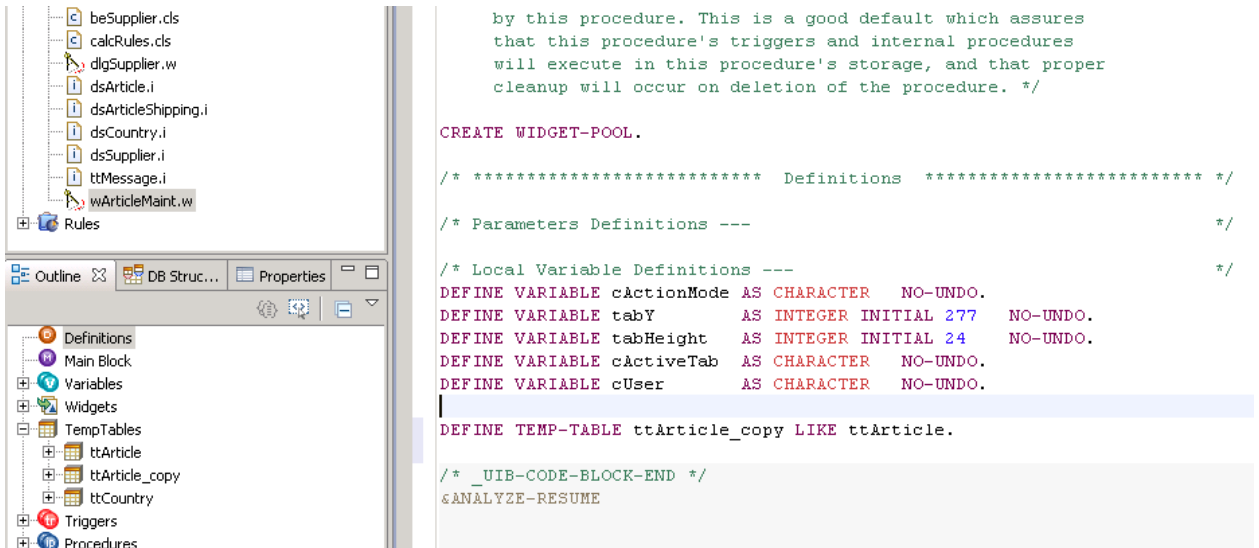
Go to the section code editor with right-click and View source or F9.

Add temp-table definition to the definition section.

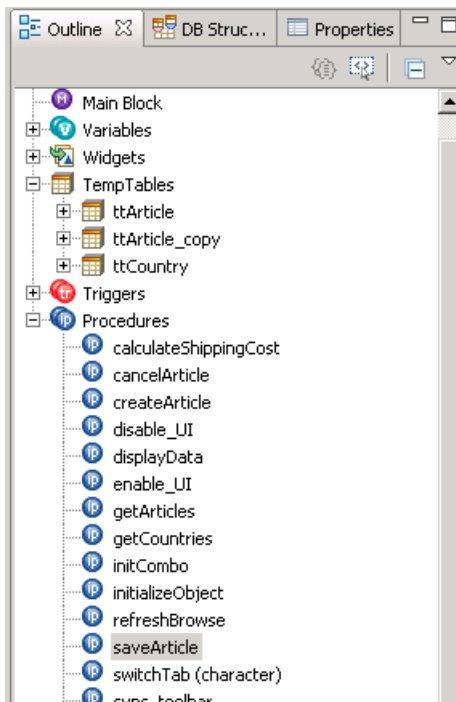
We define a second temp-table to store only the changed record to send to the backend.

```
/* Local Variable Definitions --- */
DEFINE VARIABLE cActionMode AS CHARACTER NO-UNDO.
DEFINE VARIABLE tabY AS INTEGER INITIAL 343 NO-UNDO.
DEFINE VARIABLE tabHeight AS INTEGER INITIAL 29 NO-UNDO.
DEFINE VARIABLE cActiveTab AS CHARACTER NO-UNDO.
DEFINE VARIABLE cUser AS CHARACTER NO-UNDO.
```

```
DEFINE TEMP-TABLE ttArticle_copy LIKE ttArticle.
```



Now Select the saveArticle procedure in the outline pane.



Change the 'saveArticle' procedure that it calls the backend validation and save method.

- Comment or remove the validation check code section
- Comment or remove the Set new status code
- Comment or remove the Save code

Add following code before the "show error" code section.

Your code should look like on the next 2 pages.

```

PROCEDURE saveArticle :
/*-----
Purpose:
Parameters: <none>
Notes:
-----*/
DEFINE VARIABLE cError AS CHARACTER NO-UNDO.

/* Assign screen-values */
ASSIGN FRAME frmArticle {&FIELDS-IN-QUERY-frmArticle}.

IF ERROR-STATUS:ERROR
THEN DO:
ASSIGN cError = "Article number already exists.".
/* Show CLIENT-SIDE error */
MESSAGE cError
VIEW-AS ALERT-BOX ERROR BUTTONS OK.
APPLY "entry" TO ttArticle.article_nr.
/* RETURN NO-APPLY. */
UNDO, LEAVE.
END.
DISPLAY ttArticle.article_nr ttArticle.supplier_code WITH FRAME frmSupplier.
DISPLAY ttArticle.article_nr ttArticle.supplier_code WITH FRAME frmMerchandising.

ASSIGN FRAME frmSupplier {&FIELDS-IN-QUERY-frmSupplier}.
ASSIGN FRAME frmMerchandising {&FIELDS-IN-QUERY-frmMerchandising}.

/* Validation */
/*IF cError EQ ""
/*THEN DO:
/* CASE cActionMode:
/* WHEN "new"
/* THEN DO:
/* /* Article_nr unique ? */
/* IF ttArticle.article_nr EQ ""
/* THEN
/* ASSIGN cError = "Article Nr can not be empty".
/* IF CAN-FIND(FIRST article NO-LOCK WHERE article_nr = ttArticle.article_nr)
/* THEN
/* ASSIGN cError = cError + CHR(13) + "Article Code already exists. Choose unique one."
/* /* Supplier exists? */
/* IF NOT CAN-FIND(FIRST supplier NO-LOCK WHERE supplier_code = ttArticle.supplier_code)
/* THEN
/* ASSIGN cError = cError + CHR(13) + "Supplier does not exist.".
/* END.
/* END CASE.
/* IF ttArticle.article_code_supplier EQ ""
/* THEN
/* ASSIGN cError = cError + CHR(13) + "Supplier Reference can not be empty.".
/* IF ttArticle.article_desc EQ ""
/* THEN
/* ASSIGN cError = cError + CHR(13) + "Description can not be empty.".
/* IF ttArticle.weight_gross < ttArticle.weight_net
/* THEN
/* ASSIGN cError = cError + CHR(13) + "Gross Weight can not be smaller than net weight.".
/*END.

/* Call backend code */
DEFINE VARIABLE objArticle AS CLASS beArticle.
EMPTY TEMP-TABLE ttArticle_copy.
CREATE ttArticle copy.
BUFFER-COPY ttArticle TO ttArticle_copy.
objArticle = NEW beArticle().
objArticle:validateSaveArticle(INPUT cActionMode,
INPUT TABLE ttArticle_copy,
INPUT cUser,
OUTPUT cError).

```



```

/* Show error */
IF cError NE ""
THEN DO:
    MESSAGE cError
        VIEW-AS ALERT-BOX ERROR BUTTONS OK.
    APPLY "entry" TO ttArticle.article_nr.
    RETURN NO-APPLY.
END.

/* Set New Status */
/*CASE cUser:
/* WHEN "Supplier"
/* THEN ASSIGN ttArticle.article_status = "Administered".
/* WHEN "Merchandising"
/* THEN ASSIGN ttArticle.article_status = "Purchase".
/*END CASE.

/* Save */
/*FIND FIRST article EXCLUSIVE-LOCK
/* WHERE article.article_nr = ttArticle.article_nr NO-ERROR.
/*IF NOT AVAILABLE article
/*THEN
/* CREATE article.
/*BUFFER-COPY ttArticle TO Article.

/* Refresh data */
RUN refreshBrowse.

RUN sync_widgets("Save").

END PROCEDURE.

```

The backend code to create an article already exists.
 You can find this by opening the beArticle class and validateSaveArticle method.

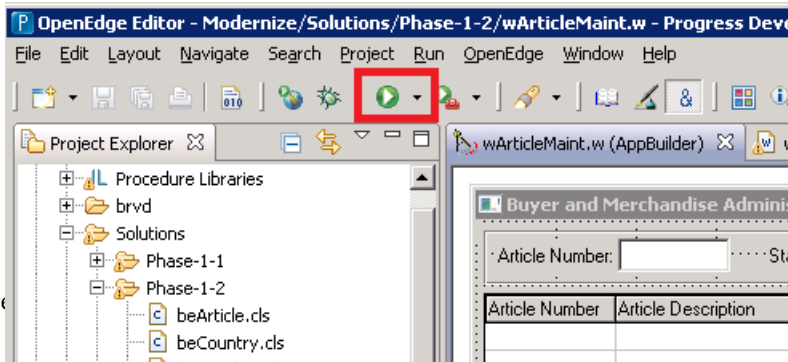
Save your code.

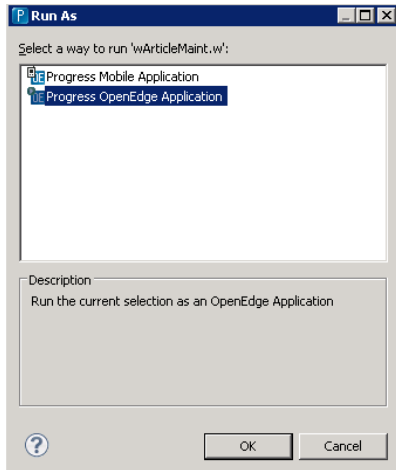
Test the ERP application to check if create of an article is still working.
 Use the "Buyer" user login.

Easiest way to test your code is by starting the ERP application from the icon on your desktop.

Or you can select the wArticleMaint.w tab in the development environment and click the "Run As" button.

Then Choose "Progress OpenEdge Application". And Click OK.





Extra work

Extract the fill of country records for the combo-box from the GUI.
 Replace the fill of temp-table in the getCountries method with a backend call. Don't forget to define the object (beCountry) and dataset (dsCountry) definition!

```

/* FOR EACH country NO-LOCK:      */
/*   CREATE ttCountry.             */
/*   BUFFER-COPY country TO ttCountry. */
/* END.                            */

```

```

DEFINE VARIABLE objCountry AS CLASS beCountry.
objCountry = NEW beCountry().
objCountry:ReadbeCountry(INPUT "",
                          OUTPUT DATASET dsCountry).

```

```

END PROCEDURE

```

```

/* Definitions */
DEFINE DATASET dsCountry FOR ttCountry.

```

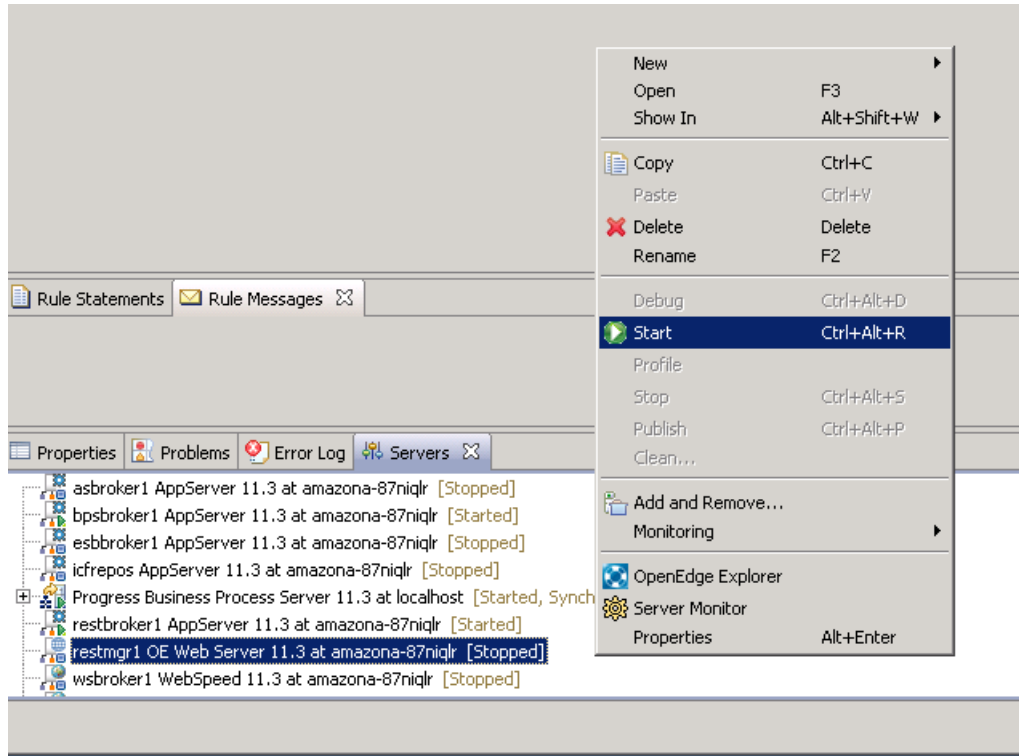
Backend class is "beCountry". "ReadbeCountry" method is already supplied.
 Test the application.

Chapter 3 – Implementing OE-Corticon

Extract rules and calculations from the GUI client.

Preperation

Go to the server view, right-click on the 'restmgr1 OE WebServer 11.3' and press Start. This will start the rules engine. Startup takes about 1 minute.



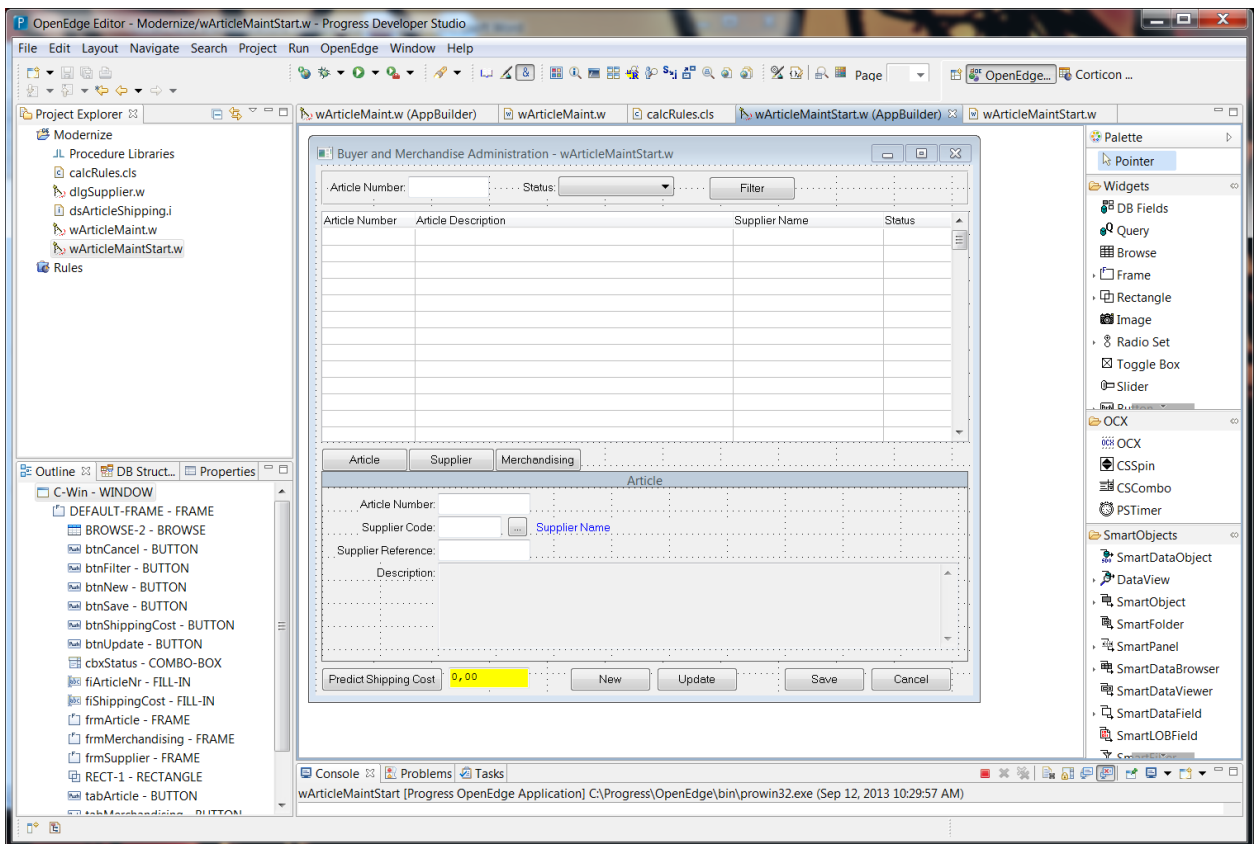
Add extra “CalculateRule” button on the current screen and fill-in for the result.

*For testing later on we will leave the old code like it is and add a second calculate button.

Open the Modernize project in the Project Explorer.
This project is the ABL side of the application.

Open the “OpenEdge Editor” perspective.

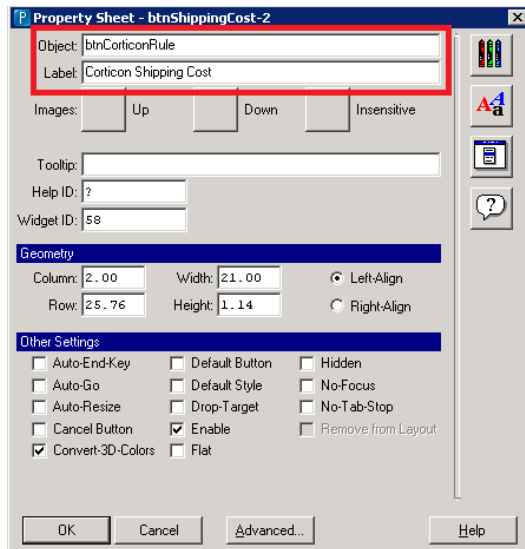
Open the “wArticleMaint” file.



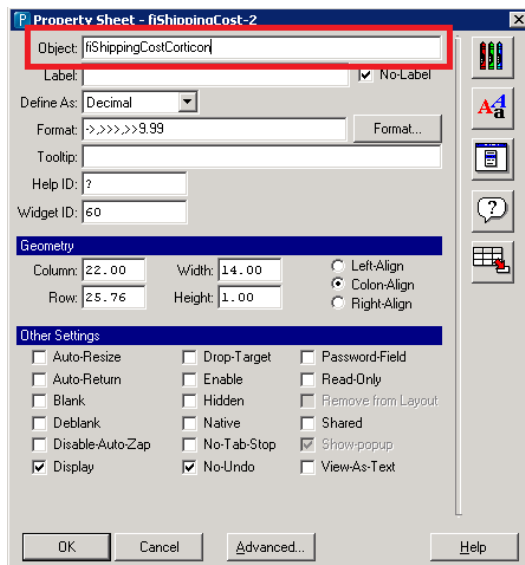
Make the screen a bit higher so that we can add an extra button and fill-in for our code.
A new button and fill-in will be added below the “predict and yellow result fill-in.

We do this so you can test the old OE logic and compare it with the External Corticon rule.

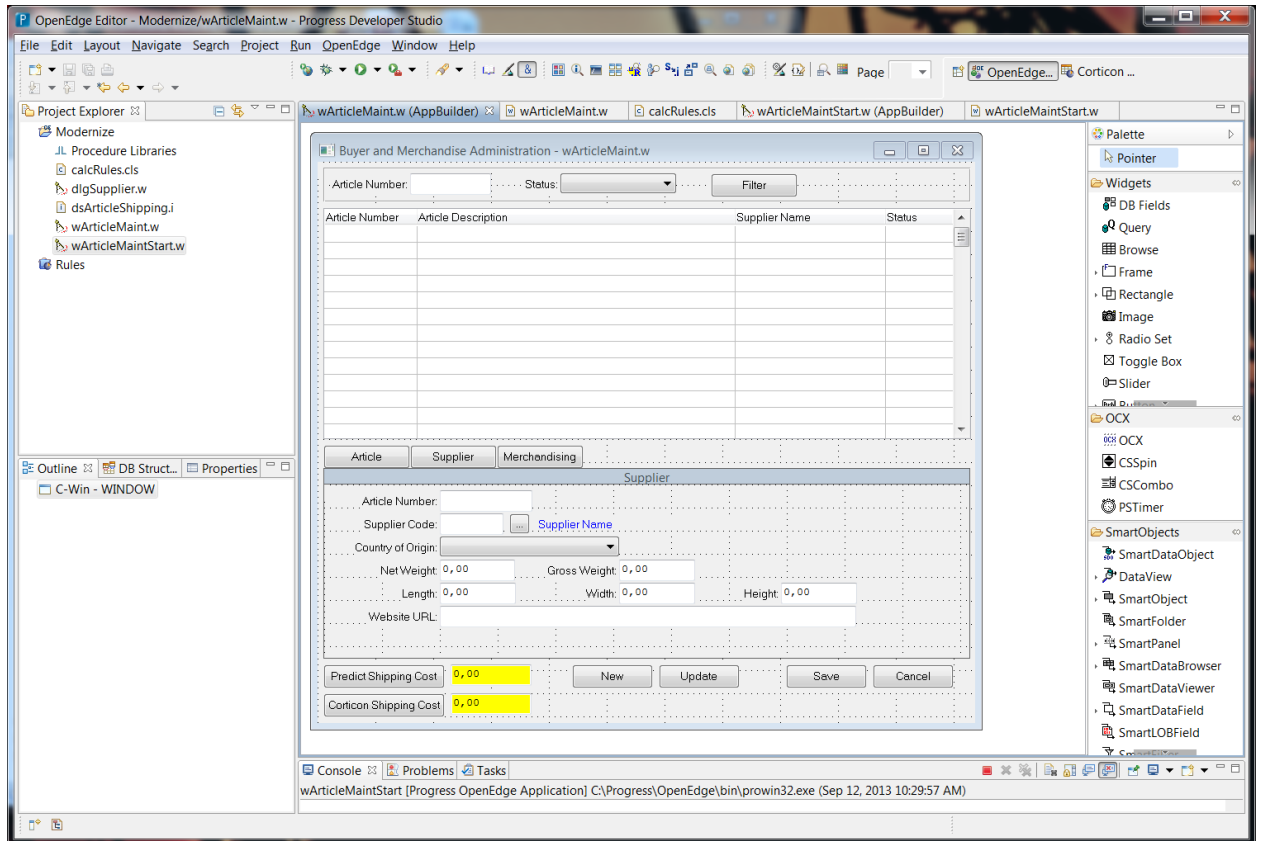
-Copy and paste the “Predict Shipping Cost” button.
 Change the label and object name of the new button.
 You can do this by double-clicking the widget.
 Object Name : btnCorticonRule
 Label : Corticon Shipping Cost



-Copy and paste the “yellow result fill-in” field.
 Change name of the new fill-in.
 Object Name : fiShippingCostCorticon



Result:

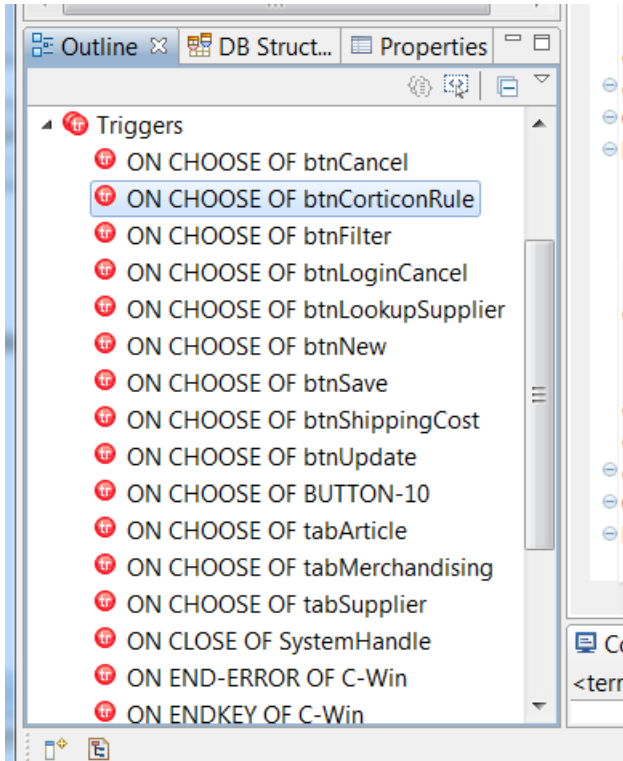


Change the trigger code behind the new button so that it will call the Corticon rule.

Change the trigger code of the new button.

First we need to go to the code section editor by pressing F9.

Now select the “ON CHOOSE OF btnCorticonRule” trigger in the outline pane.

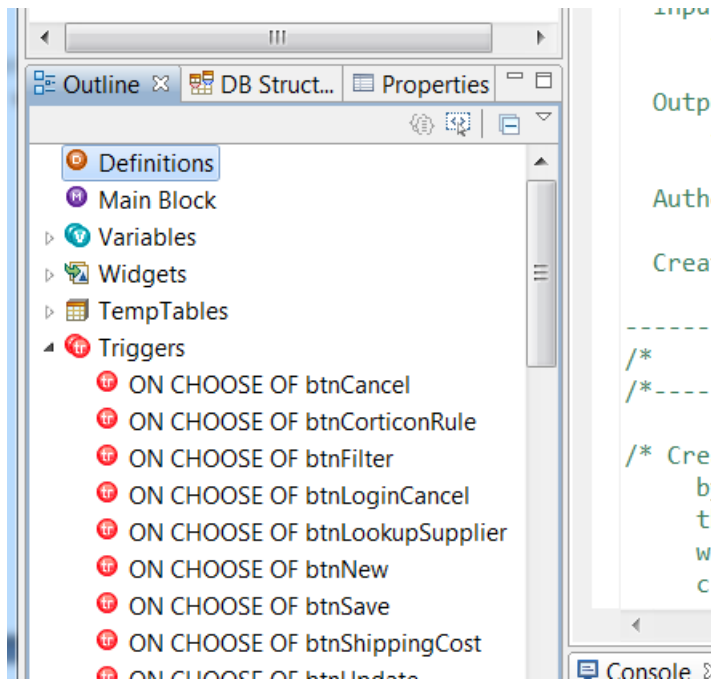


Change the run statement to a new internal procedure.

From `calculateShippingCost` to `calculateShippingCostCorticon`

```
ON CHOOSE OF btnCorticonRule IN FRAME DEFAULT-FRAME /* Corticon Shipping Cost
*/
DO:
    RUN calculateShippingCostCorticon.
END.
```

Add the schema-file to the window which we will use to send to the rules Engine.
Go to the definitions section in the outline pane.



Add The schema include file to the definitions section

```

/* ***** Definitions ***** */
/* Parameters Definitions --- */
/* Local Variable Definitions --- */
DEFINE VARIABLE cActionMode AS CHARACTER NO-UNDO.
DEFINE VARIABLE tabY AS INTEGER INITIAL 343 NO-UNDO.
DEFINE VARIABLE tabHeight AS INTEGER INITIAL 29 NO-UNDO.
DEFINE VARIABLE cActiveTab AS CHARACTER NO-UNDO.
DEFINE VARIABLE cUser AS CHARACTER NO-UNDO.

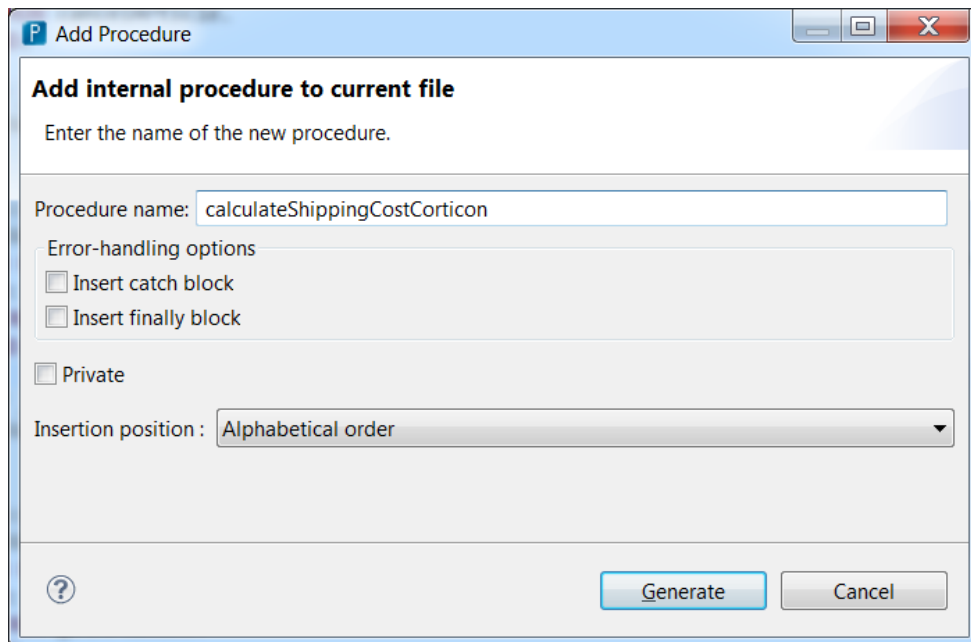
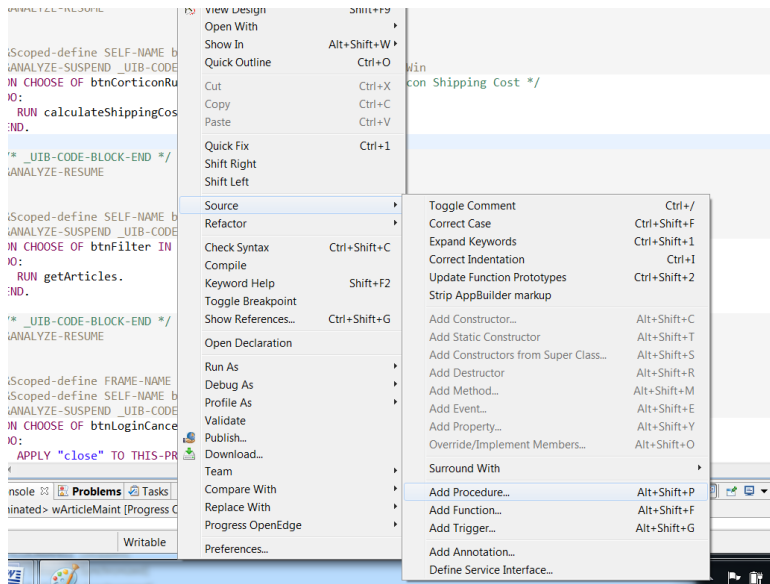
DEFINE TEMP-TABLE ttArticle_copy LIKE ttArticle.
DEFINE DATASET dsCountry FOR ttCountry.

{dsArticleShipping.i}

```

Save your project.

Create a new internal procedure called “**calculateShippingCostCorticon**” by right-clicking the editor window, select source and select add-procedure.



Add code to the “calculateShippingCostCorticon” procedure.

```

PROCEDURE calculateShippingCostCorticon :
/*-----
Purpose:
Notes:
-----*/
DEFINE VARIABLE deShippingCost AS DECIMAL NO-UNDO.
DEFINE VARIABLE deVolume AS DECIMAL NO-UNDO.
DEFINE VARIABLE cResult AS CHARACTER NO-UNDO.
DEFINE VARIABLE objCalcRules AS CLASS calcRules NO-UNDO.

IF NOT AVAILABLE ttArticle
THEN
    RETURN.

/* Empty and fill temp-table with record to send to the rules engine */
EMPTY TEMP-TABLE ttArticleshipping.
CREATE ttArticleshipping.
BUFFER-COPY ttArticle TO ttArticleShipping.

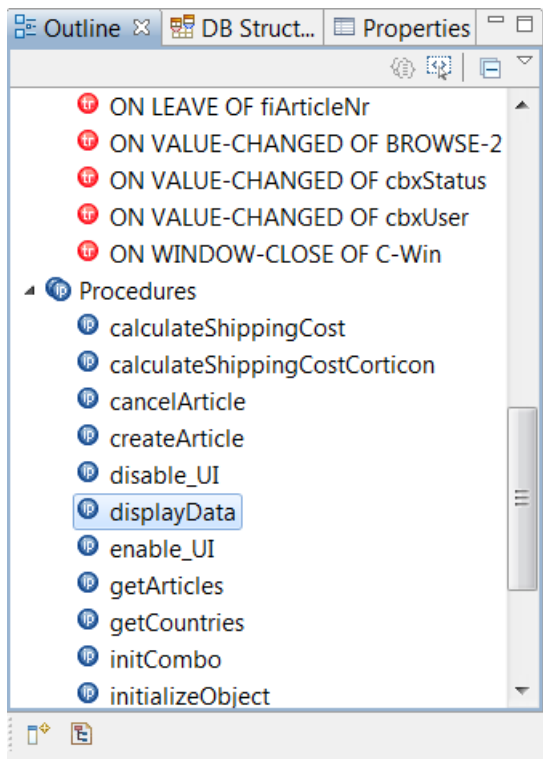
/* Instantiate CalcRules class and invoke calcShipmentCost method */
objCalcRules = NEW CalcRules().
objCalcRules:calcShipmentCost(INPUT-OUTPUT DATASET dsArticleShipping).

/* Display result in new cost field */
FIND FIRST ttArticleShipping NO-ERROR.
IF AVAILABLE ttArticleShipping THEN
DO WITH FRAME {&frame-name}:
    ASSIGN fiShippingCostCorticon:SCREEN-VALUE =
STRING(ttArticleShipping.ShippingCost)
        fiShippingCostCorticon:TOOLTIP = cResult
        fiShippingCostCorticon:BGCOLOR = 10.
END.

END PROCEDURE.

```


Uncomment the fill-in color reset code.
Go to the “displayData” internal procedure and uncomment the code.

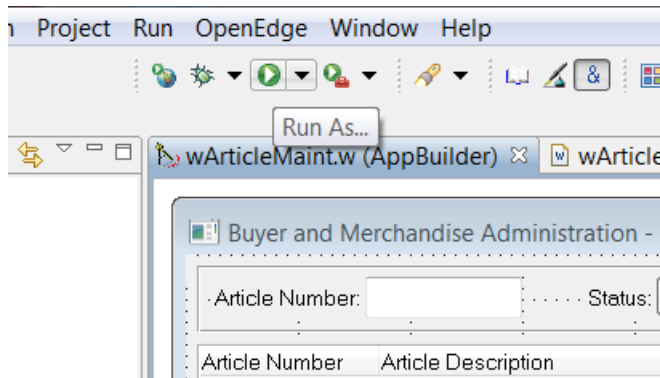


```
PROCEDURE displayData :  
  
IF AVAILABLE ttArticle  
THEN DO:  
  
    DO WITH FRAME frmArticle:  
        ASSIGN fiSupplierName = ttArticle.supplier_name.  
        DISPLAY fiSupplierName.  
    END.  
  
    DISPLAY {&FIELDS-IN-QUERY-frmArticle} WITH FRAME frmArticle.  
    DISPLAY {&FIELDS-IN-QUERY-frmSupplier} WITH FRAME frmSupplier.  
    DISPLAY {&FIELDS-IN-QUERY-frmMerchandising} WITH FRAME frmMerchandising.  
  
END.  
DO WITH FRAME {&frame-name}:  
    ASSIGN fiShippingCost:SCREEN-VALUE = STRING(0)  
           fiShippingCost:TOOLTIP      = ""  
           fiShippingCost:BGCOLOR     = 14.  
    /* ASSIGN fiShippingCostCorticon:SCREEN-VALUE = STRING(0)  
           fiShippingCostCorticon:TOOLTIP      = ""  
           fiShippingCostCorticon:BGCOLOR     = 14. */  
END.  
  
END PROCEDURE.
```

Test calling the external class.

Run the window and login with “Supplier”.

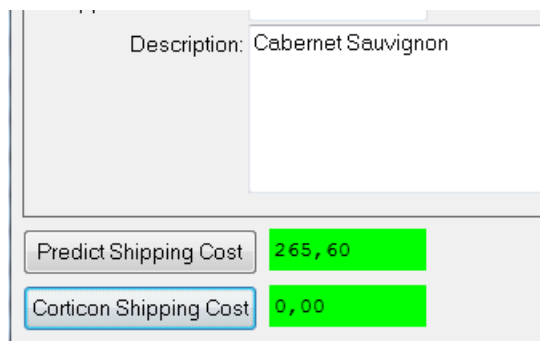
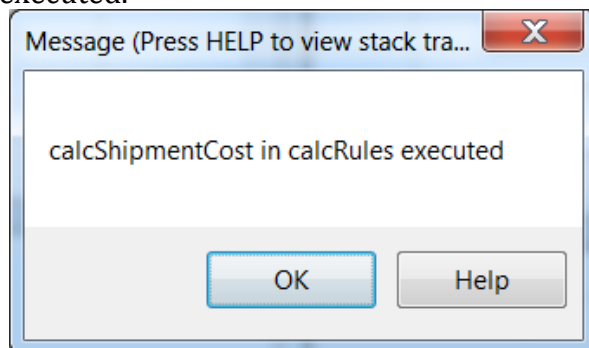
Press the green arrow ‘Run As...’ button in the menu. In the menu choose the Run As submenu and then Progress Openedge Application.



Go to the “Supplier” Tab

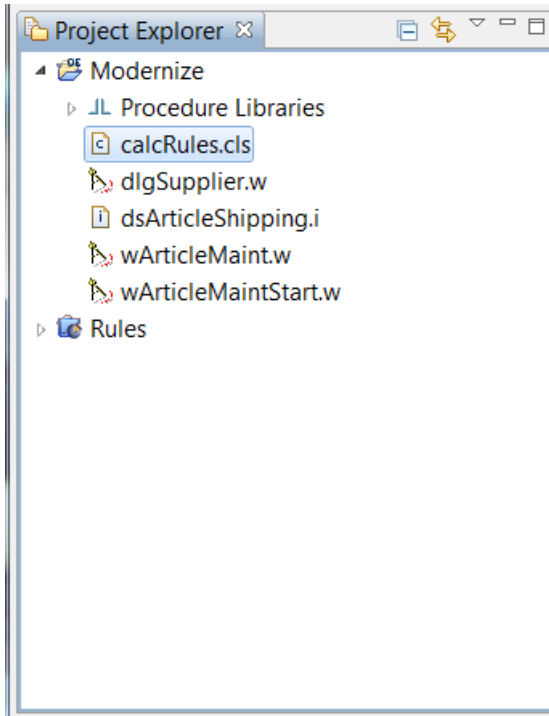
And press the 2 calculate buttons.

When clicking on a button the fill-in behind the button should become green, but the new button will not give a result yet. And you should see a message that the method has been executed.



Adding Code to call the rules Engine.

Open the file “calcRules” in the Project Explorer.



Add code to the “calcShipment” method and comment the message.

```
        METHOD PUBLIC VOID calcShipmentCost(INPUT-OUTPUT dataset FOR
dsArticleShipping  ):

/*      MESSAGE "calcShipmentCost in calcRules executed"*/
/*      VIEW-AS ALERT-BOX.          */

/* Make connection to ShipmentCost rules service */
connectRules(INPUT 1.0,
              INPUT 'ShipmentCost').

/* Invoke the decision service */
decisionService:InvokeService(INPUT-OUTPUT DATASET dsArticleShipping
BY-REFERENCE).

/* Structered Error Handling */
CATCH oAppError AS AppError:
    MESSAGE
        oAppError:GetMessage(1)
        VIEW-AS ALERT-BOX ERROR TITLE 'Error invoking service'.
END CATCH.

END METHOD.
```



Test the new calculation rule.

Start the GUI application again. Select a record and press the 2 rule buttons and compare the result.

If all works fine you should get the same result.

! First time you test Corticon rule it will be a bit slow because the new rule needs to be Initialized (few seconds).

After this it will become faster.

The screenshot shows a window titled "Buyer and Merchandise Administration". At the top, there are input fields for "Article Number:" and "Status:" with a "Filter" button. Below this is a table with the following data:

Article Number	Article Description	Supplier Name	Status
1	Cabernet Sauvignon	Wines International	Administered
2	Special Spirits	Spirits Co.	new
3	White stuff above 80°	Moonshine Inc	Administered
4	Alco pop Red	Spirits Co.	New
7	dddd	Wines International	New
8	magic stuff	Wines International	New

Below the table are tabs for "Article", "Supplier", and "Merchandising". The "Supplier" tab is active, showing a form for "Supplier" details. The form includes:

- Article Number: 1
- Supplier Code: WINE001 (with a dropdown arrow and "Wines International" text)
- Country of Origin: France (with a dropdown arrow)
- Net Weight: 10,00
- Gross Weight: 11,00
- Length: 8,00
- Width: 200,00
- Height: 200,00
- Website URL: (empty field)

At the bottom of the form, there are two buttons: "Predict Shipping Cost" and "Corticon Shipping Cost", both showing a value of "265,60". There are also "Update", "Save", and "Cancel" buttons.

Change the rule without writing code

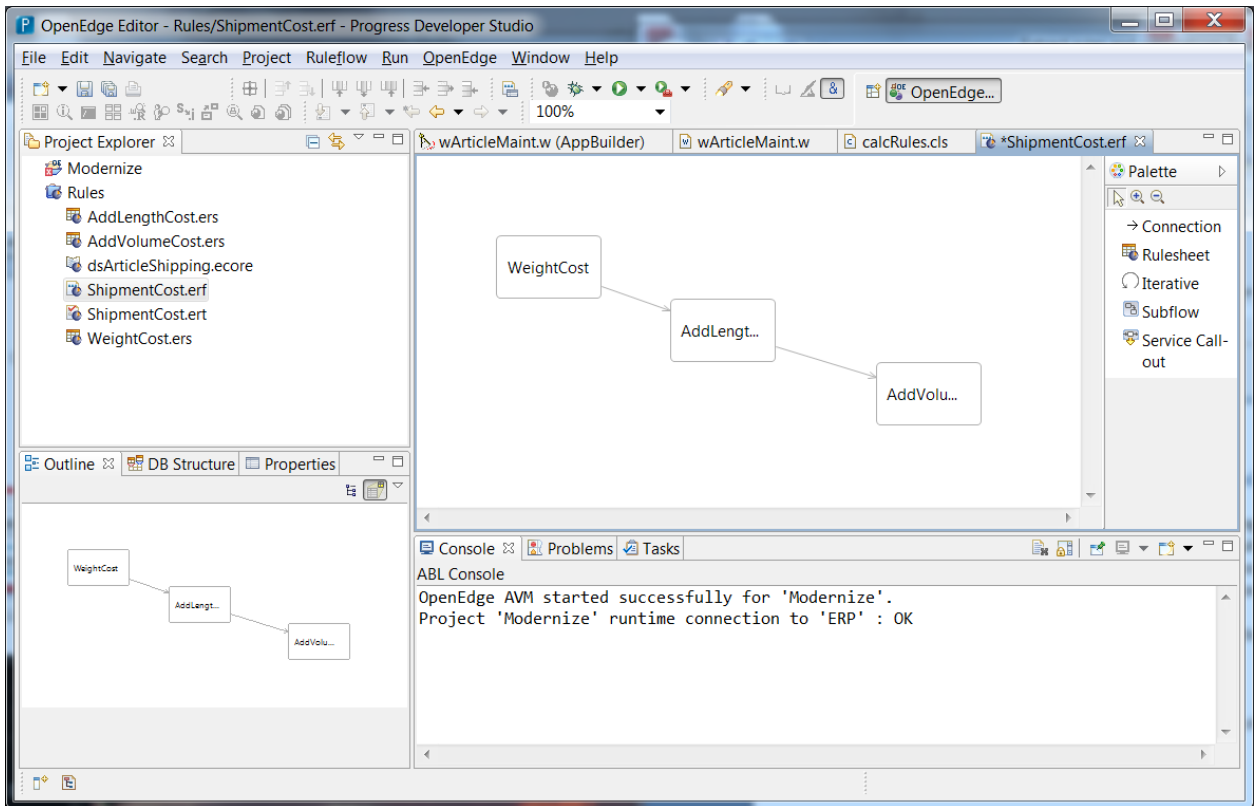
One of the application requirements is that shipping price calculation should be flexible to change.

So the first request is that an additional price should be added when we ship something to 'France' and to 'Canada'. This is because there is no local warehouse in those countries for a certain customer. For other customers it can be they have other country-specific regulations so changing the rules should not be done in code.

We now will add an extra rule to calculate an additional shipping cost depending on the 'Country of Origin'.

Open the Rules project in the Project Explorer.

Open the ShipmentCost.erf file. This is the ruleflow of the Shipment calculation.



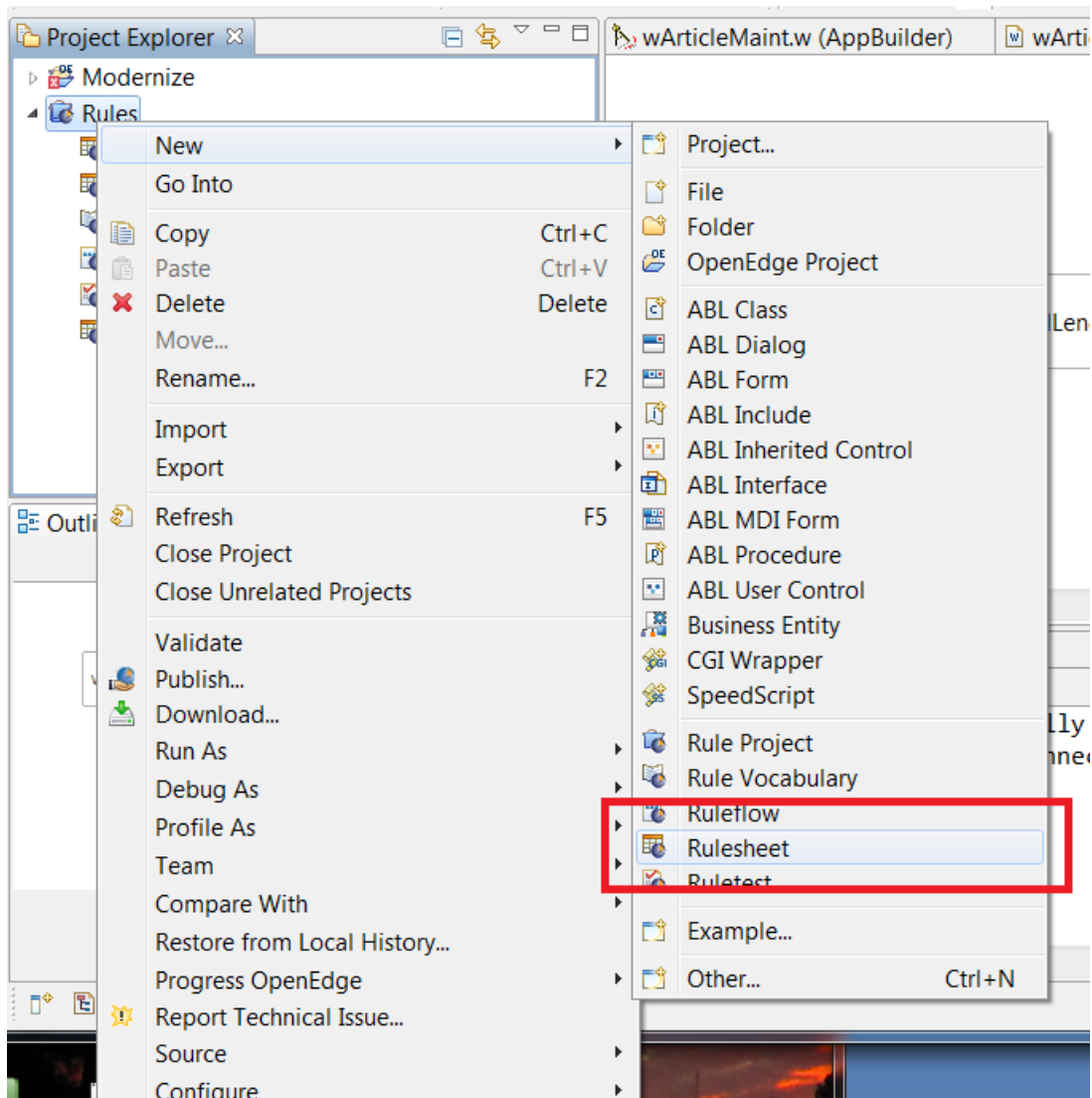
Here we see the ruleflow.

First the cost on Weight is calculated then an additional length cost is added and as last a possible additional cost for volume is added.

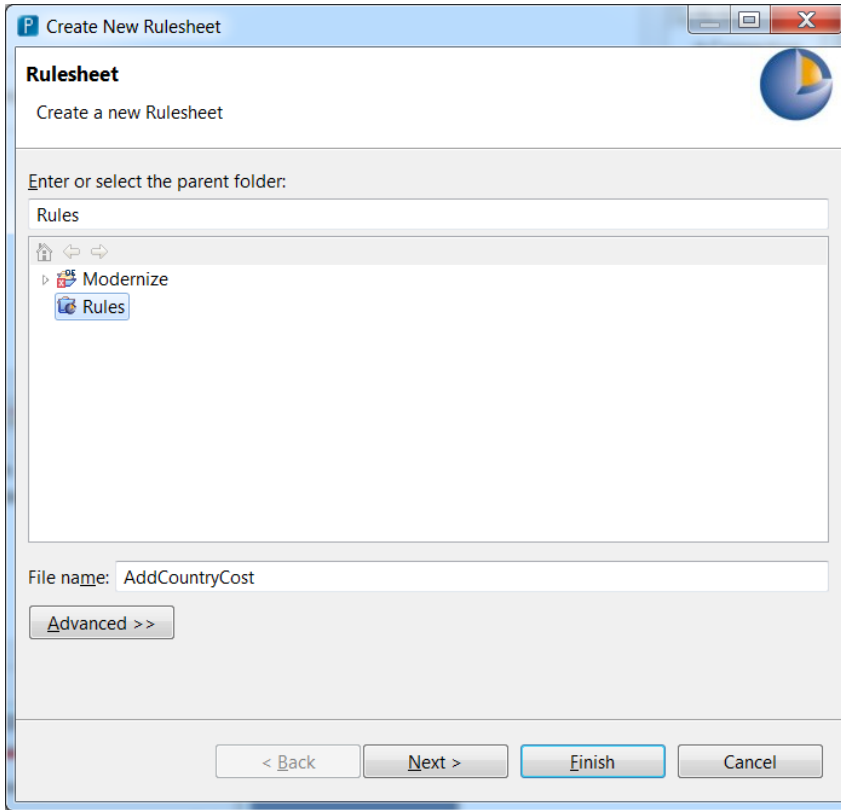
We will add an extra step that an extra cost for specific countries will be added.

Create new rulesheet

Right-click on the Rules project and select new->Rulesheet

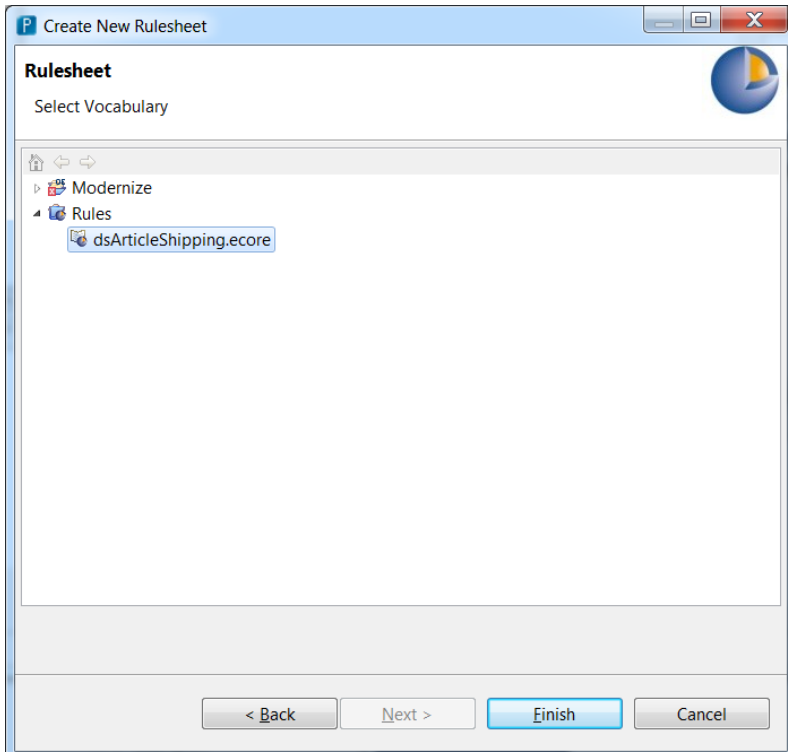


Call the new rule 'AddCountryCost'
And select the 'Rules' project



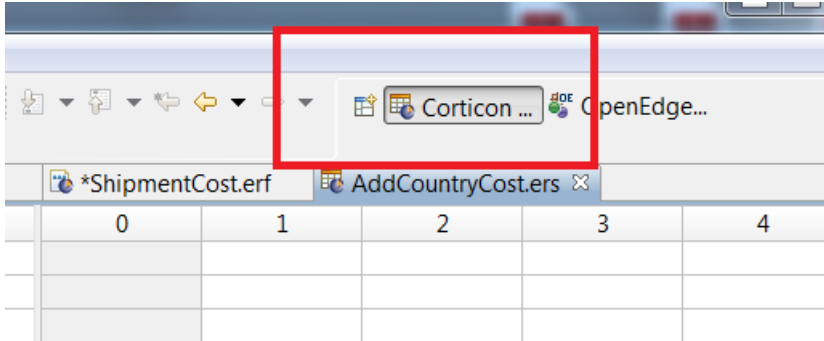
Press Next

Select dsArticleShipping.ecore. This holds the signature of the data that Corticon will receive from OpenEdge.



Press Finish.

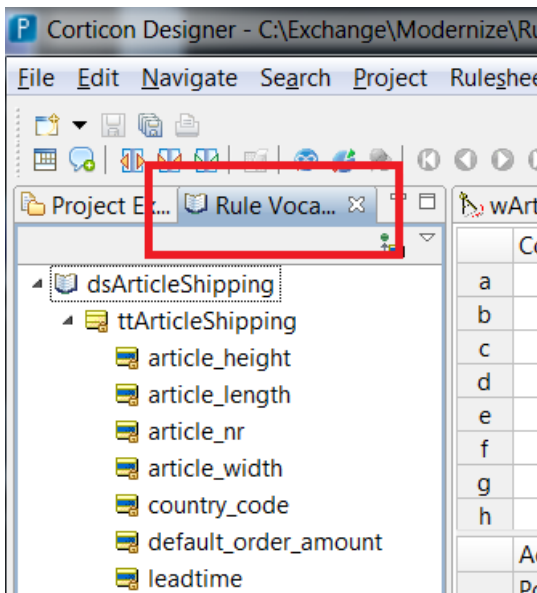
Check that the 'Corticon Designer' perspective is active.
If not selected, then select it.



Now we can start modeling our rules.

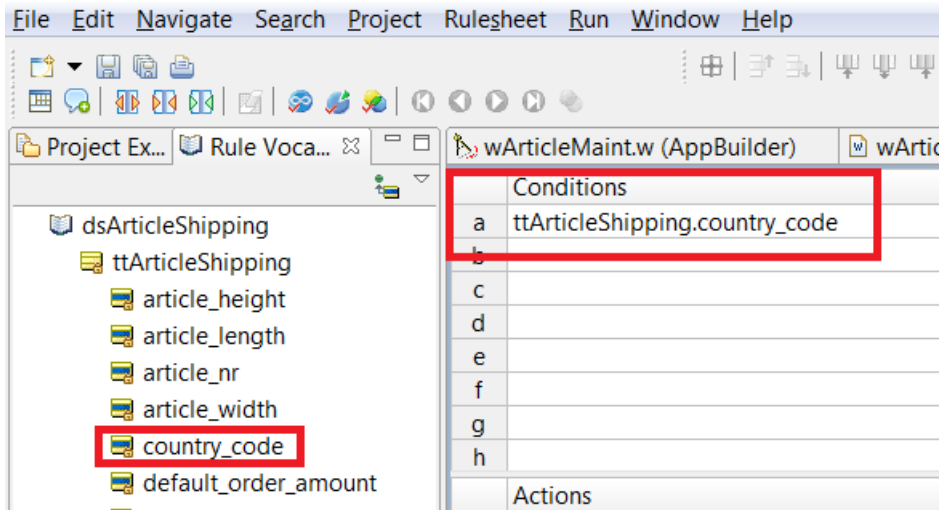
The rule should be that:
For France and extra fee of \$12 will be added.
For Canada and extra fee of \$25 will be added.
Other countries will have no extra fee.

Select the Rule Vocabulary view.
And expand the 'ttArticleShipping' tree.



Now drag and drop the 'country_code' from the tree to the first field in the conditions browser.

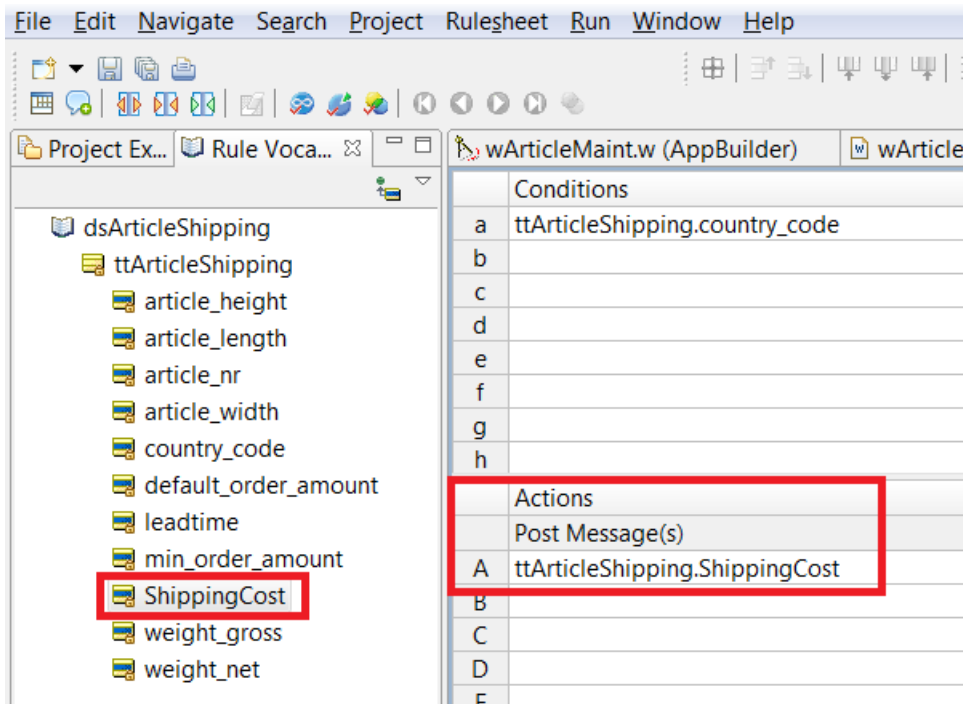
This means on this field we will add a condition check for the country.



Add the 'ShippingCost' to the action field

Here we will define the extra cost for each possible condition.

You can also use drag and drop here.



Now we will define the possible conditions.

FR = shippingcost + 12

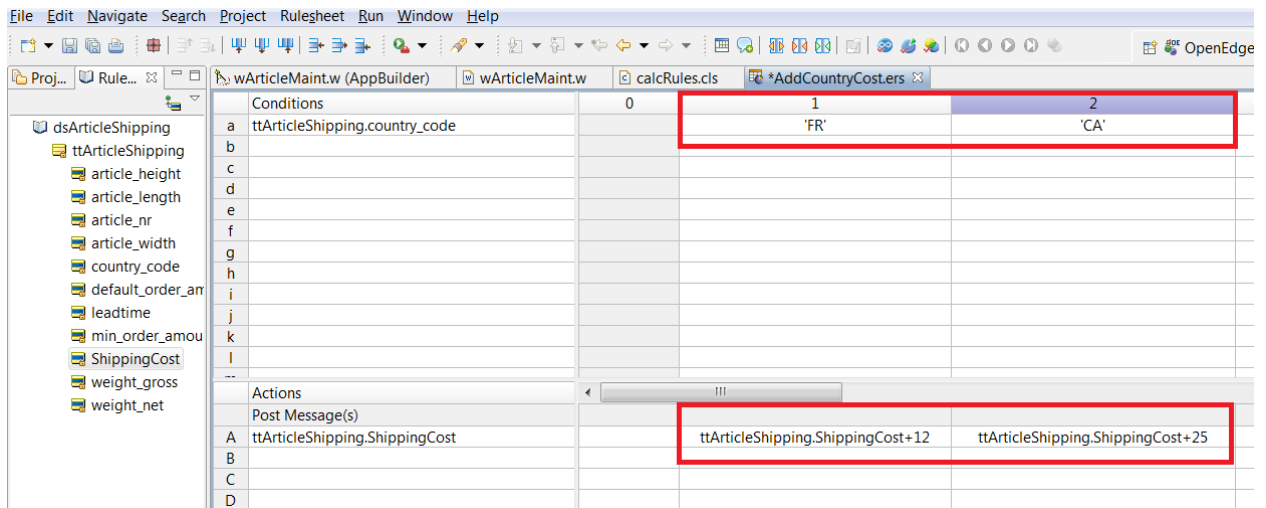
CA = shippingcost + 25.

! For the condition fields, just click in the cell and type the country code. Quotes will be added automatically.

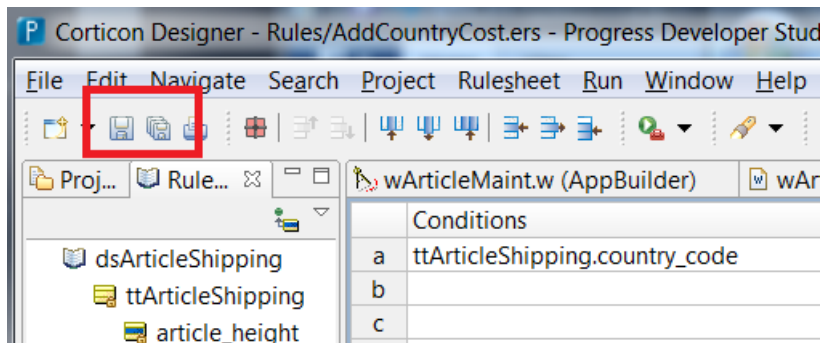
It maybe also shows a dropdown box but you can ignore this.

! For the calculation fields you can also drag and drop the field in column 1 and 2 then just add "+12" or "+25"

When it colors red it means there is something wrong.

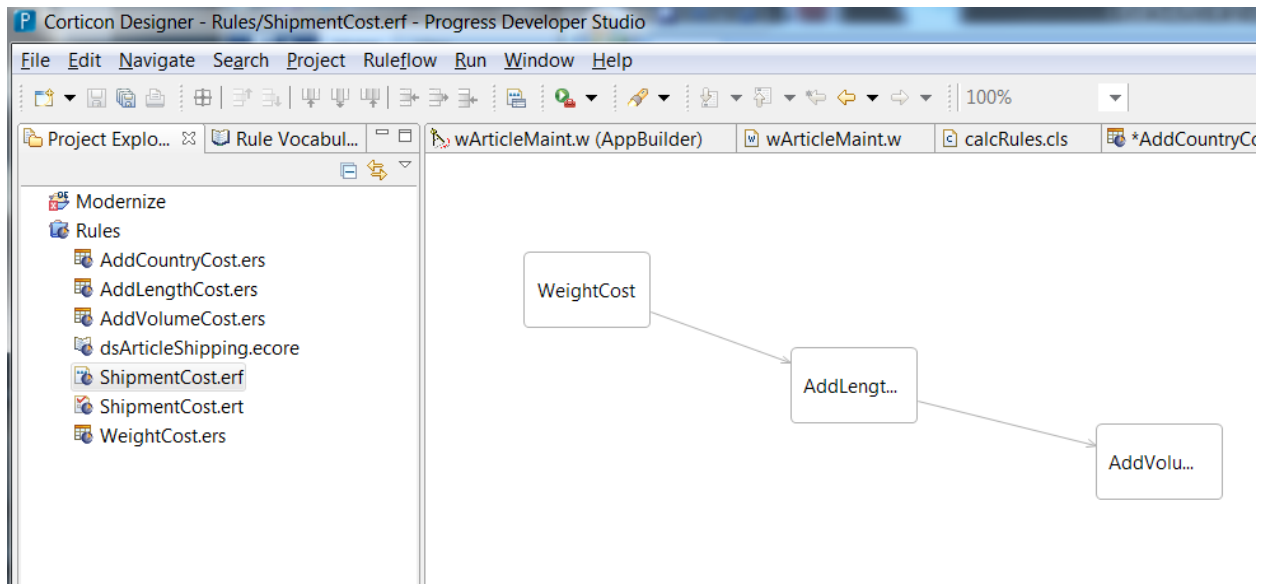
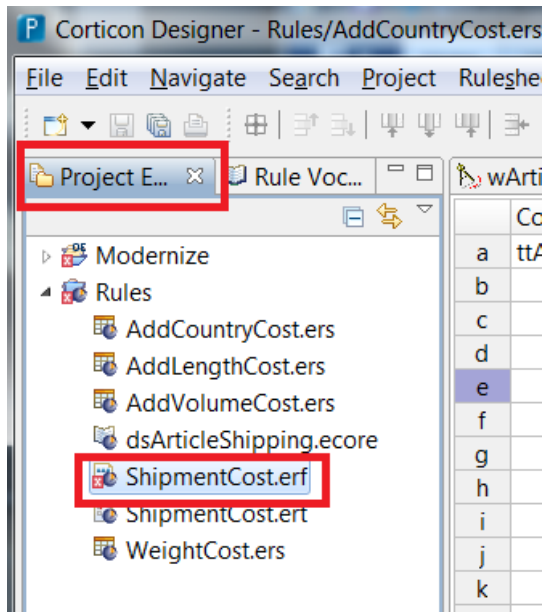


Save the rulesheet.

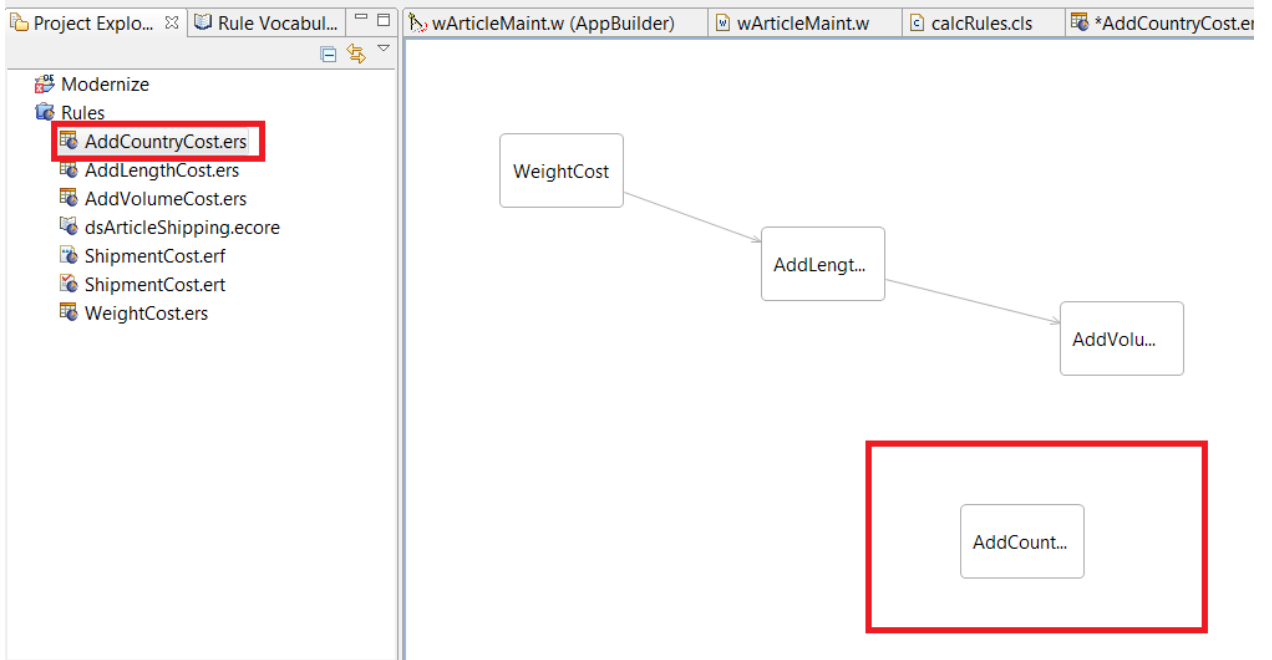


Add the new rulesheet to the existing rulflow.

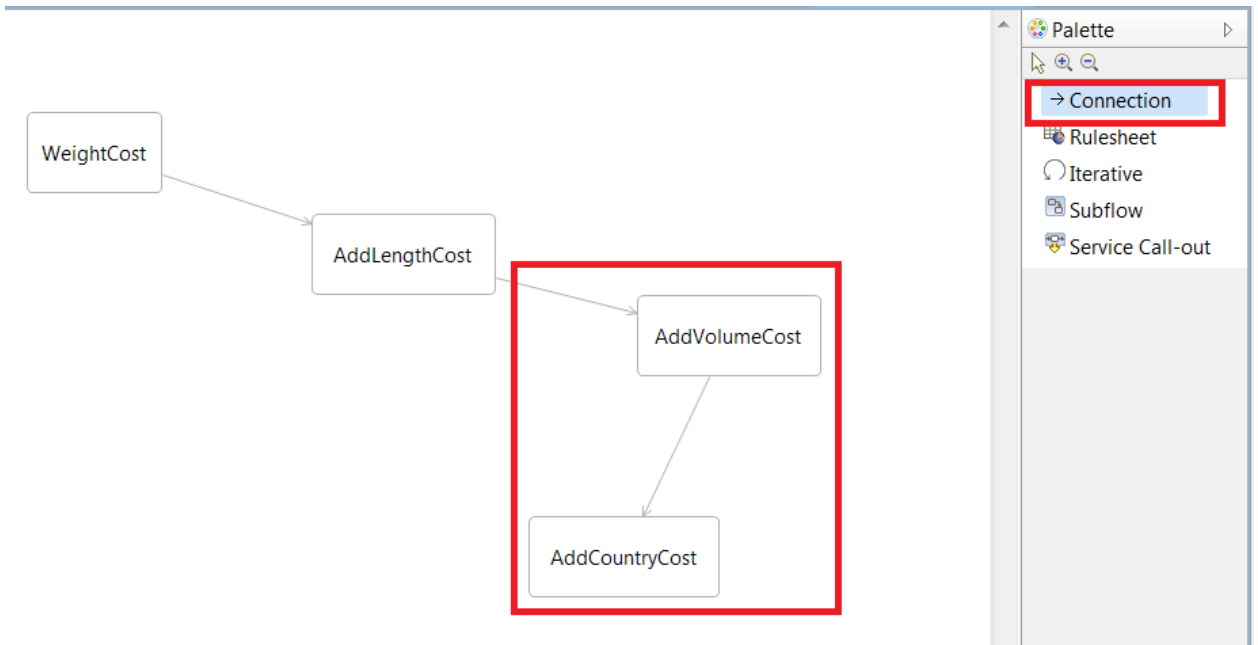
Go to the Project Explorer view again and open the 'ShipmentCost.ers' file



Drag and drop the new rulesheet 'AddCountryCost' on the flow page.



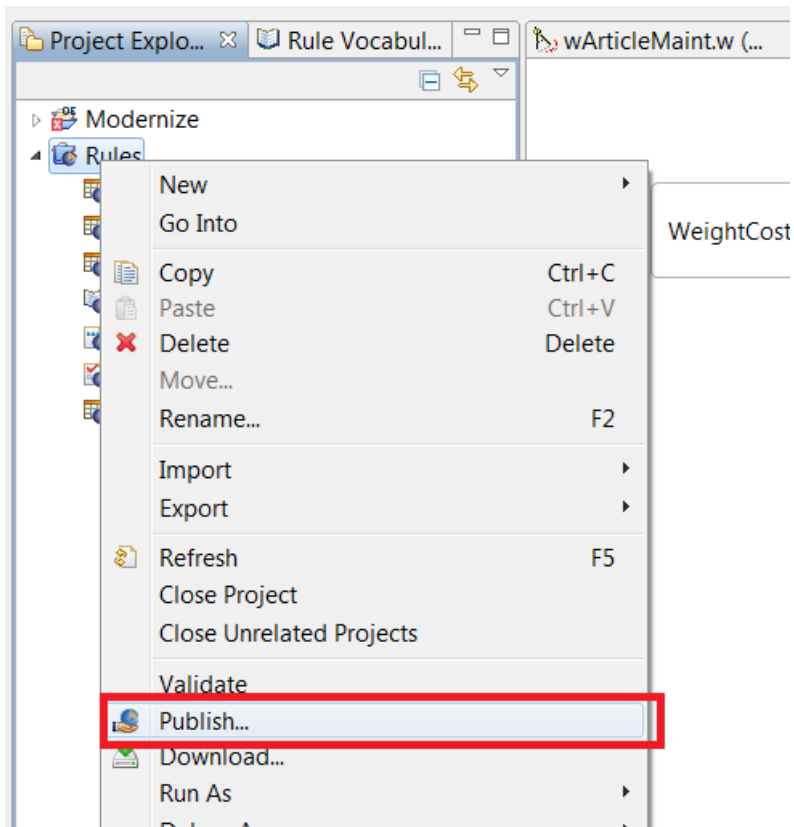
Now add a connection from the 'AddVolumeCost' to the 'AddCountryCost'.



Save the ruleflow.

Publish/ Activate the new shippingCost ruleflow.

Right-click the 'Rules' project and select publish.



Make connection to the rules engine.

Publish Decision Services

Server Connection

Specify the Server URL, username and password.

Server

Server:

Username:

Password:

Server : <http://localhost:8980/axis>

Username : admin

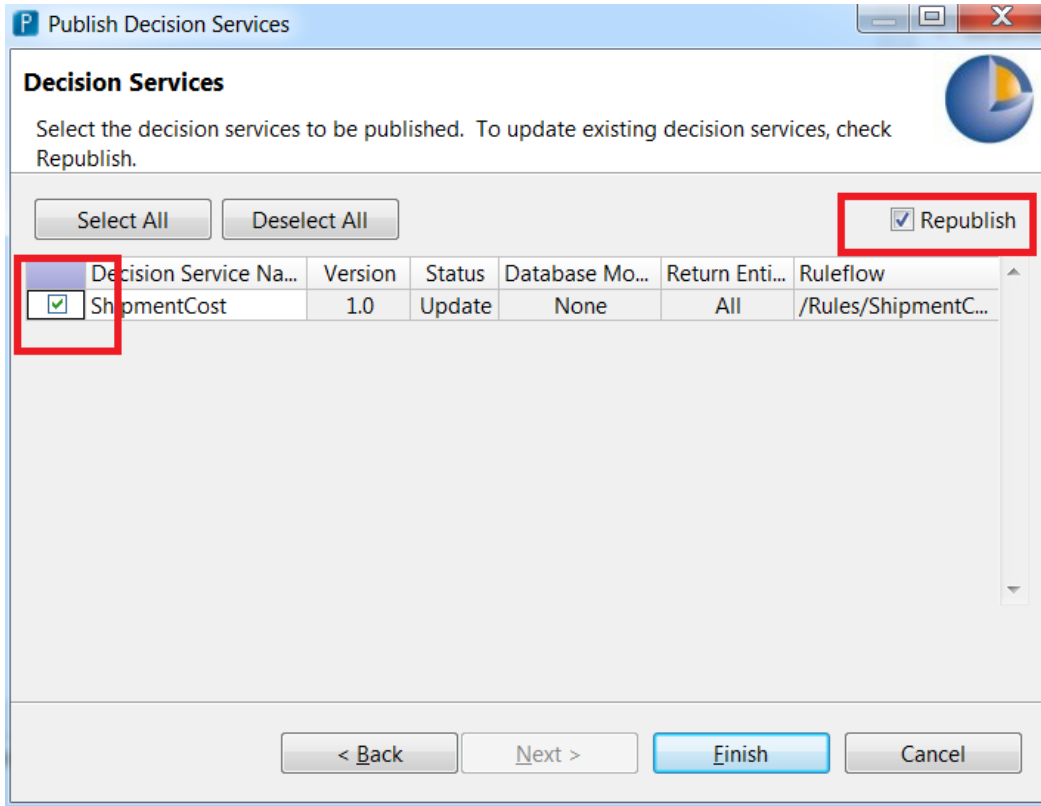
Password : admin

Try 'Test connection'

Press Next

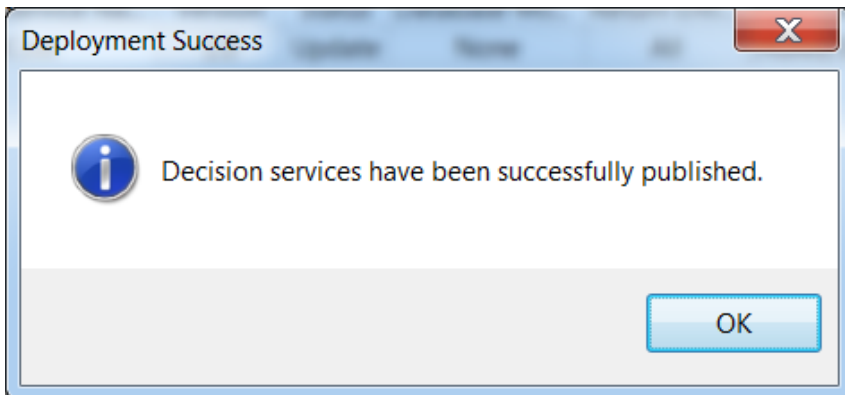
Toggle Republish

And toggle the 'ShippingCost' Service



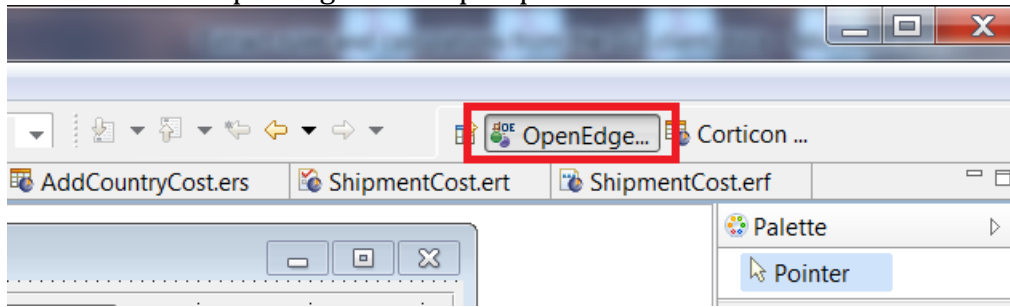
Press Finish.

You now should get the message that the service is published.



Test the CountryCost rule from the old UI

Go back to the 'OpenEdge Editor' perspective.



Select the 'wArticleMaint.w' program tab, or open the 'wArticleMaint.w' file from the Modernize project.

Run the project and test the 'Corticon Shipping Cost' function.

Select an article in the browser where the 'Country of origin' is France and compare both results of the rules.

The new rule should have an extra cost of \$12.

Buyer and Merchandise Administration

Article Number: Status: Filter

Article Number	Article Description	Supplier Name	Status
1	Cabernet Sauvignon	Wines International	Administered
2	Special Spirits	Spirits Co.	new
3	White stuff above 80°	Moonshine Inc	Administered
4	Alco pop Red	Spirits Co.	New
7	dddd	Wines International	New
8	magic stuff	Wines International	New

Article Supplier Merchandising

Supplier

Article Number: 1

Supplier Code: WINE001 [Wines International](#)

Country of Origin: France

Net Weight: 10,00 Gross Weight: 11,00

Length: 8,00 Width: 200,00 Height: 200,00

Website URL:

Predict Shipping Cost 265,60

Corticon Shipping Cost 277,60

Update Save Cancel

You can now update and save the values on the Supplier tab and see the difference in rule calculations.

Extra work

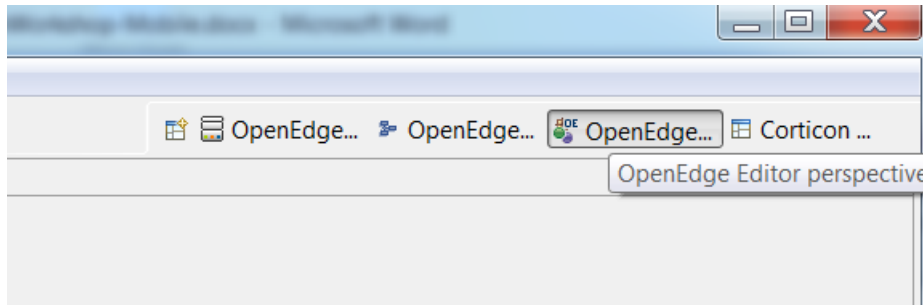
Add rule messages in Corticon

Show rule messages as tooltip in the GUI

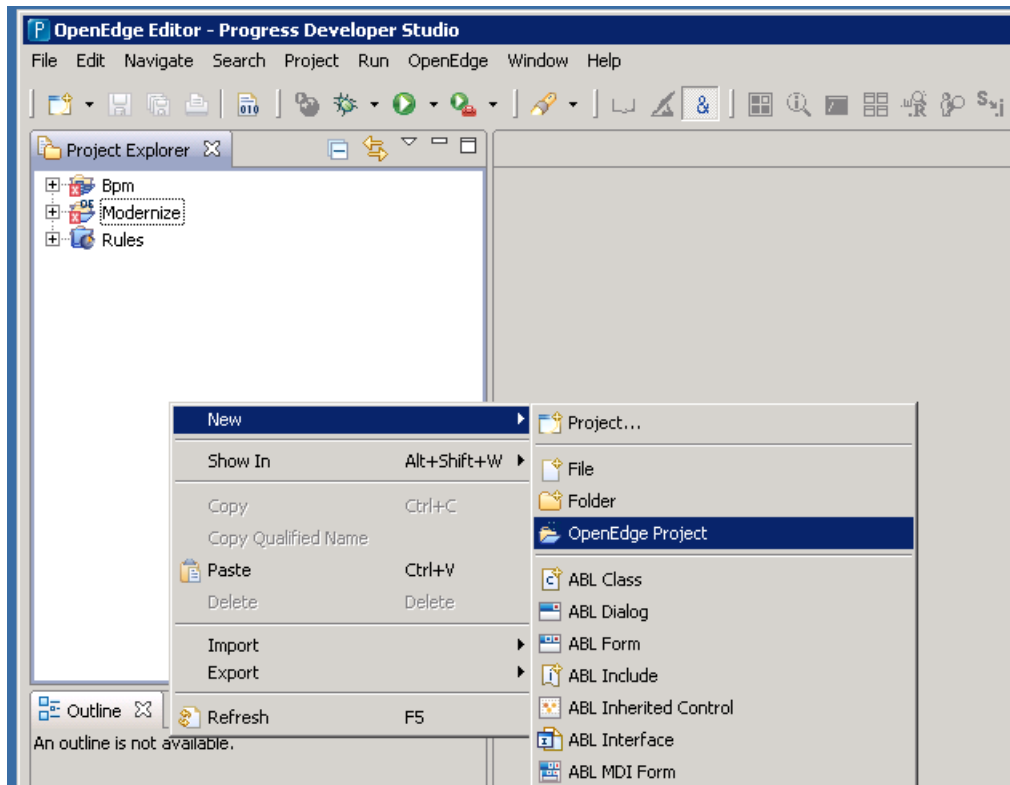
Chapter 4 – Implementing OE-Mobile screen

Mobile Project Setup

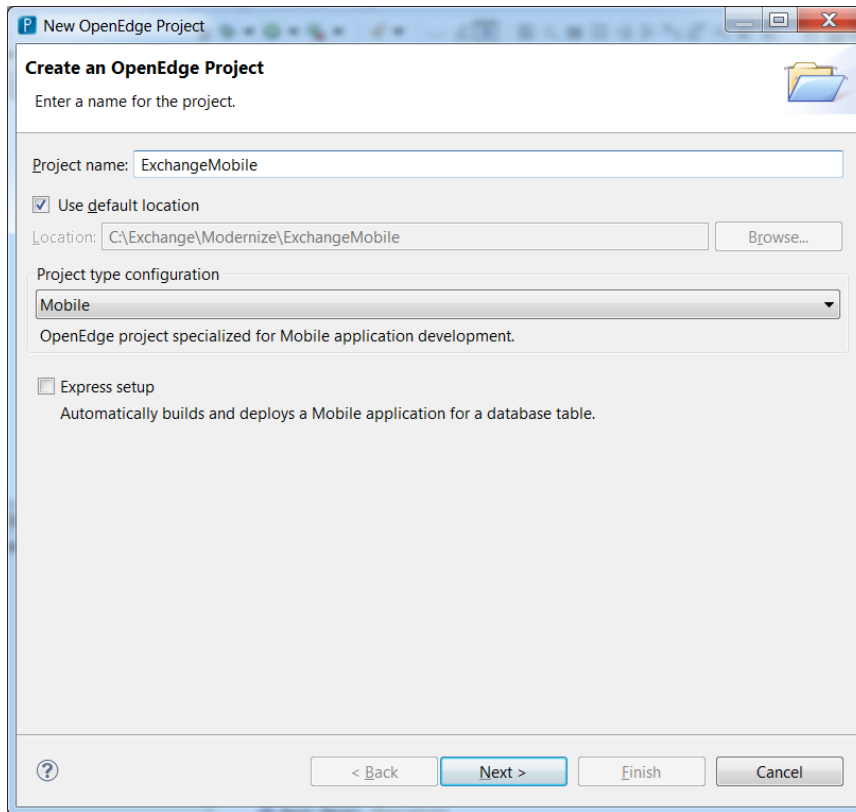
Select the OpenEdge Editor perspective.



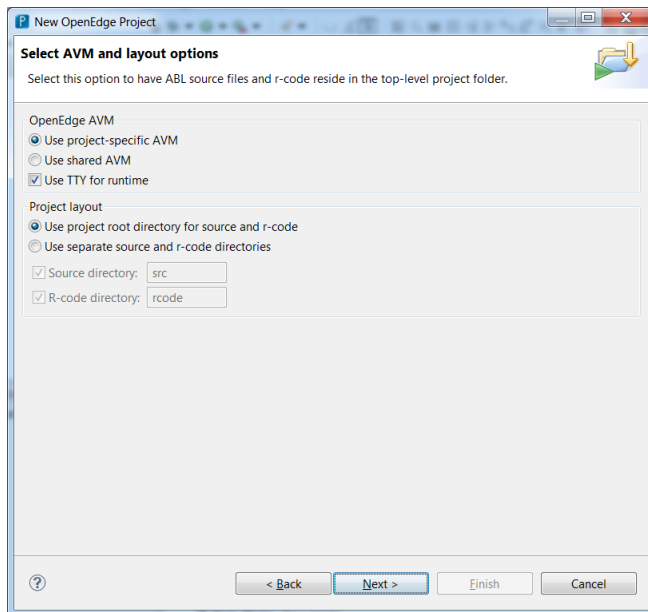
Create a new Mobile project by right-click in the Project Explorer view, choose New and OpenEdge Project.



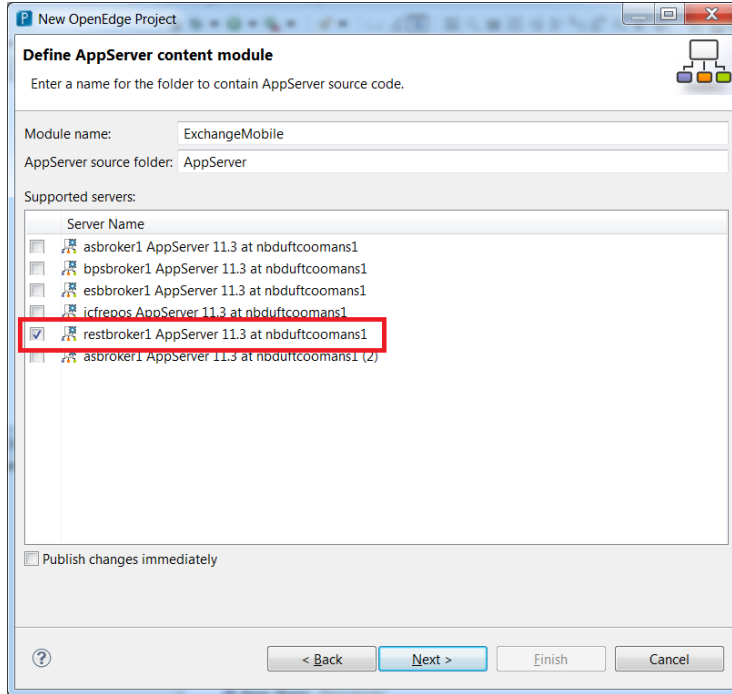
Project Name will be 'ExchangeMobile'
Project type configuration will be 'Mobile'
The click Next



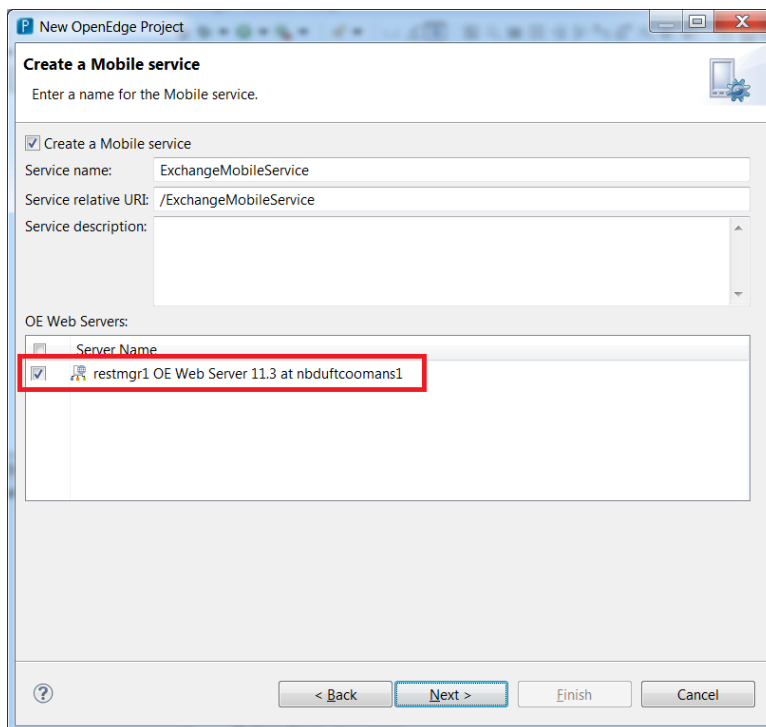
Next again



Select 'restbroker1' if not yet selected. This is the application server where your backend code will run on. Communication to the mobile app will be done via REST.
Click Next

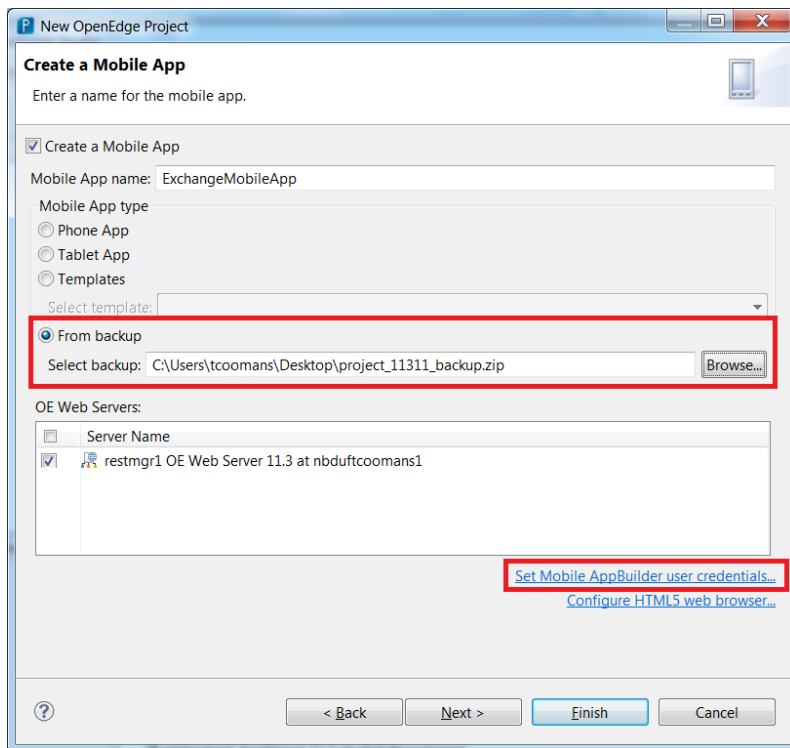


Select 'restmgr1 OE Web Server'. On this webserver will the REST services be deployed.
Click Next

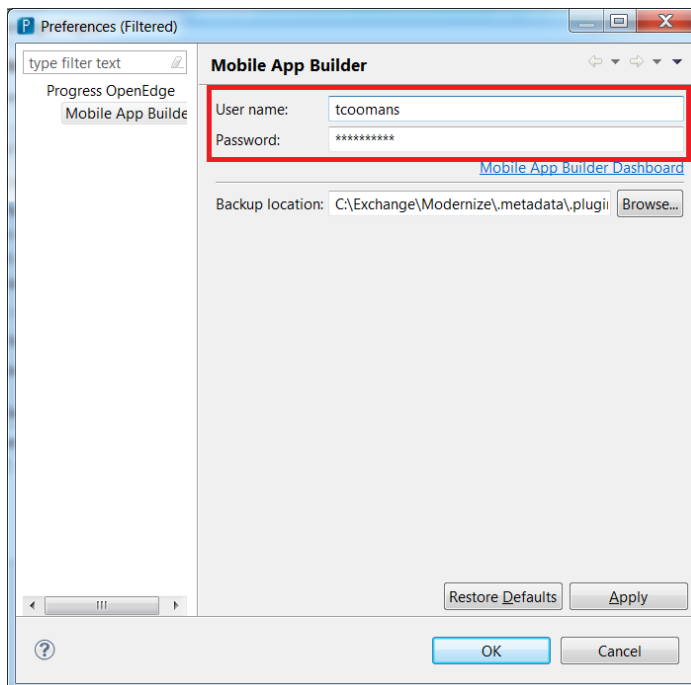


Select 'From Backup'

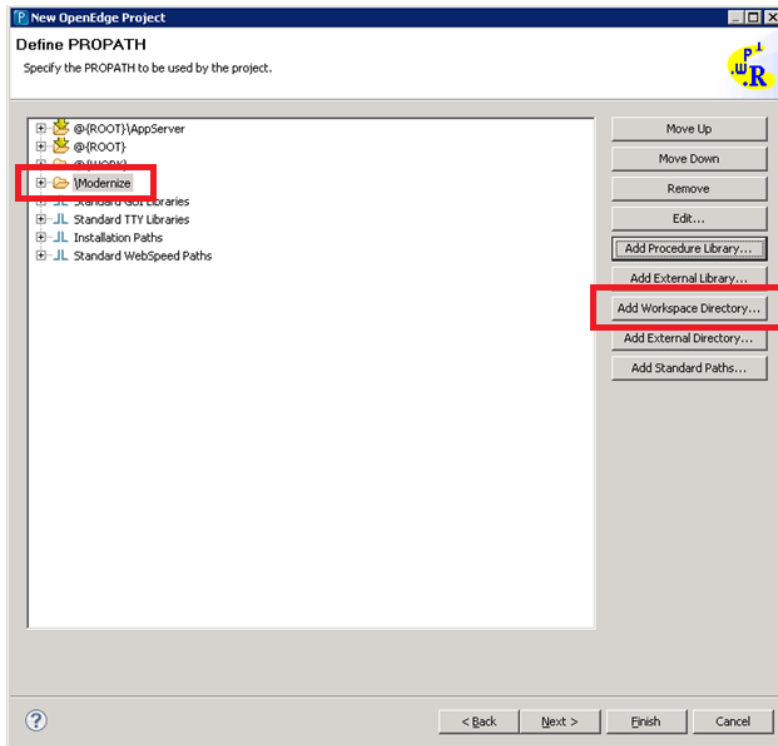
And browse to your desktop where a 'mobile-start' folder is, select the zip file in this folder. Then go to 'Set Mobile AppBuilder user credentials'.



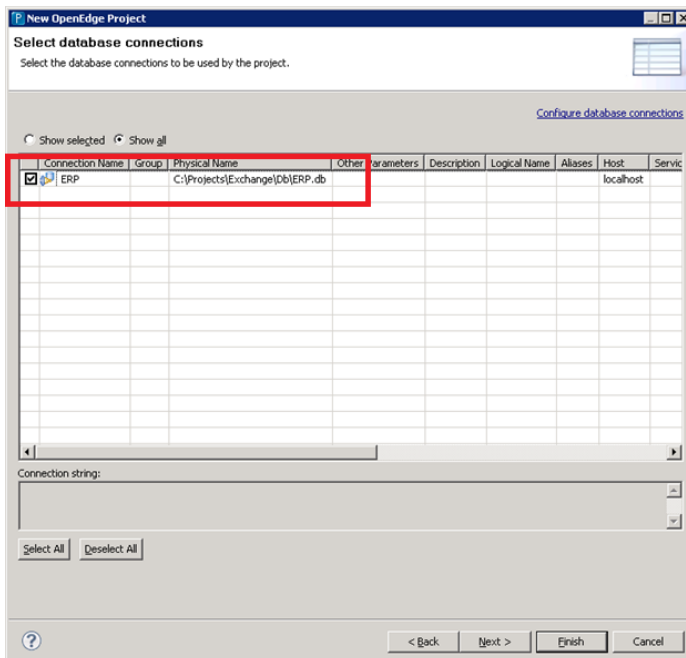
Add here your psdn account credentials or the provided mobile credentials. ! If you do not have a psdn account yet. See Appendix A how to create one. Click OK and Finish.



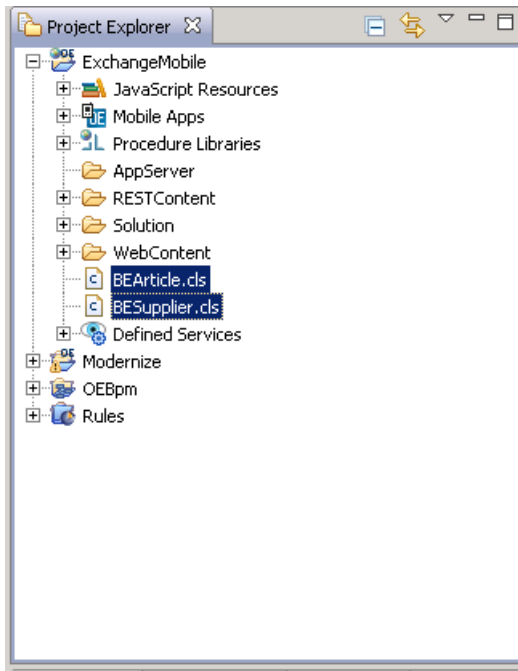
Add the 'Modernize' project to your propath. This is needed because we have our data definitions there and some backend code which is used. Choose 'Add Workspace Directory' and choose 'Modernize' project. Press 'Move Up' until it is located behind the "@{WORK}" entry.



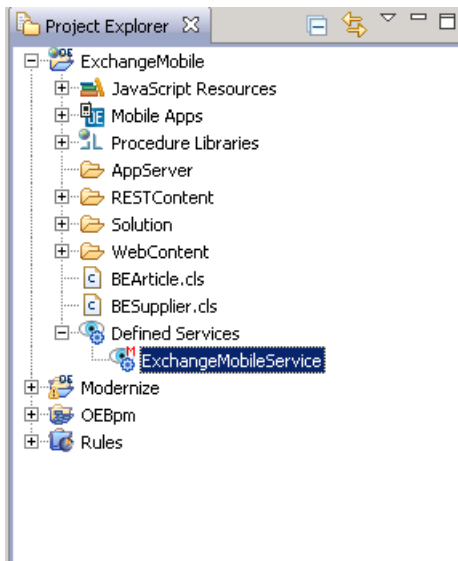
Select the 'ERP' database.



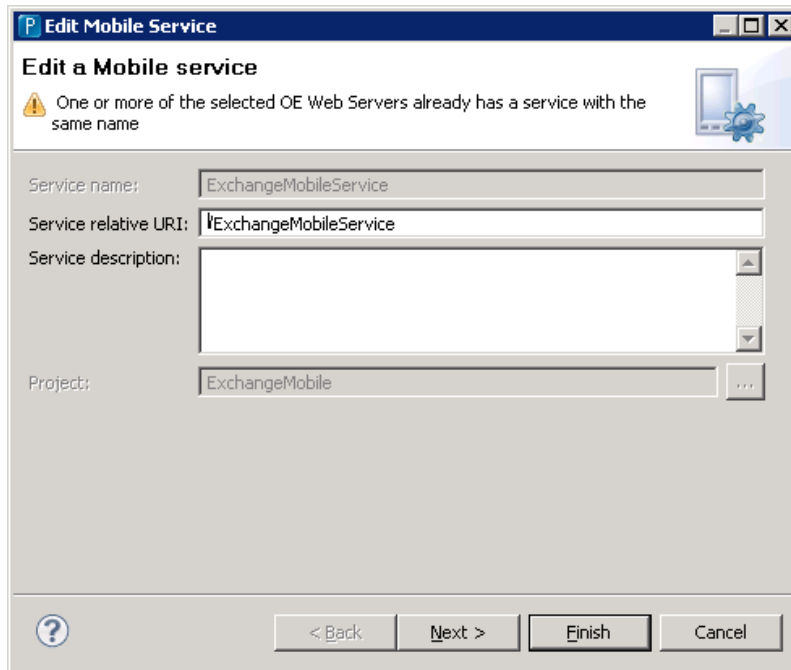
Copy the 2 BE*.cls files from the “mobile-start” folder on your desktop to your ‘mobile project’



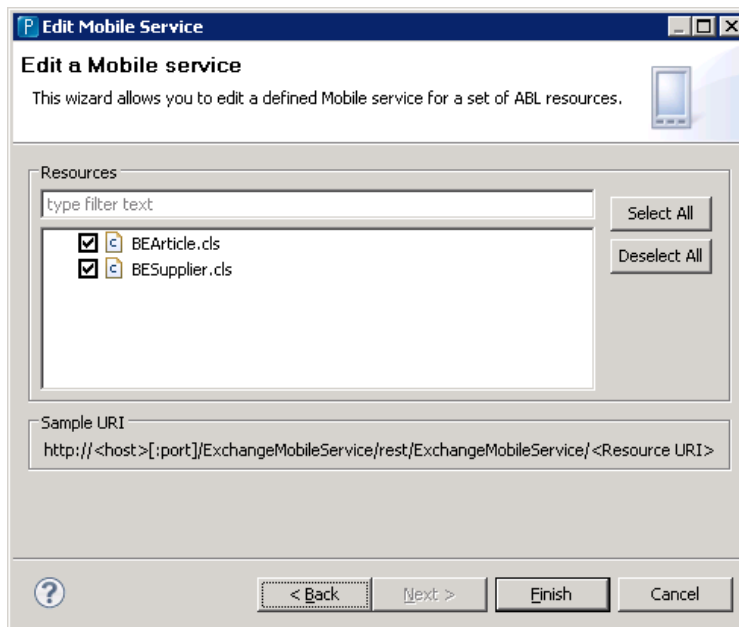
Generate REST services.
Go to 'Defined Services' and double-click 'ExchangeMobileService'.



Click Next



Select the BEArticle and BESupplier service and click Finish

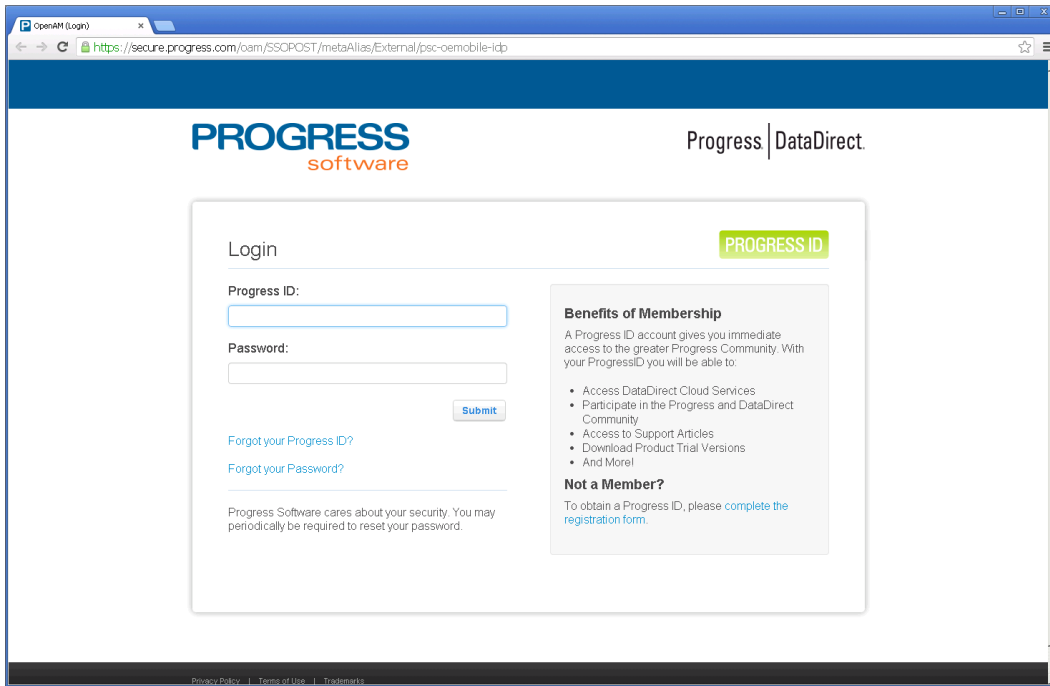


Add user-interface to app

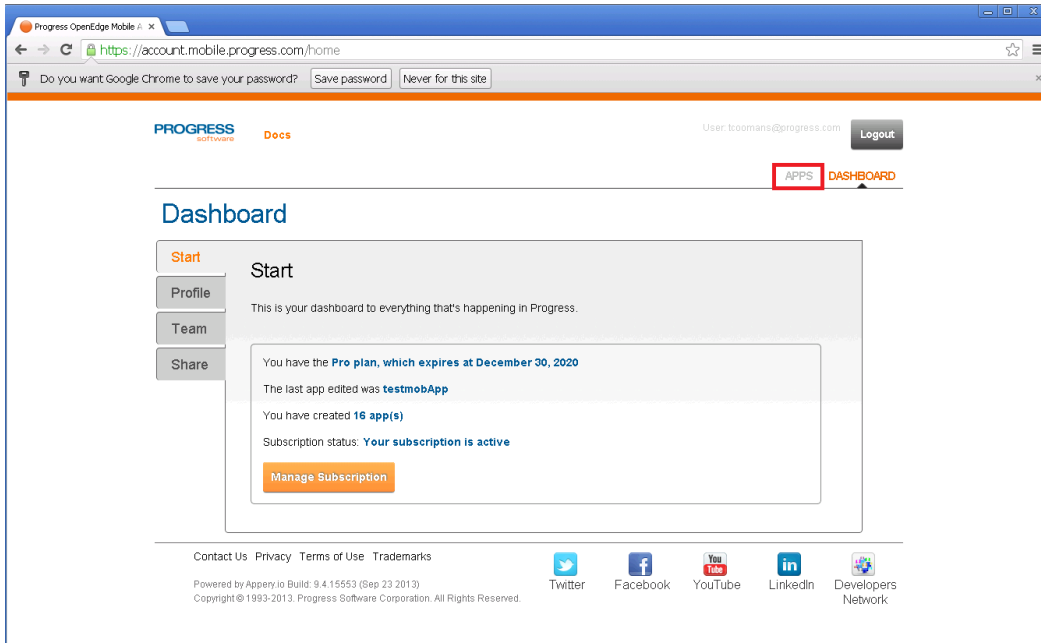
When the project is first opened in the Mobile AppBuilder it will appear as follows:

First Login again to open the mobile appbuilder. These are the same credentials as you used in your development environment.

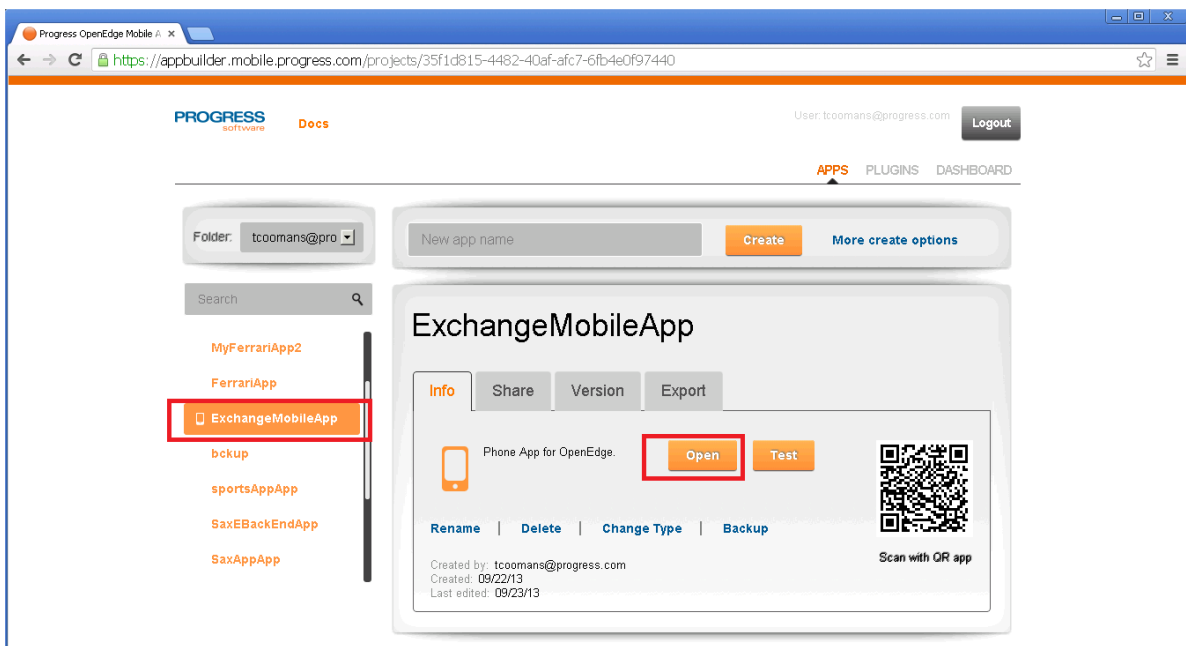
Mobile Appbuilder can always be started by using the URL <http://mobile.progress.com> in your webbrowser.



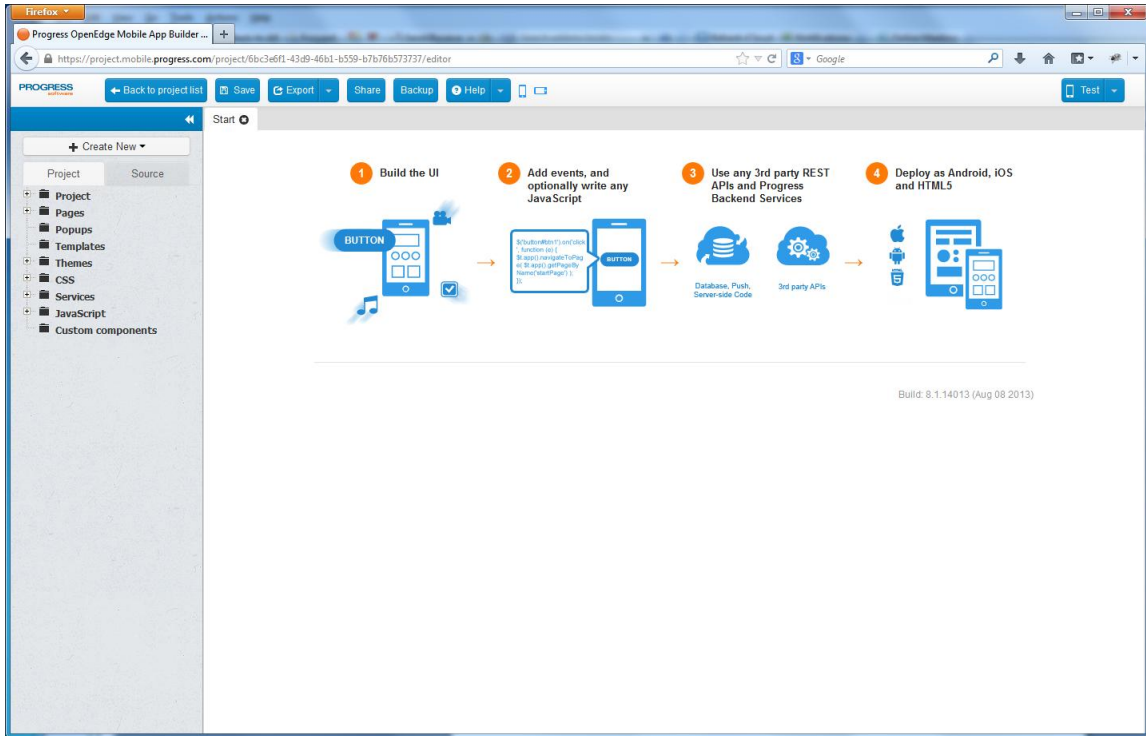
Depending if you used Mobile Appbuilder before you will get some initial screens. Choose "APPS"



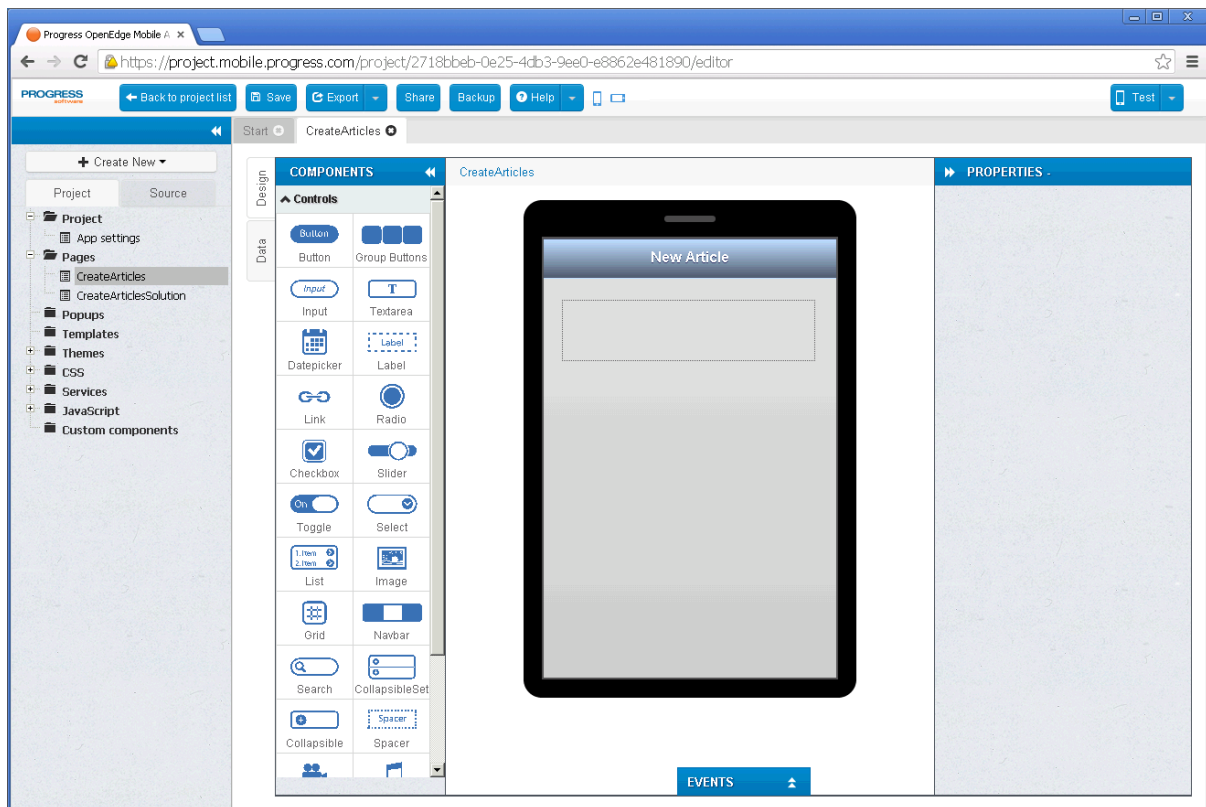
Select the “ExchangeMobileApp” and press open.



You now will see the welcome screen of the mobile Appbuilder.

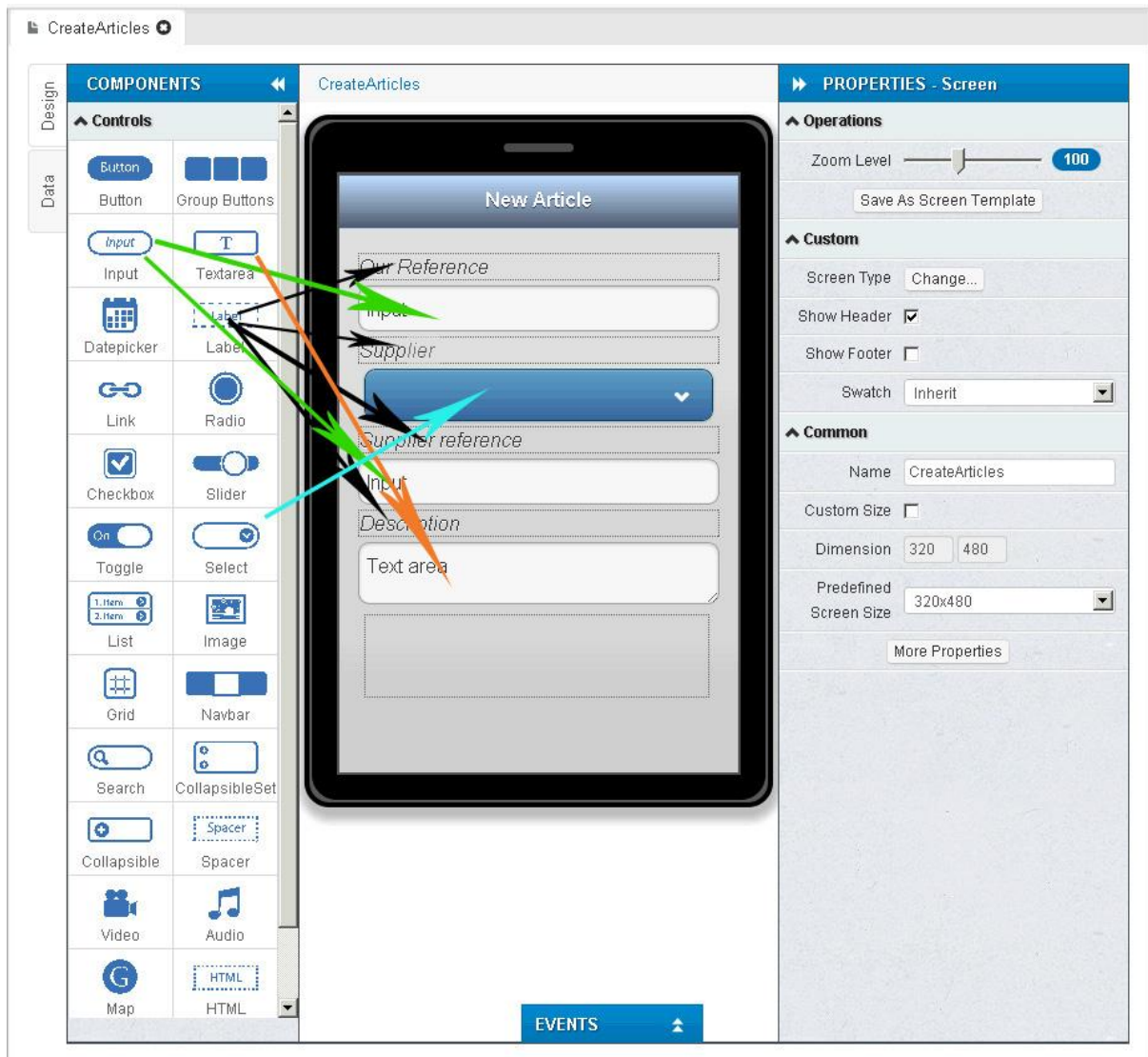


Expand the pages branch of the tree on the left side and select the CreateArticles page. The page should then appear as follows:

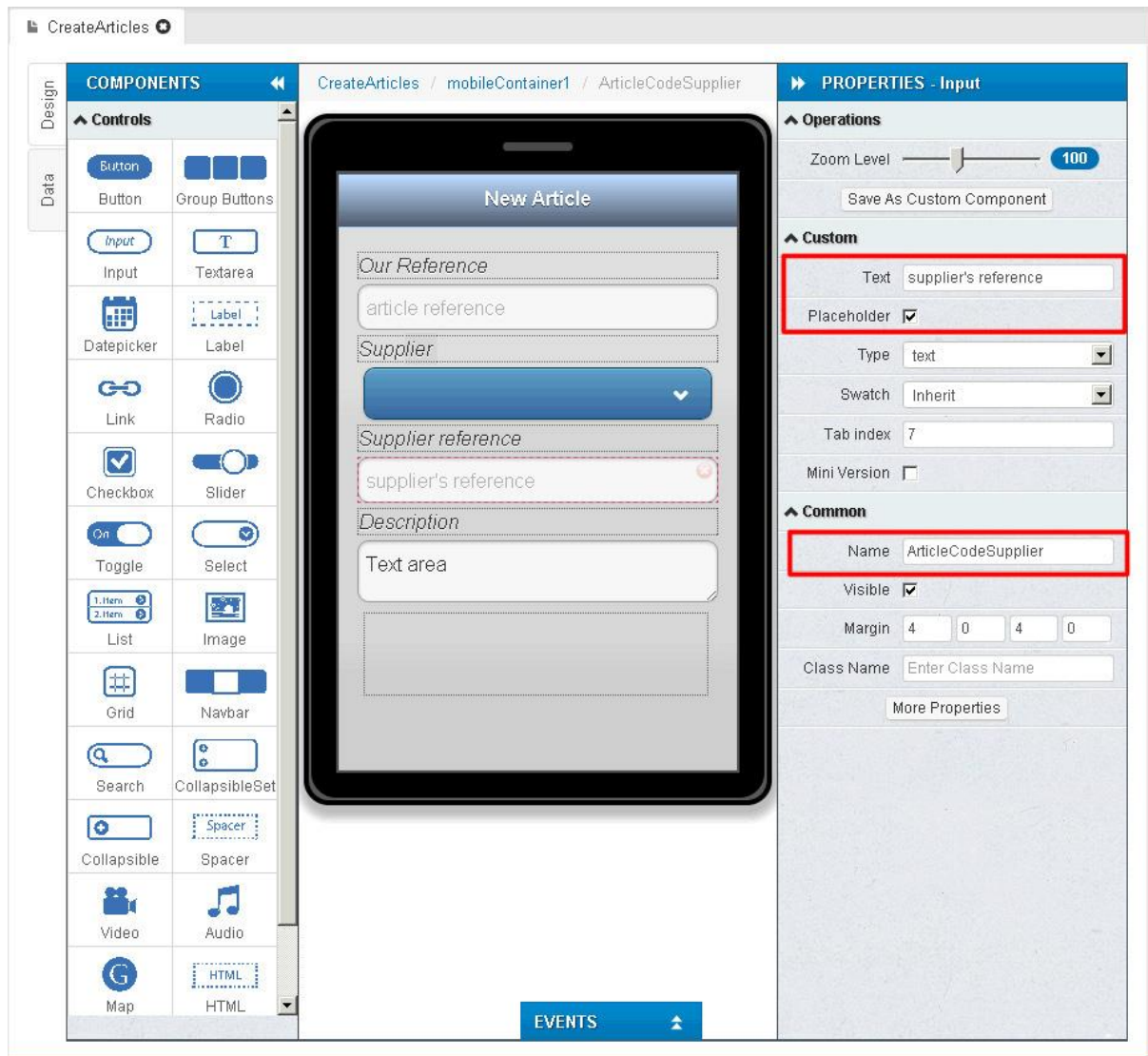


Here we have a almost blank page. The only item on it is a panel that is used for messages. The service definitions needed for this page to interact with the OpenEdge REST Adapter are defined. You now need to add the user-interface elements and map them to REST requests and responses.

Begin by adding (drag from the palette onto the page) the fields shown in the picture below. The arrows indicate which component types to use.



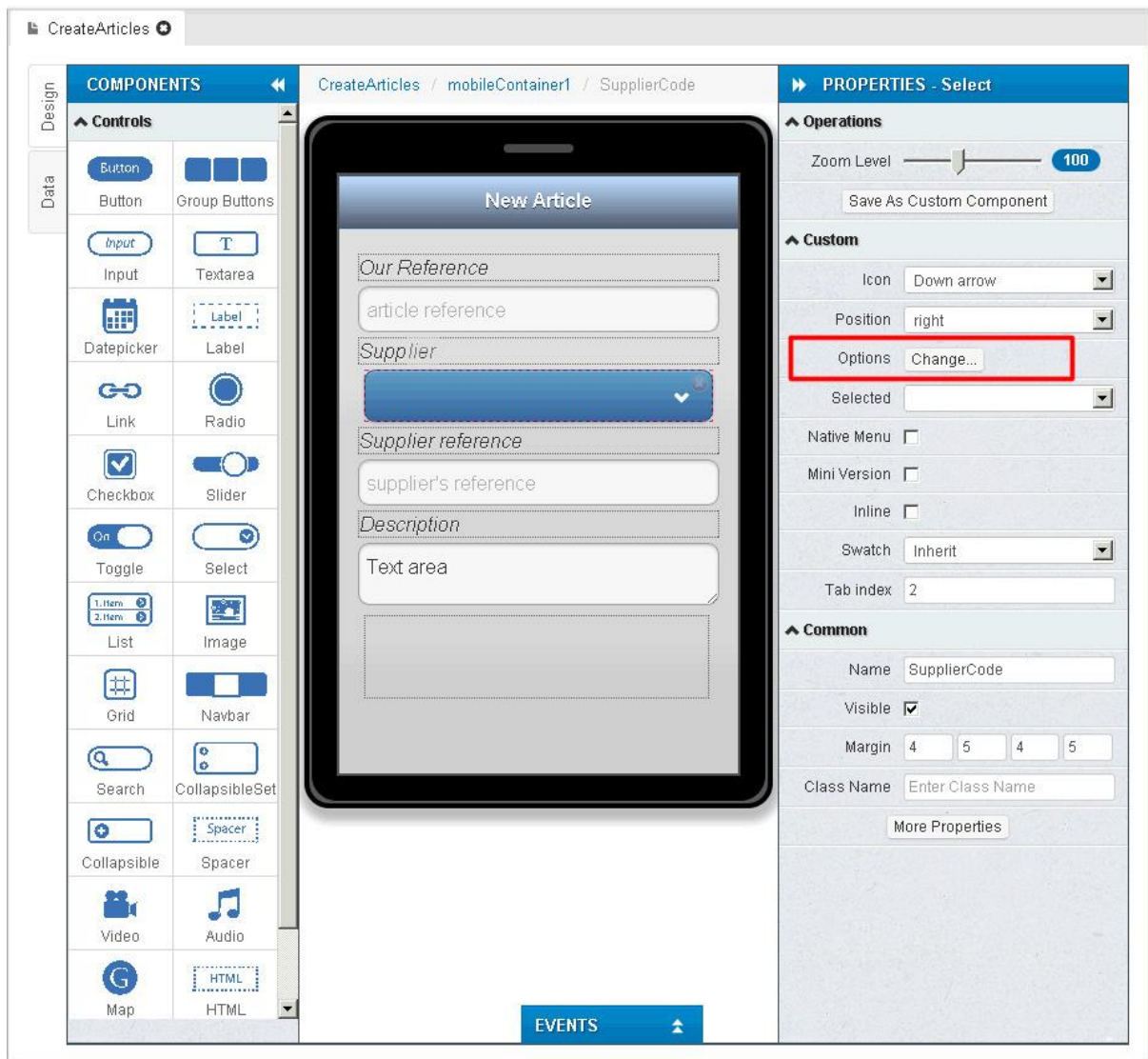
For the input elements, set their name and placeholder text, e.g.



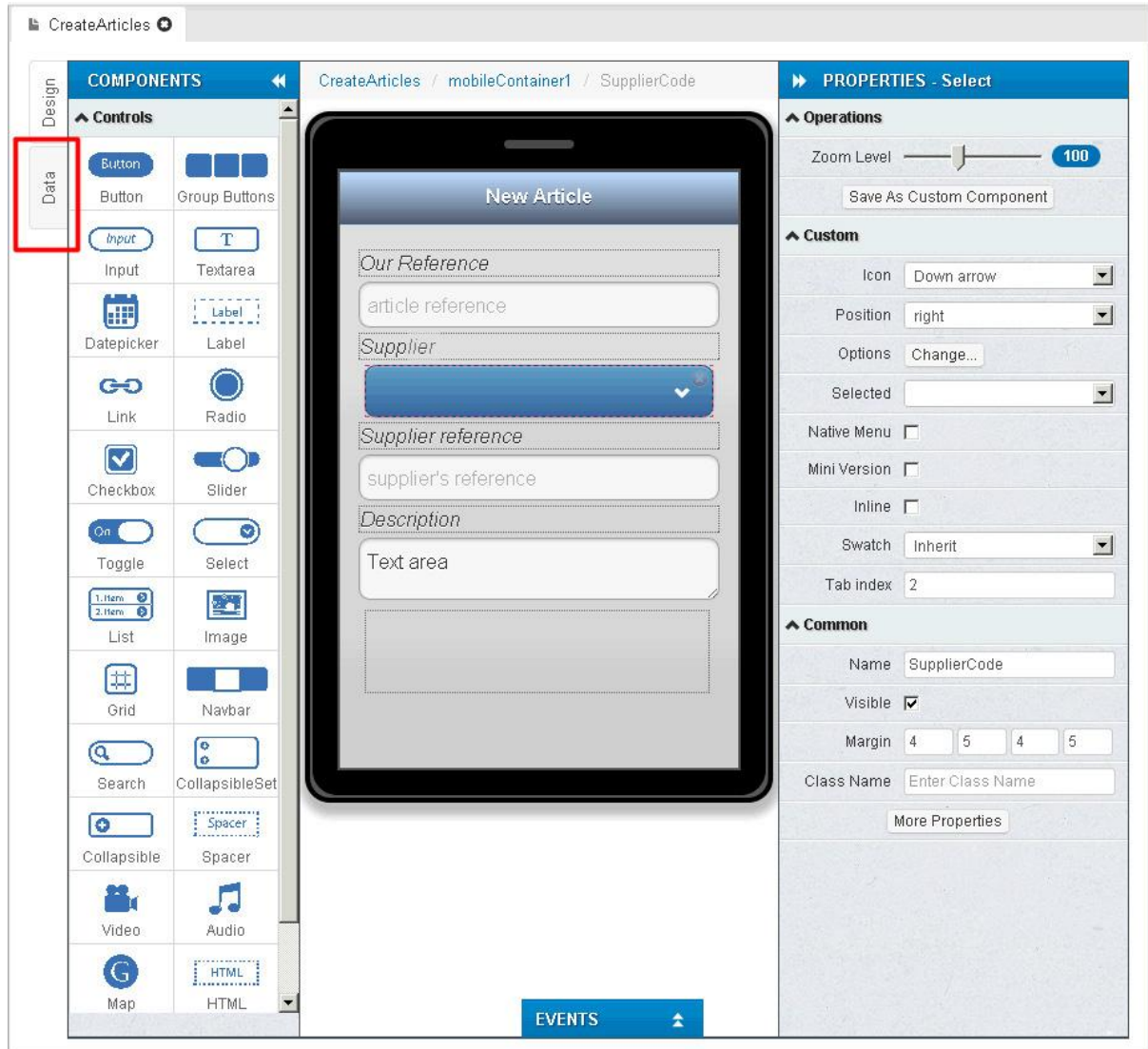
Choose placeholders and names of your choice. This will make it more clear later when you need to map the screen fields with the data-object.

! Don't use spaces in the "Name" property

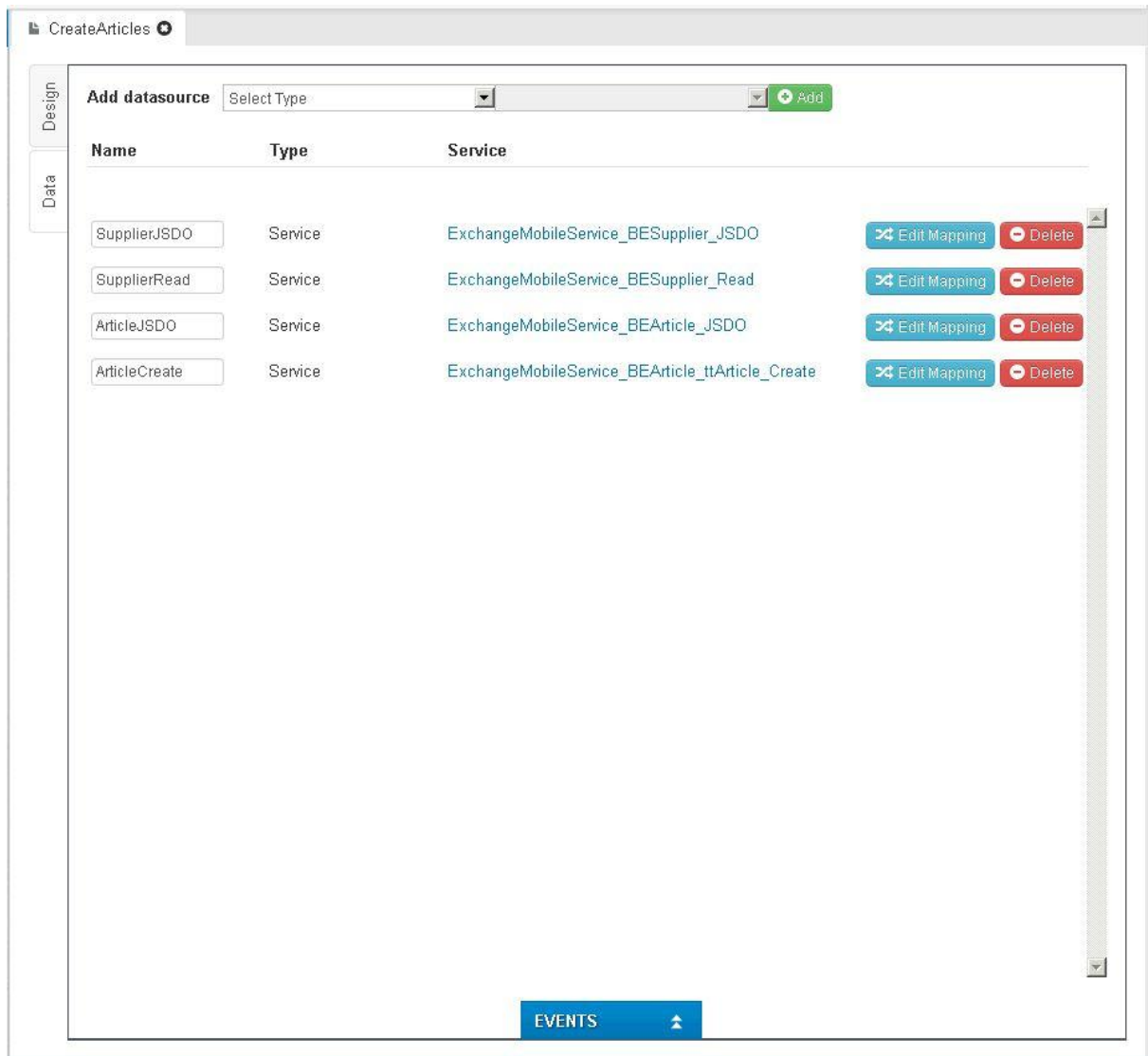
The Supplier select element will have default values in, since this will be populated from the database, the default options can be removed using the 'options...' button:



With the screen elements created, they now have to be linked to the services that interact with the OpenEdge server. To get to these to click on the Data tab, to the left of the Component palette, and indicated below:



The page should appear below.



We first want to populate on the Supplier select and to do this click on the 'Edit Mappings' button for the "ExchangeMobileService_BESupplier_Read" service.

No values are sent to the server to filter the request, we just want to make a request and get all the suppliers back. We do, however, need to define what happens to the suppliers that are returned and we do this by mapping the responses, which is done on the “Response” tab, circled below:

The screenshot shows the configuration interface for a service named "CreateArticles". On the left, under the "Data" tab, the "Response" tab is selected and highlighted with a red circle. Below it, a list of service parameters is shown:

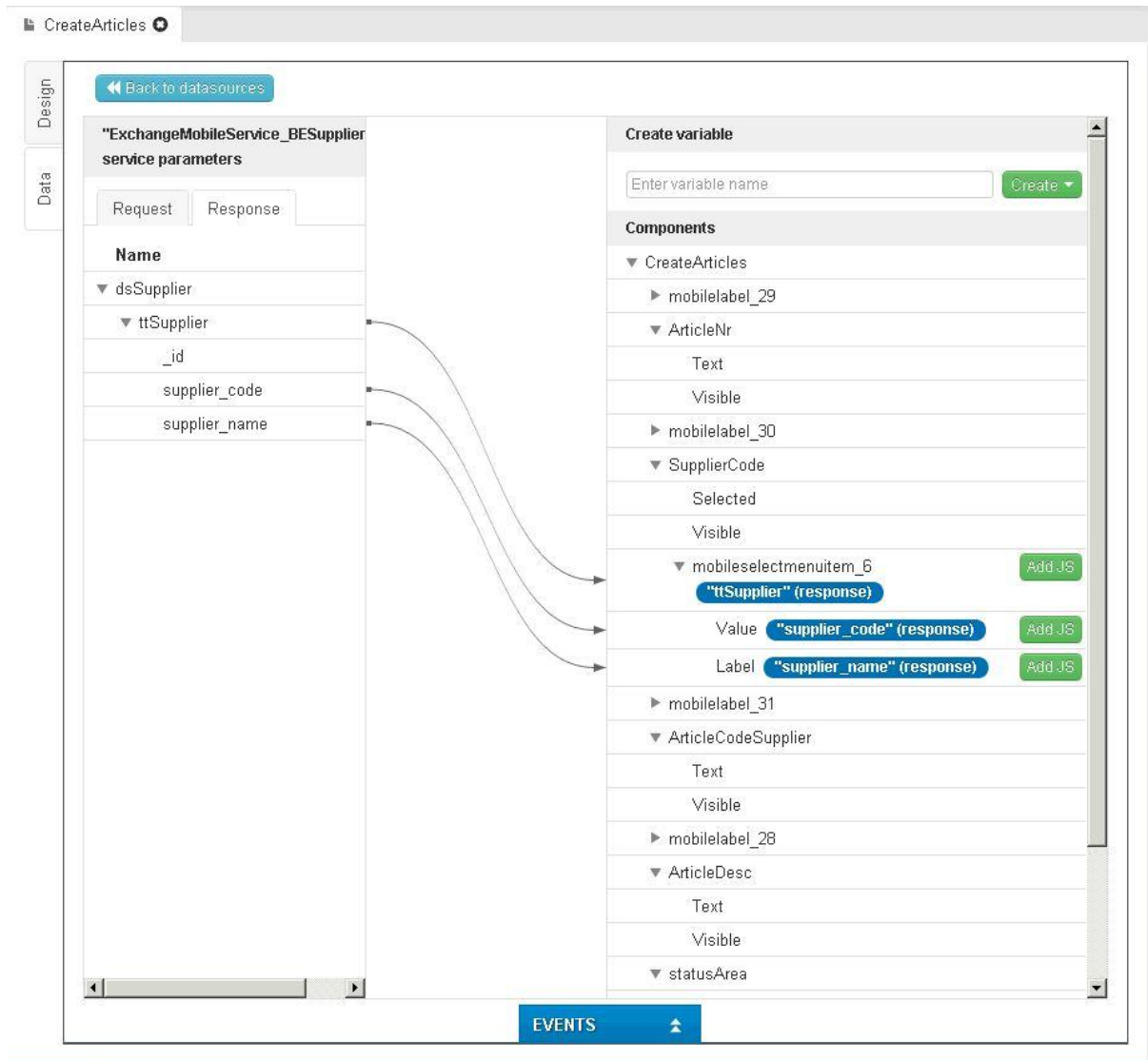
Name
dsSupplier
ttSupplier
_id
supplier_code
supplier_name

On the right, the "Components" list shows a tree structure of UI elements:

- ▼ CreateArticles
 - ▶ mobilelabel_29
 - ▼ ArticleNr
 - Text
 - Visible
 - ▶ mobilelabel_30
 - ▼ SupplierCode
 - Selected
 - Visible
 - ▼ mobileselectmenuitem_6
 - Value
 - Label
 - ▶ mobilelabel_31
 - ▼ ArticleCodeSupplier
 - Text
 - Visible
 - ▶ mobilelabel_28
 - ▼ ArticleDesc
 - Text
 - Visible
 - ▼ statusArea
 - HTML_code

The names that appear in the components area on the right of your page may differ to that shown above, since it will show the names you chose.

On both sides of the responses area, expand the trees to see the lowest levels and map the response and user-interface elements, as shown below by drag and drop from left pane to the right pane for ttSupplier to “mobileselectmenuitem_6” and supplier_code to “Value” and supplier_name to “Label”:



Having done the above, save the project (by pressing the Save button on the toolbar and then the Test button.

Once the supplier select is working, we can do the mapping for sending new Articles to the server. Mappings to do are as follows.

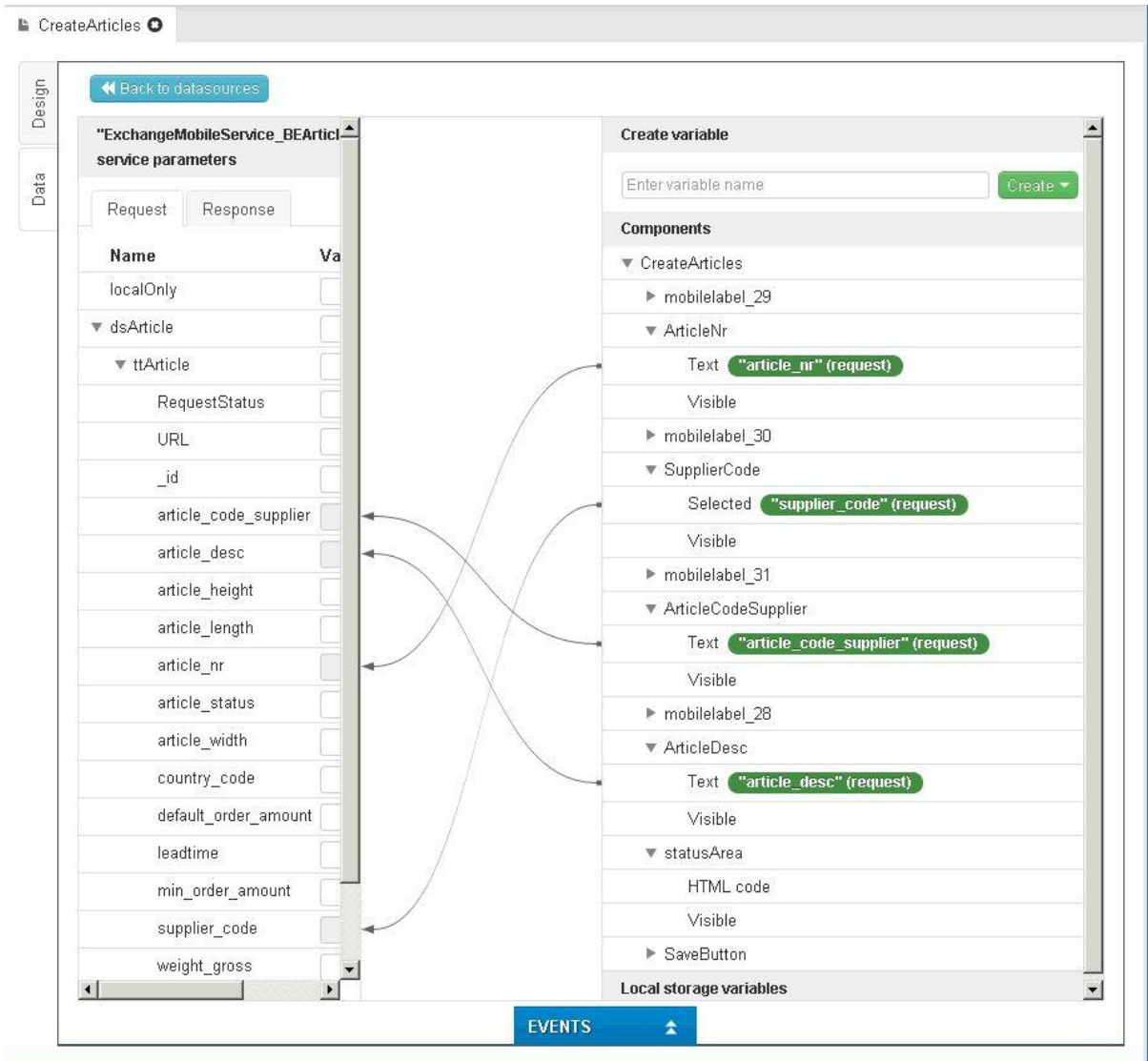
Select the "ExchangeMobileService_BEArticle_ttArticle_Create" from the Datasource and select the "Request" tab.

Map The ArticleNr from the right to "article_nr" on the left,

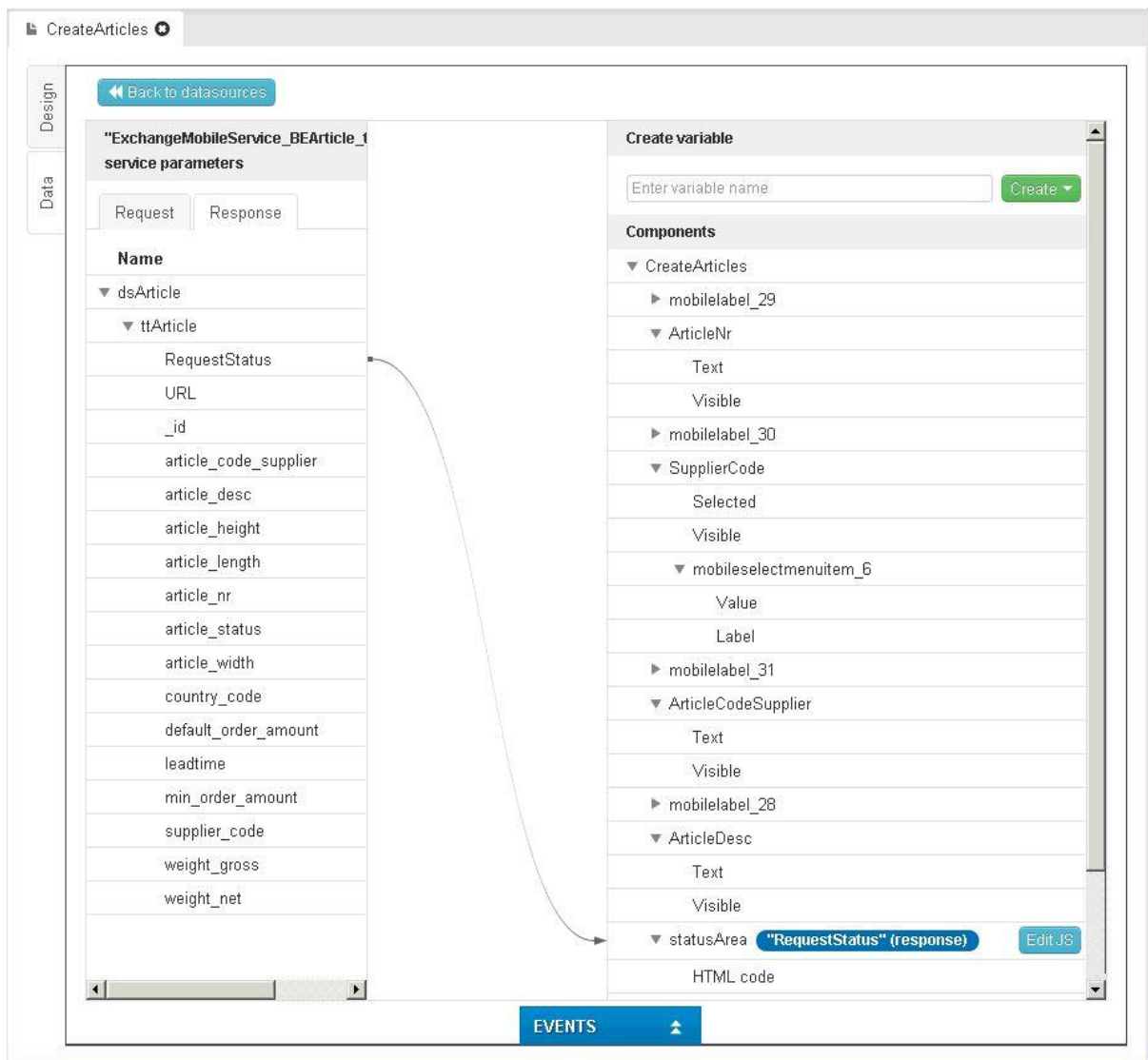
SupplierCode to supplier_code,

ArticleCodeSupplier to article_code_supplier and

ArticleDesc to article_desc.



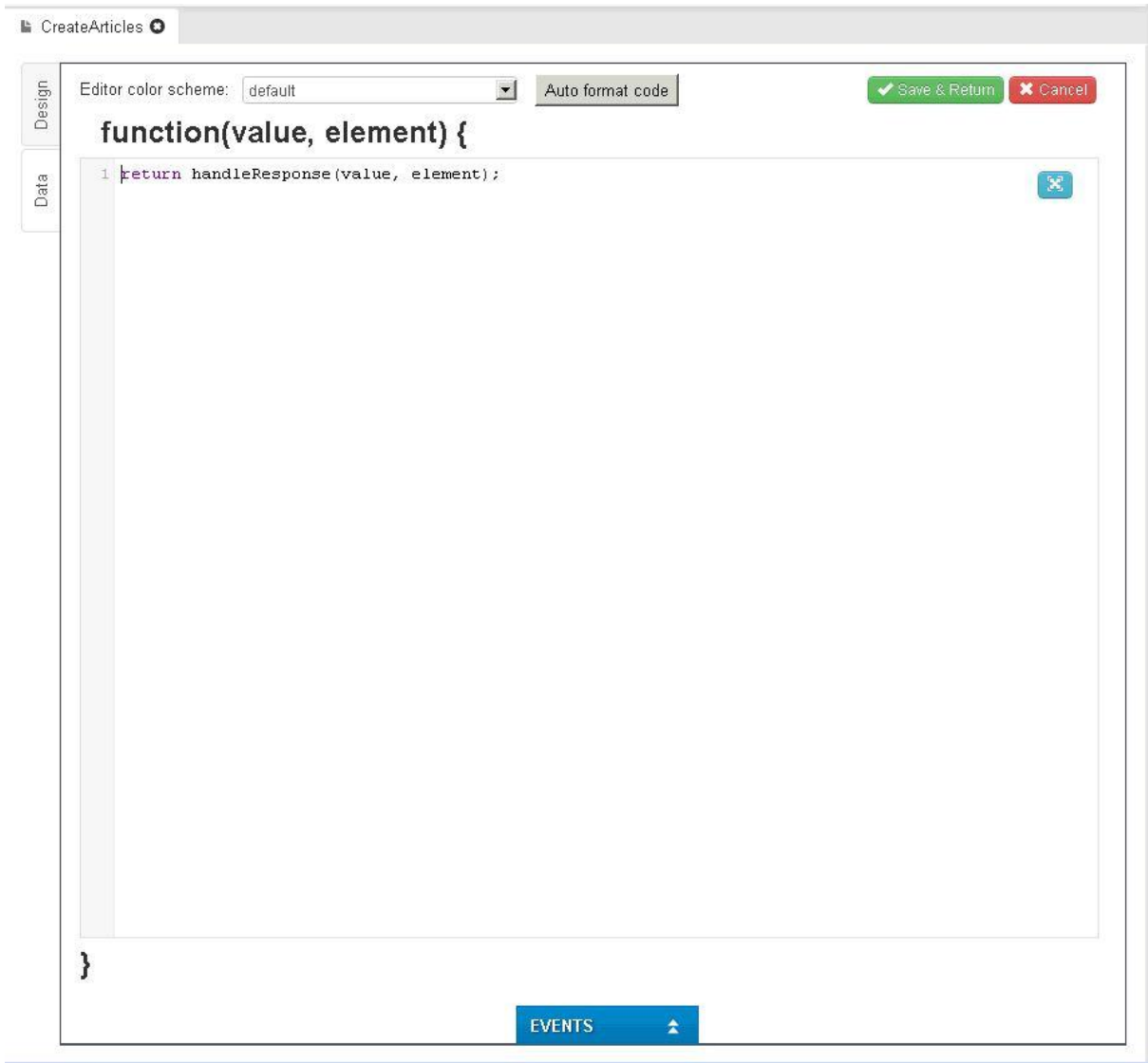
The server sends a response back which is mapped as follows on the “Response” tab from dsArticle/ttArticle/RequestStatus to StatusArea on the right side:



Once the mapping has been made, click the green “Add JS” button on the target StatusArea element (just to the right of the target) and enter:

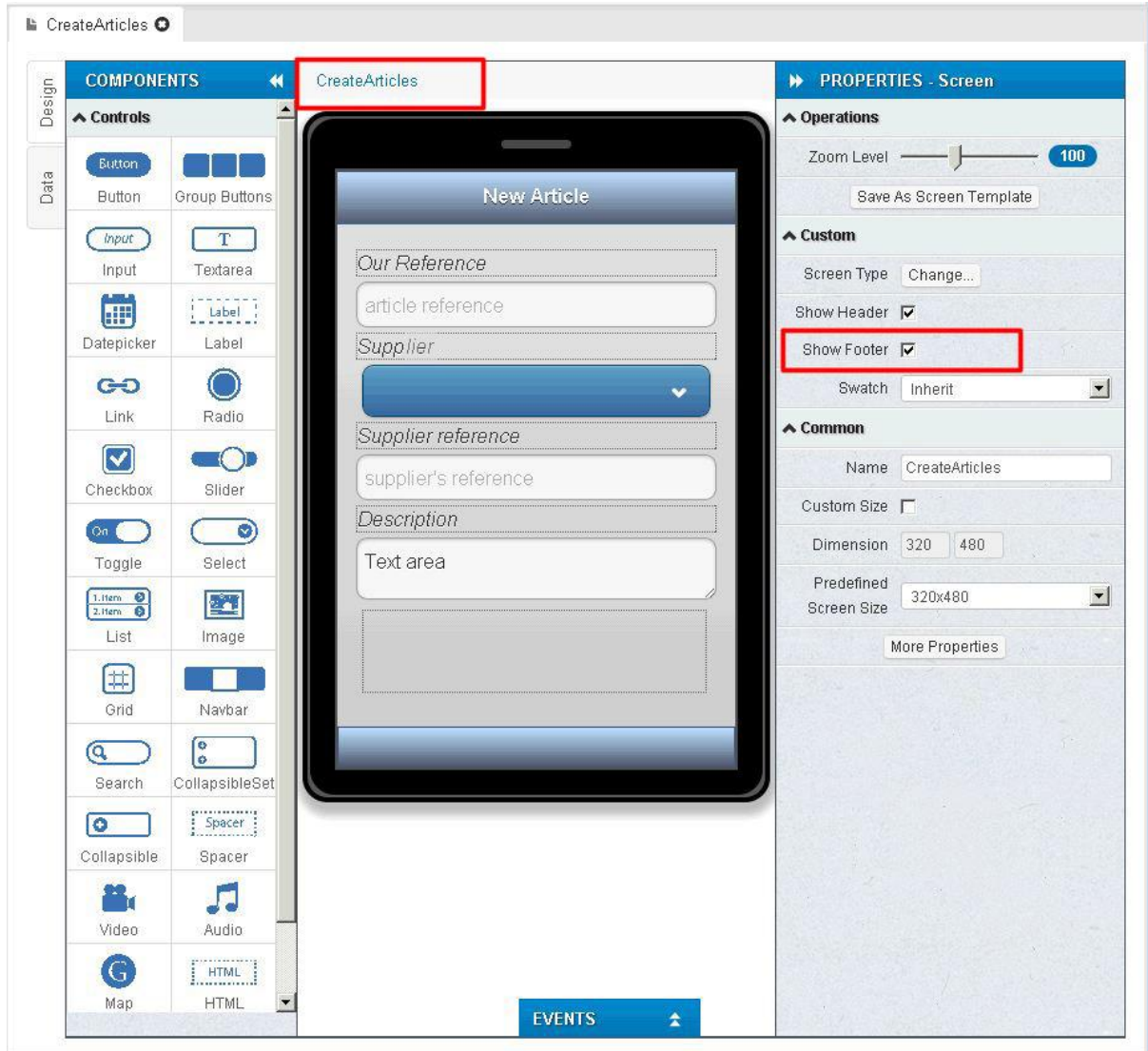
```
return handleResponse(value, element);
```

as shown below:

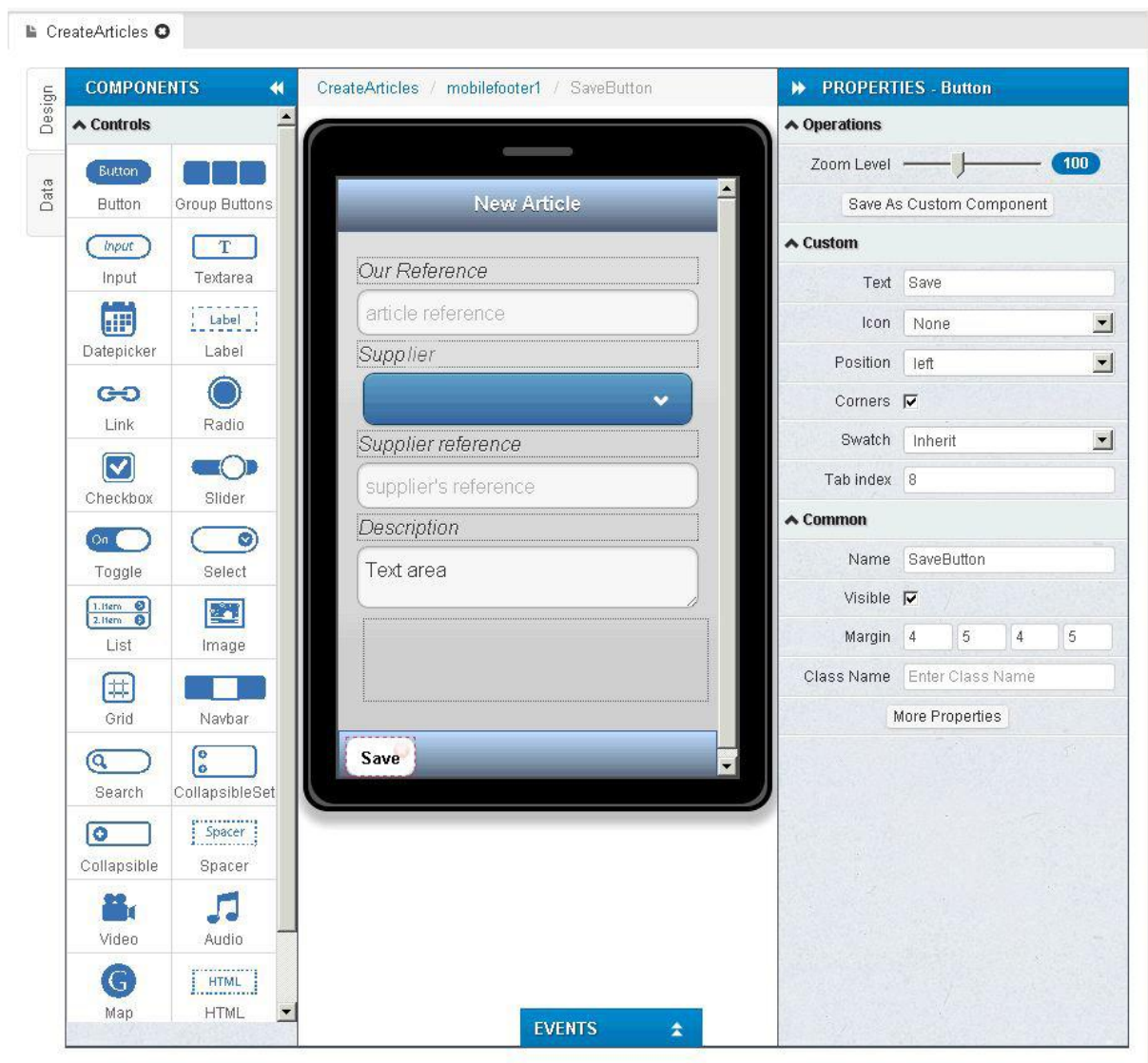


handleResponse() is a JavaScript function added to process the server response to a request to create a new Article (See Exchange, under JavaScript, on the left hand side of the page)

To be able to test the Create Article functionality we will add a 'Save' button into a footer at the bottom of the page. To add the footer itself, return to the design view and click on CreateArticles to show the required property sheet and then 'Show Footer'



Once the footer appears, drag a button into it and give it an appropriate label, as shown below:



To get the button to work it needs to be associated with an action. Expand the 'Events' slider at the bottom of the page as choose the events shown below: Add to "SaveButton" on the "Click" event the "Invoke Service" action on "ArticleCreate" datasource.

The screenshot displays a mobile application development tool interface. The central area shows a mobile device screen with a form titled "New Article". The form contains several input fields: "Our Reference" (with a sub-label "article reference"), "Supplier" (with a dropdown arrow), "Supplier reference" (with a sub-label "supplier's reference"), and "Description" (with a sub-label "Text area"). A "Save" button is located at the bottom left of the form. To the right of the form, an "EVENTS" slider is expanded, showing a configuration table.

Component	Event	Order	Action	Details
SaveButton	Click		Invoke service	Datasource: ArticleCreate

Below the table are buttons for "Add event" and "Cancel".

On the left side, there is a "COMPONENTS" panel with "Controls" and "Data" categories. The "Properties - Button" panel on the right shows settings for the selected button, including "Zoom Level" (100), "Text" (Save), "Icon" (None), "Position" (left), "Corners" (checked), "Swatch" (Inherit), "Tab index" (8), "Name" (SaveButton), "Visible" (checked), "Margin" (4, 5, 4, 5), and "Class Name" (Enter Class Name).

Do not forget to click Save.
Save the project and test.

Extra work

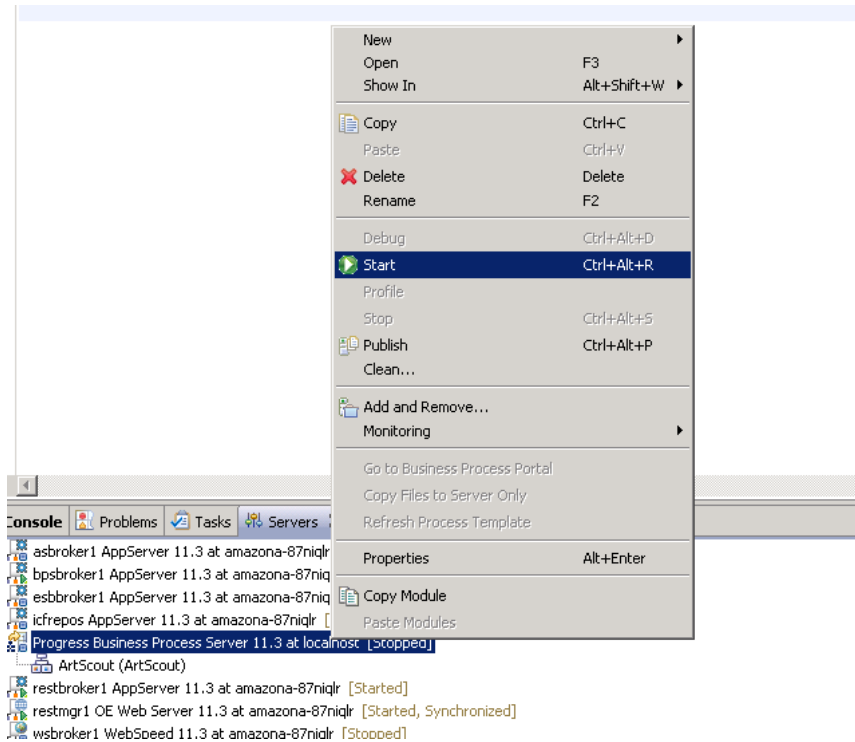
Add Article list overview page to check article status.

Chapter 5 – Implementing OE-BPM

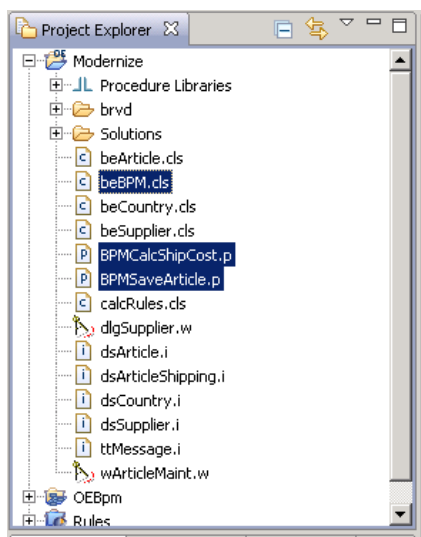
Build BPM process and Supplier Webscreen.

Preparation

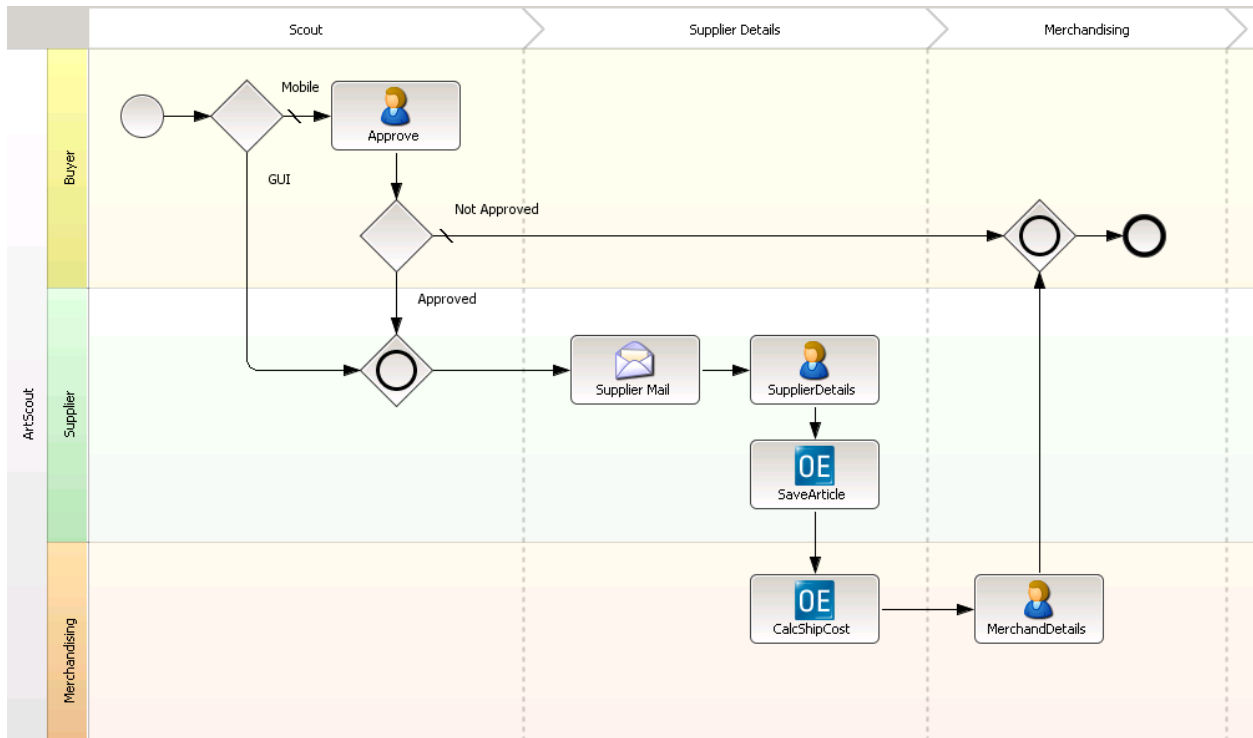
Go to the server view, right-click on the ‘Progress Business Process server 11.3’ and press Start. This will start the BPM engine. Startup takes about 2 minutes.



Copy the beBPM.cls and BPM*.p files from the “bpm-start” folder on your desktop to your ‘modernize project’



Process Explanation



- **Start**
At the start of the application the client type will be given to the process. This will determine whether the client is mobile or a GUI application.
- **Decision 1**
Here the BPM server will decide which way to go based on the client type. If it's a Mobile client the article first needs to be approved. Else there will be sent a mail to the supplier.
- **Approve**
Here someone from the backoffice has to decide if the article is good enough to be sold. If it is a good article the supplier will be sent a mail asking him to fill in the extra article details, otherwise the process will end.
- **Mail supplier**
The BPM server will send a mail to the supplier asking him to fill in the extra article details.
- **Add supplier details**
In this step the supplier will have a new task added to his tasklist. In this task he has to enter additional details for the article in the webform.
- **Save article**
The BPM server will call an Openedge procedure that will save the article.
- **Calculate shipment cost**
In this step the BPM server calls a Openedge procedure that will calculate the shipment cost.
- **Merchandise details**
In this step the merchandiser will have a task added to his tasklist. He will have to enter the last details to the article in the desktop GUI application. After he has filled in the details this step will be completed and the process will end.
- **End step**

After all the steps have been completed the process will end.

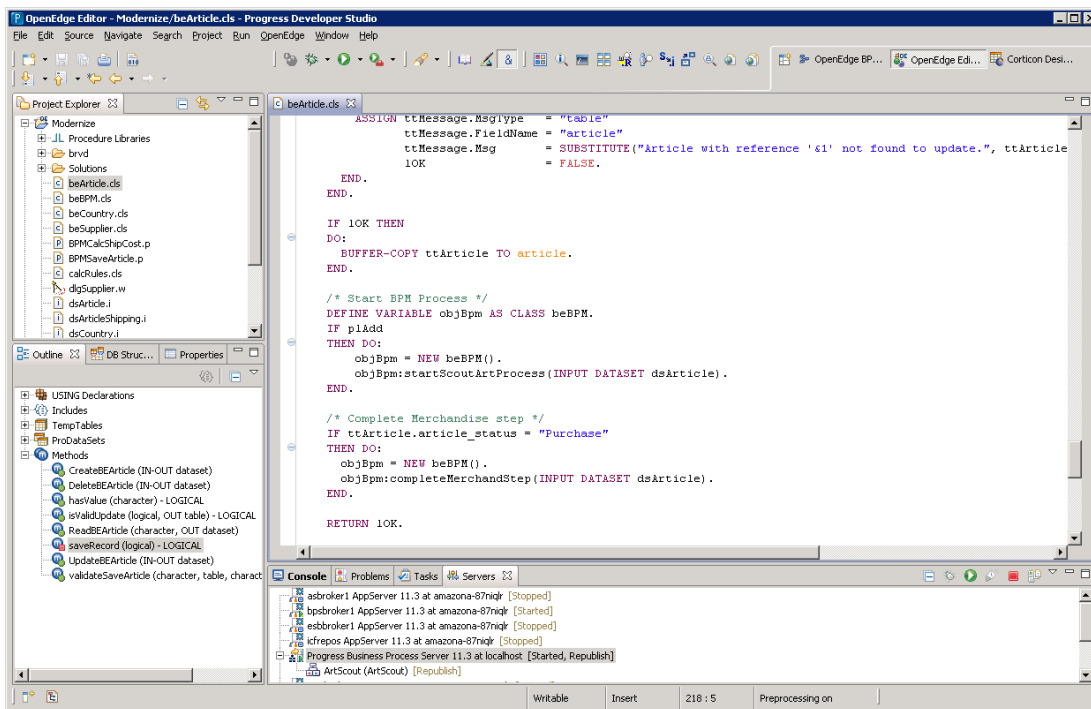
Code changes

Start the process on create of an article

Go to the 'OpenEdge Editor' Perspective.

Open the 'beArticle.cls' file by double-clicking it in the 'Project Explorer'.

Select the 'saveRecord' method in the 'Outline' view.



Add the Start BPM process code and also add the Complete merchandise step code at the end of the method.

```
IF IOK THEN
DO:
    BUFFER-COPY ttArticle TO article.
END.

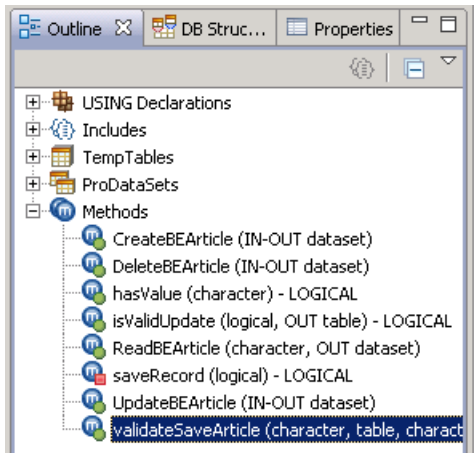
/* Start BPM Process */
DEFINE VARIABLE objBpm AS CLASS beBPM.
IF plAdd
THEN DO:
    objBpm = NEW beBPM().
    objBpm:startScoutArtProcess(INPUT DATASET dsArticle).
END.

/* Complete Merchandise step */
IF ttArticle.article_status = "Purchase"
THEN DO:
    objBpm = NEW beBPM().
    objBpm:completeMerchandStep(INPUT DATASET dsArticle).
END.

RETURN IOK.

END METHOD.
```

Now select the validateSaveArticle method in the 'Outline' view.



Add code to the validateSaveArticle to recognize from which UI we are coming. If it's from the GUI we assign this so that the approve step can be skipped in the process. An article created from the Mobile device has to be approved first.

```

METHOD PUBLIC VOID validateSaveArticle(INPUT cActionMode AS CHARACTER,
                                       INPUT TABLE ttArticle,
                                       INPUT cUser AS CHARACTER,
                                       OUTPUT cResult AS CHARACTER):

    DEFINE VARIABLE plAdd AS LOGICAL      NO-UNDO.
    DEFINE VARIABLE lOK  AS LOGICAL      NO-UNDO INIT FALSE.

    ASSIGN plAdd = IF cActionMode = "new" THEN TRUE ELSE FALSE.

    FIND FIRST ttArticle NO-ERROR.

    IF isValidUpdate(plAdd, OUTPUT TABLE ttMessage) THEN
    DO:
        CASE cUser: /* Set New Status */
            WHEN "Supplier"
                THEN ASSIGN ttArticle.article_status = "Administered".
            WHEN "Merchandising"
                THEN ASSIGN ttArticle.article_status = "Purchase".
        END CASE.

        /* Set status to know if article is created from GUI or mobile */
        IF cActionMode = "new"
            THEN ASSIGN ttArticle.article_status = "GUI".

        lOK = saveRecord(plAdd).
    END.

    FOR EACH ttMessage:
        ASSIGN cResult = cResult + ttMessage.Msg + CHR(13).
    END.

END METHOD.

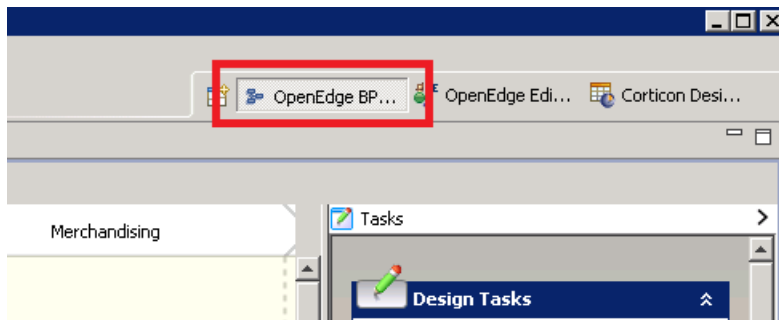
```

Do the same for the BEArticle.cls in your mobile project!!!

!!! We need todo this twice only now for test purposes. In a real environment this file only exists once. !!!

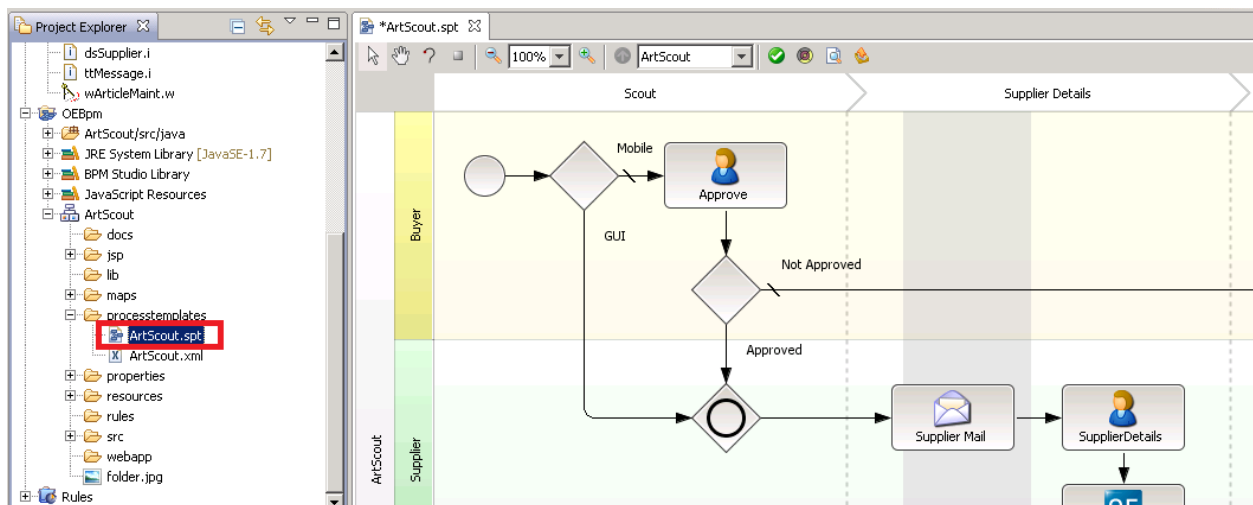
Process Changes

Switch to “OpenEdge BPM” perspective.

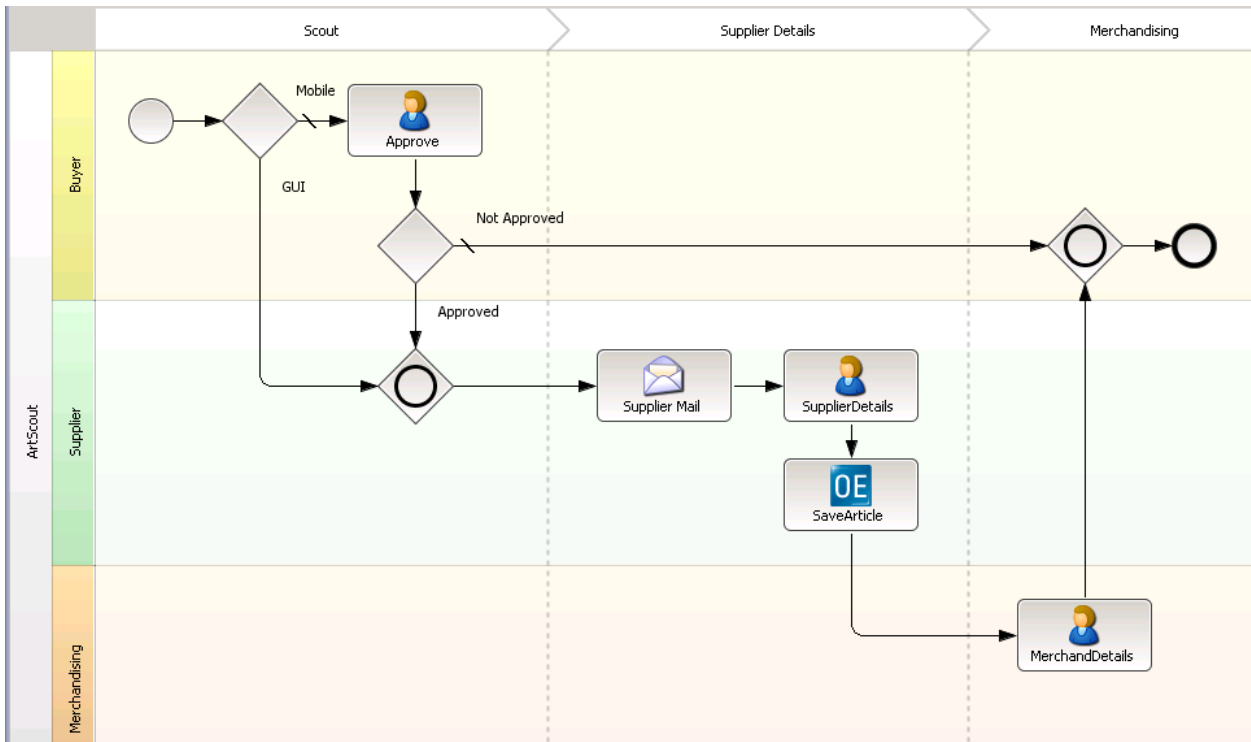


Open the “ArtScout.spt” file. You can find this in the OEBpm project -> ArtScout -> processtemplates.

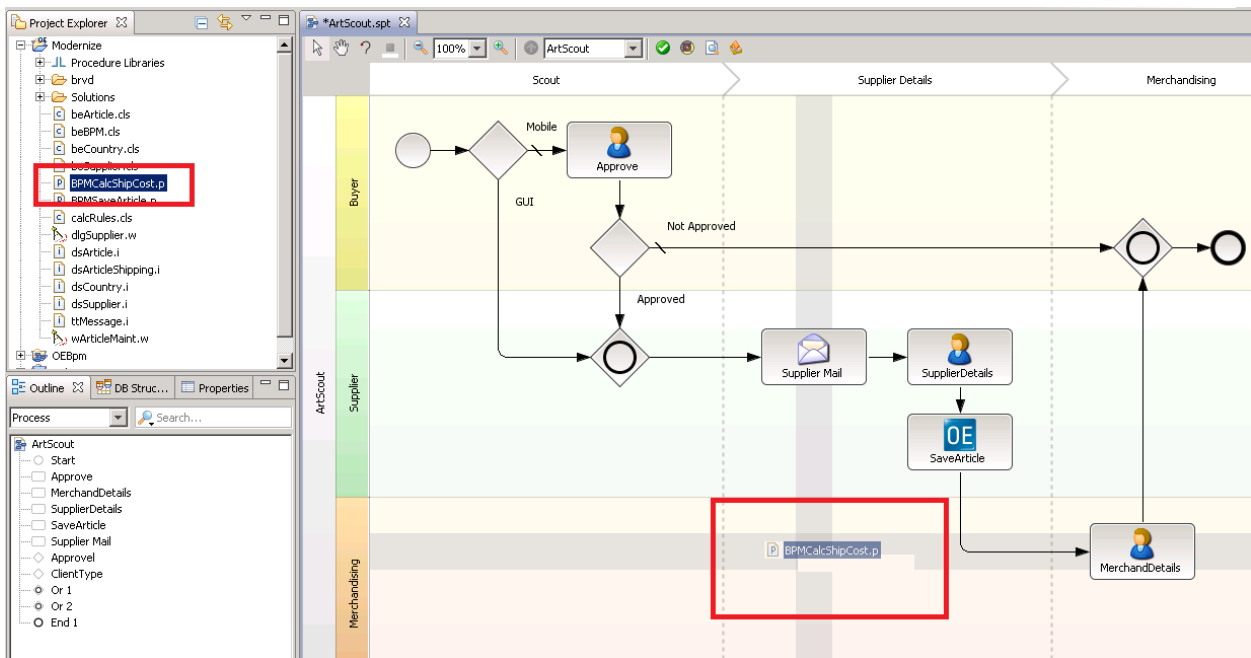
This will open de BPM process designer.



In the process you start with is the 'Calculate Shipping Cost' task missing. This needs to be added.



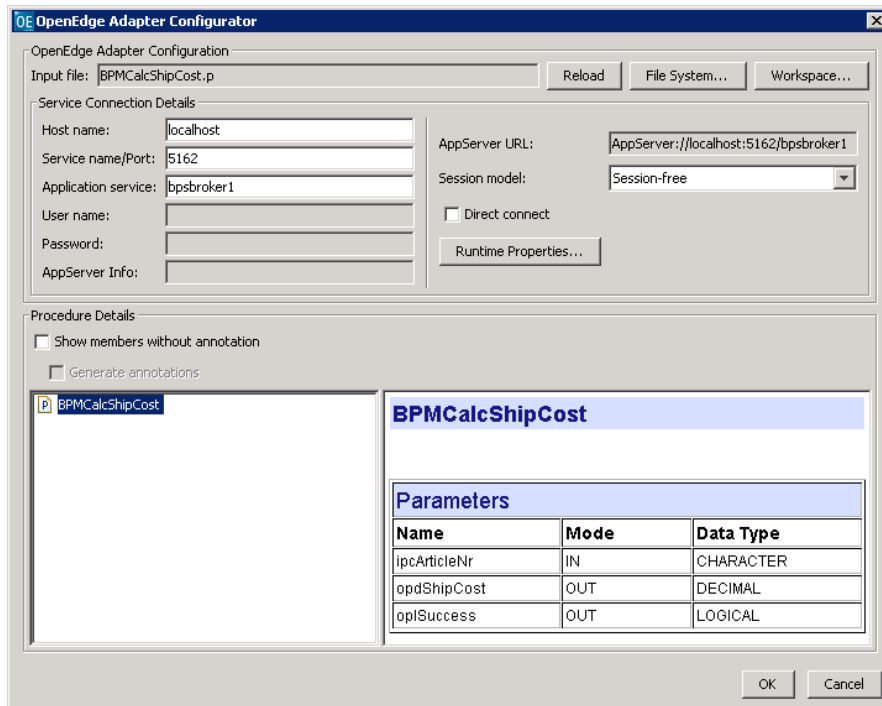
Drag and drop the 'BPMCalcShipCost.p' on the BPM canvas.



You now will get a pop-up which shows the configuration parameters to call the OpenEdge Appserver.

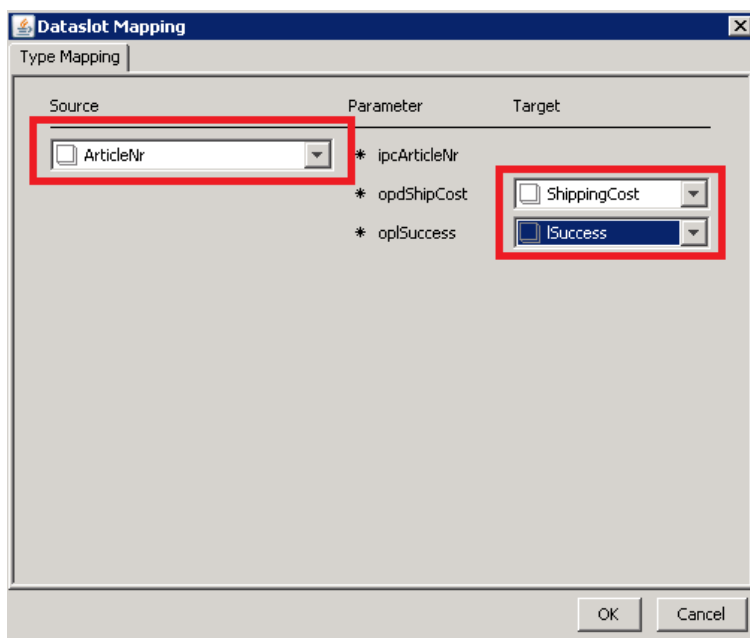
Leave all values as default.

We also see here the in and output parameters from the procedure file.

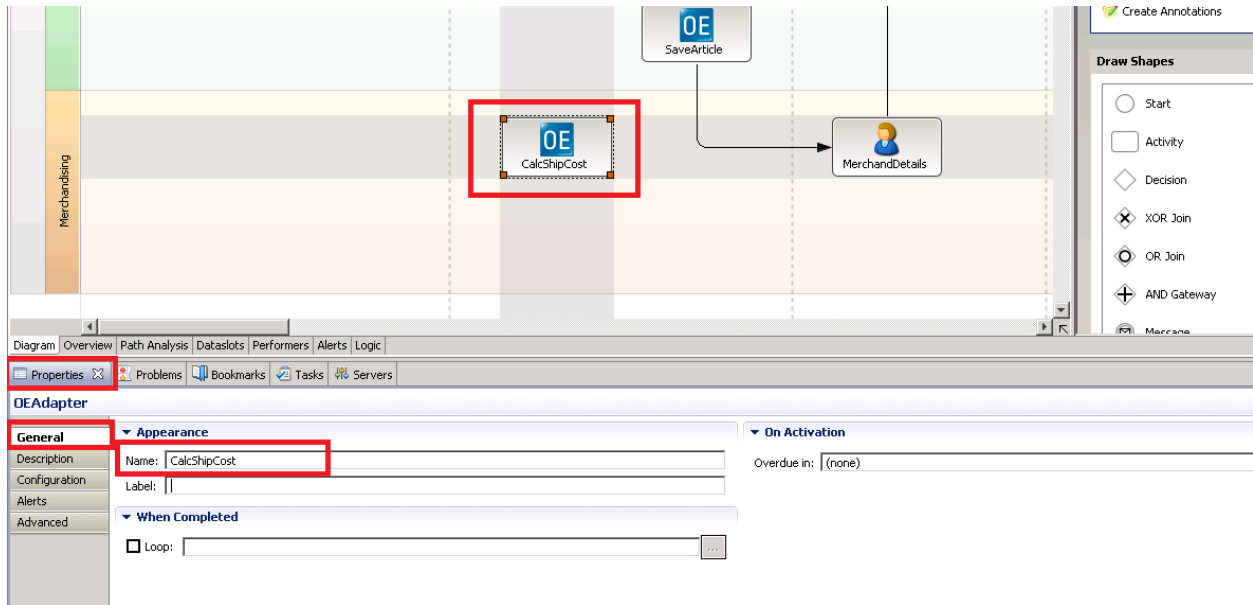


Click Ok.

On the next screen we need to make the mapping from our BPM variables to our procedure in and outputs. Link them as provided in the screenshot.

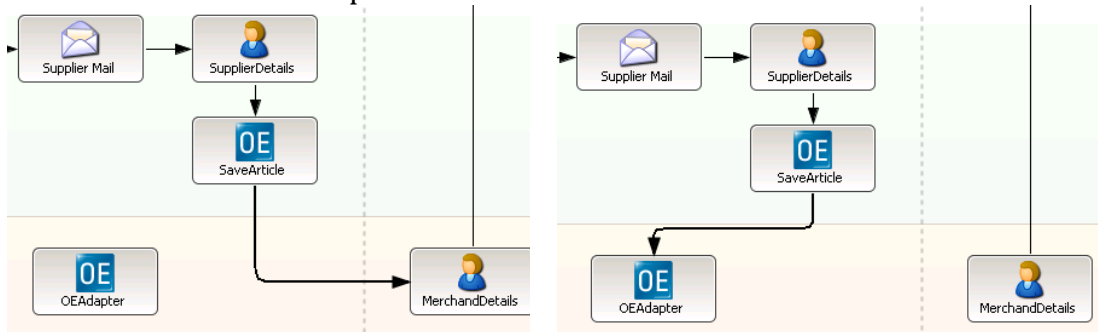


Change the name of the task.
 Select the OEAdapter
 Select properties -> General
 And change Name to "CalcShipCost"

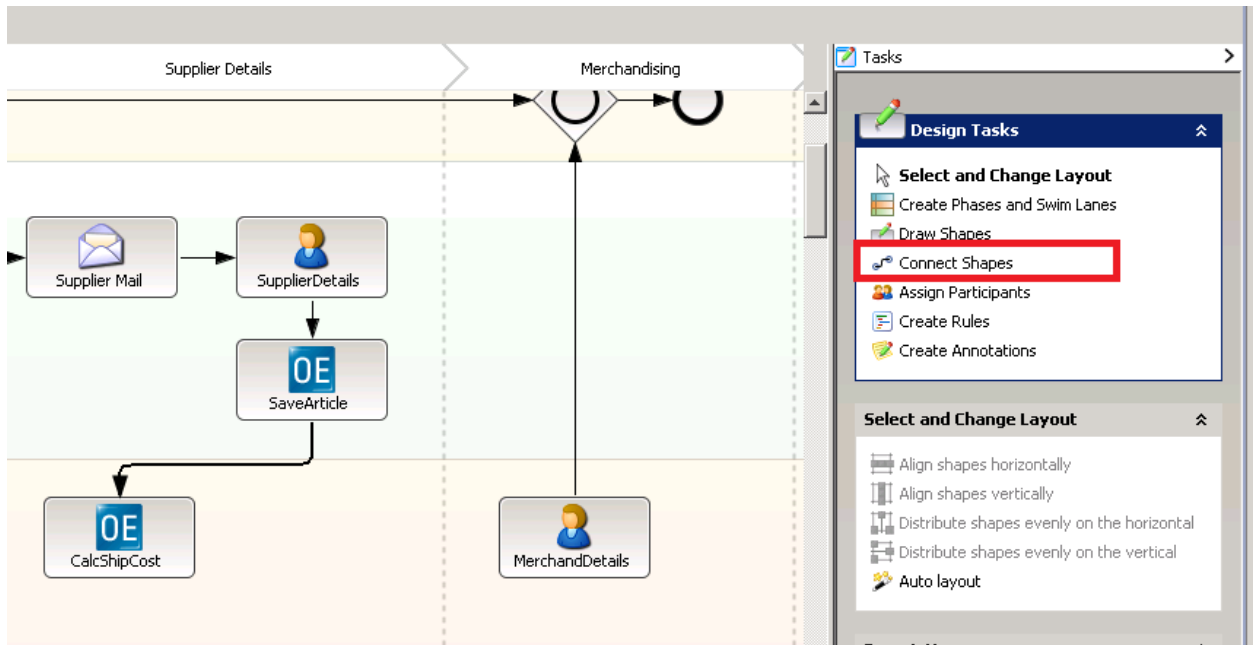


Reroute the connection line which is original from 'SaveArticle' to 'MerchandDetails'

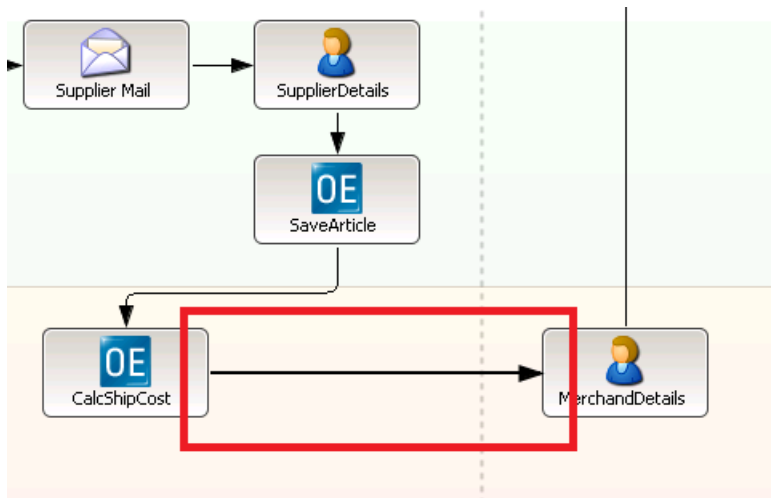
To 'SaveArticle' to OEAdapter



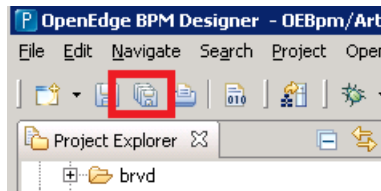
We need an extra connection from “CalcShipCost” to “Merchanddetails”.
Select the “Connect Shapes” in the Design Tasks Pane.



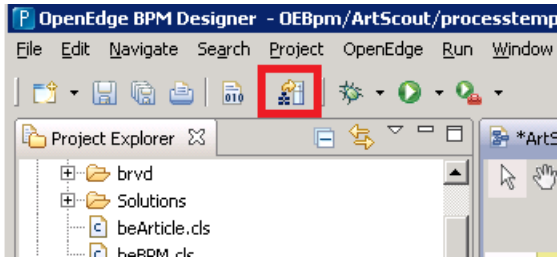
Draw a line from “CalcShipCost” to “Merchanddetails”



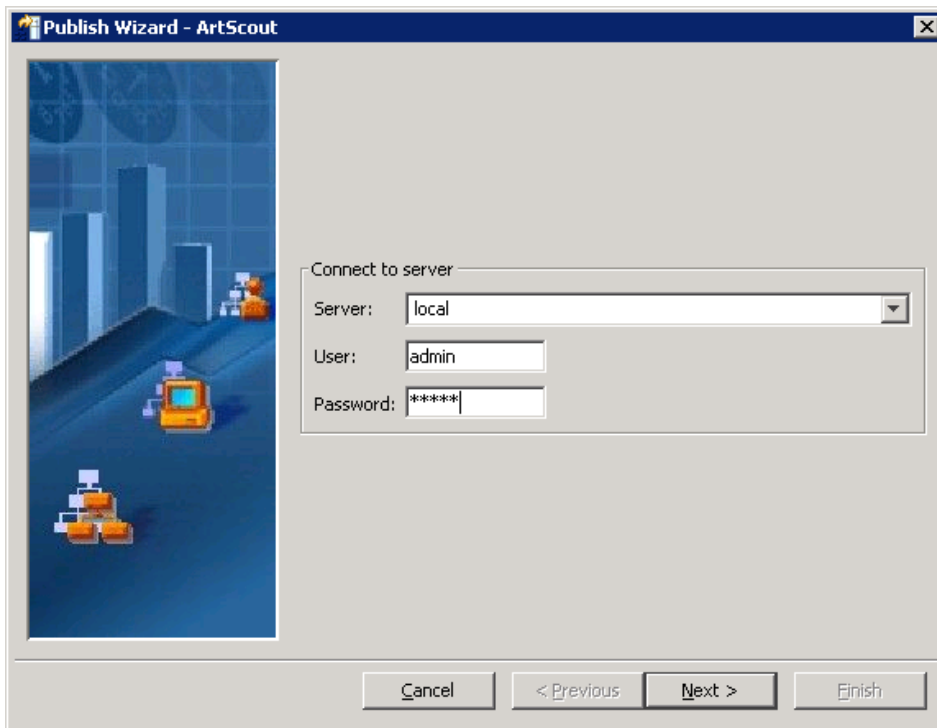
Save the BPM project



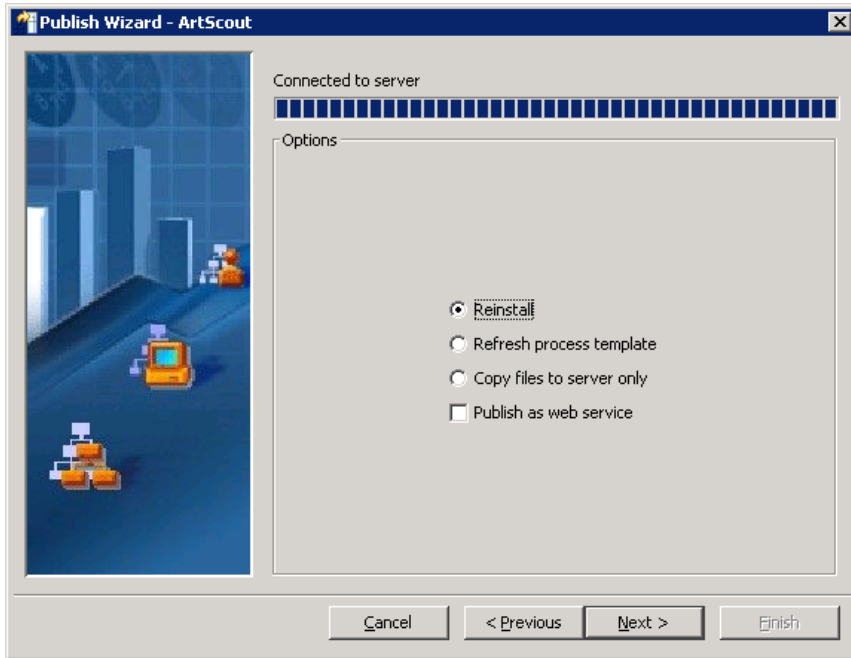
Now we need to publish our new flow to the BPM engine.
Click the “Publish” button.



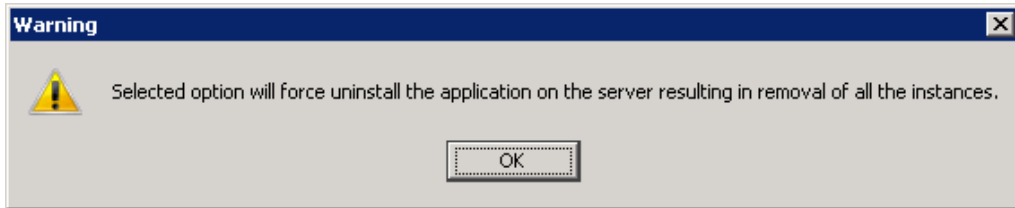
Server will be "local"
User "Admin"
Password "Admin"
This to connect to the BPM server.



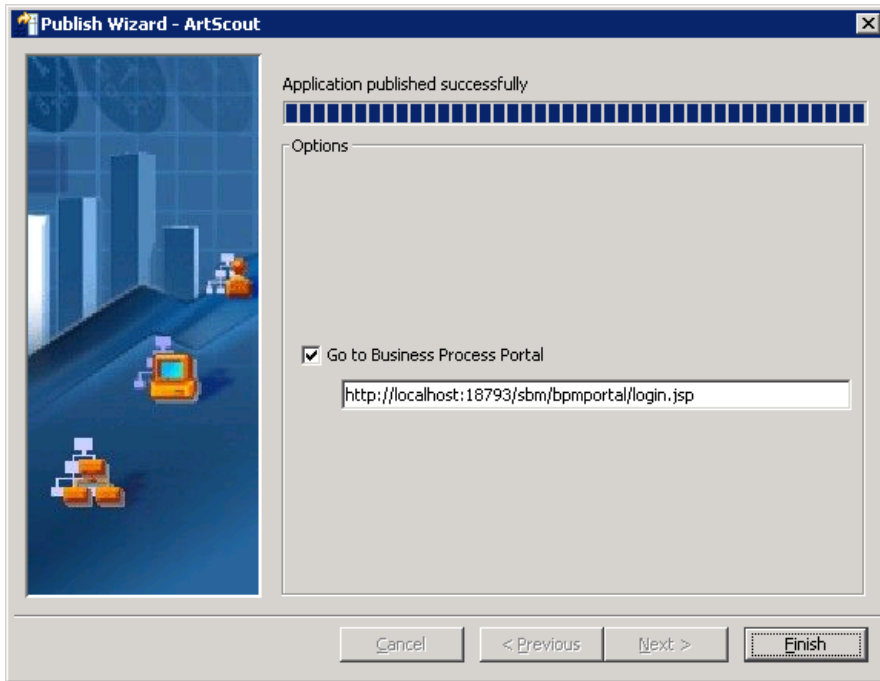
Choose “Reinstall”, this will remove the previous process.



Accept the warning with Ok.



And press “Next”
The process is now uploading to the BPM server.



On Finish the BPM webportal login page will be started. This we only need in a later phase.

Test the process.

Create Article

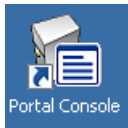
First create a new article from the GUI application.

If you still have a mobile test screen open close this first and click “test” again in the mobile appbuilder.

Second create a new article from the Mobile application.

Now we will check if a process is started for both of them.

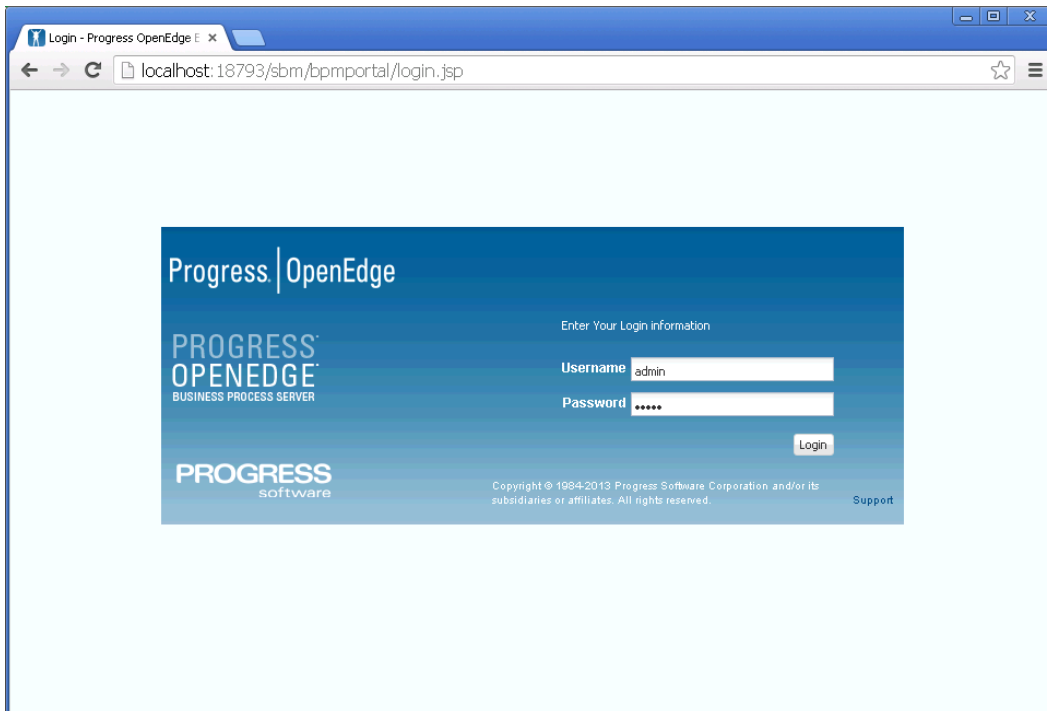
Go to the BPM portal by clicking the Portal Console Icon on your desktop.



Login with

Username : Admin

Password : Admin

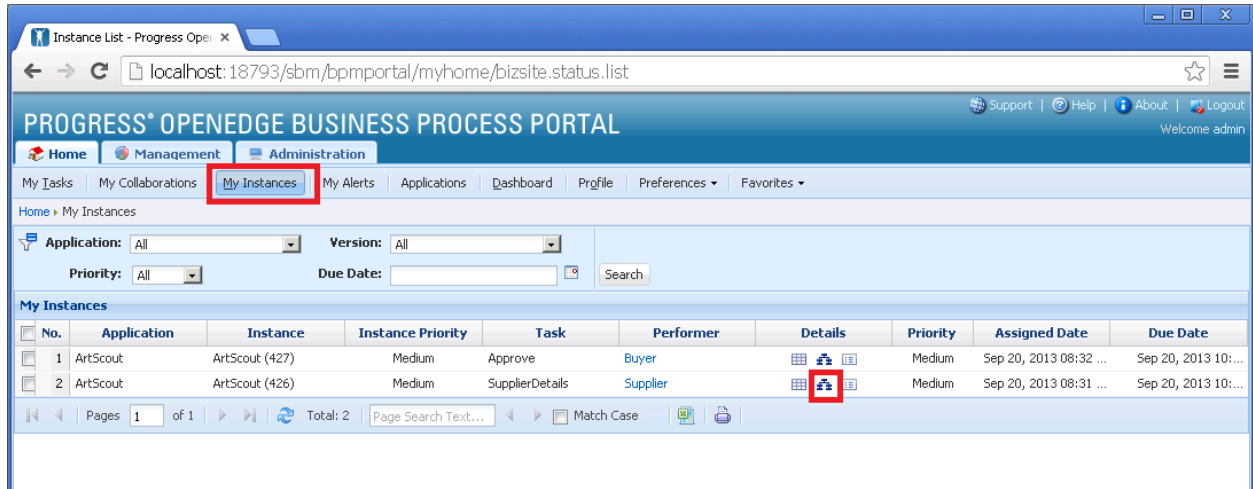


Select the 'My Instances' tab to investigate what happened.

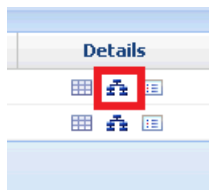
You now should see 2 processes. These are the 2 processes which are started.

One process is on the 'SupplierDetails' task, this is the task started from the GUI application and is waiting for the supplier to fill in extra article details.

The other process is on the 'Approve' task, this task is started from the Mobile Application and is waiting for someone to approve this article before the supplier is asked to add extra details.



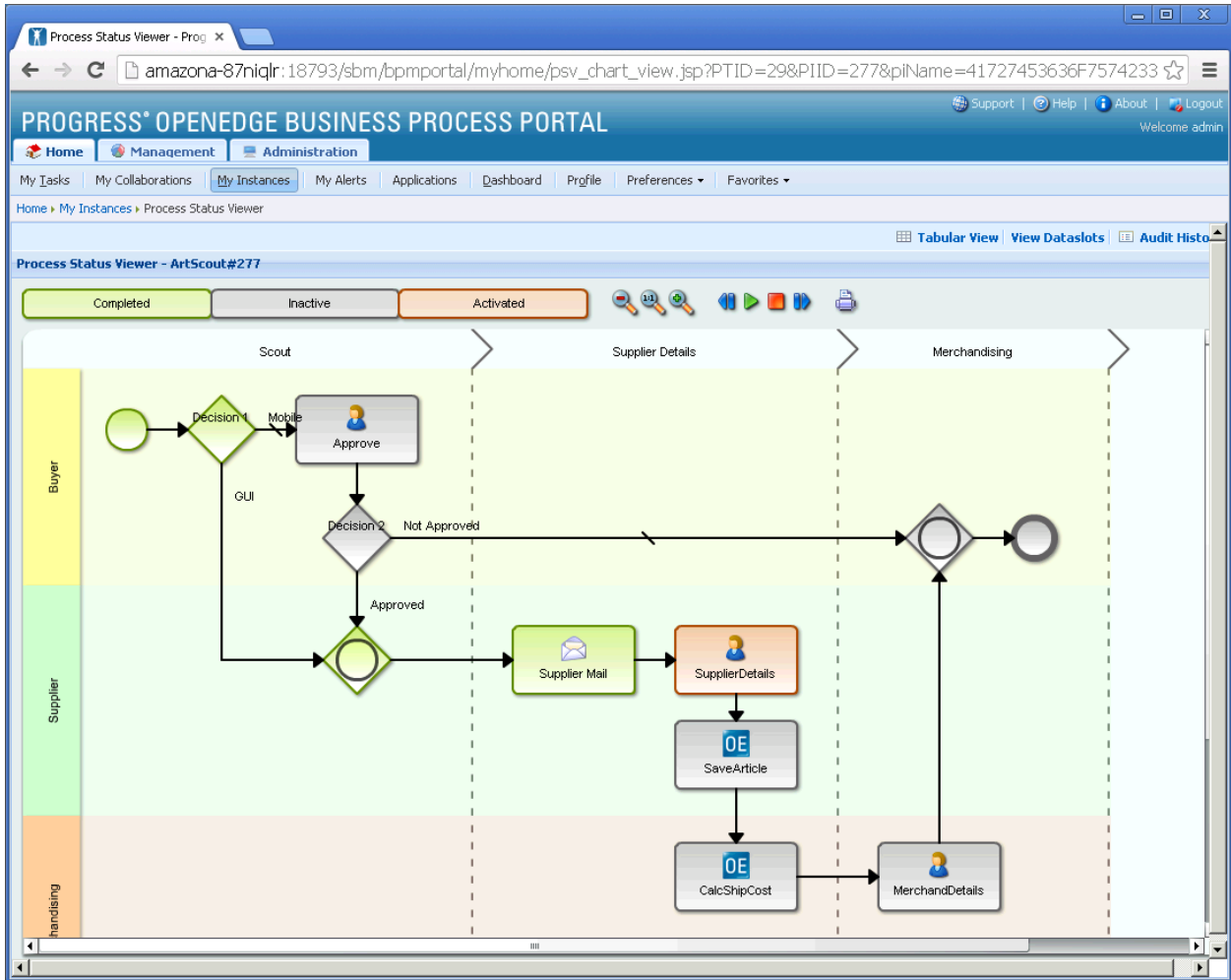
By selecting the flow icon for one instance we see the visual flow of the process and the process path which was followed. We can also see on which task the process is waiting now.



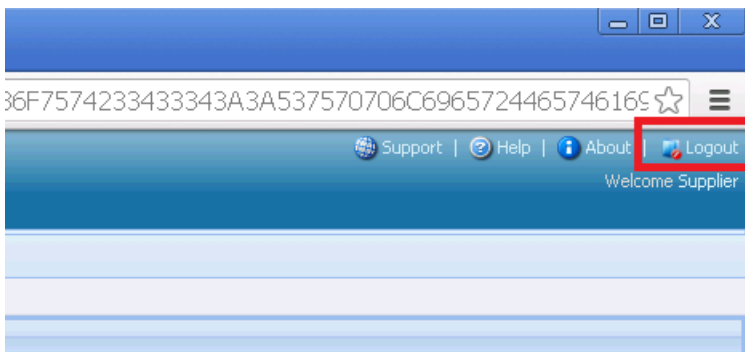
This process for example skipped the 'Approve' task which means was started from the GUI application.

Did send an email to the Supplier for a requested to fill-in more article details.

And is now waiting for the supplier to complete his task to enter more article details.



Logout of the BPM portal.



The Approval Task

Login to the BPM Portal with
User : Buyer
Password : Buyer

You should see now only 1 task.

This is the task which needs to be completed by the Approver / Buyer Administration in the backoffice.

The screenshot displays the 'My Tasks' interface. At the top, there are navigation tabs: Home, My Tasks, My Collaborations, My Instances, My Alerts, Applications, Dashboard, Profile, Preferences, and Favorites. Below these is a search section with filters for Application (All), Start Date (Range Field), Due Date (Range Field), Priority (All), Status (Assigned), Performer (Select One), and Presentation (Portal). A search button and 'Next Available Task' button are present. The main table lists one task:

No.	Application	Instance	Task	Creator	Priority	Assigned Date
1	ArtScout	ArtScout (435)	Approve#619	admin	Medium	Sep 20, 2013 09:41 AM

At the bottom of the page, there are buttons for 'Reassign', 'Make Available', and 'Complete'.

Click the 'Approve#***' Task link.

You now get a webform where the Approver can approve or disapprove the Article. Check 'Approved' and click 'Complete'. This will complete his task.

Home

My Tasks | My Collaborations | My Instances | My Alerts | Applications | Dashboard | Profile | Preferences | Favorites

Home > My Tasks > Task Details

ArtScout#435::Approve::Buyer

Instruction	None		
Priority	Medium	Start Date	Sep 20, 2013 09:41 AM
Article Nr	37000		
Supplier Name	Moonshine Inc		
Description	Homebrew 85%		
Approved	<input checked="" type="checkbox"/>		

Complete Save Reset Cancel Reassign

Task "ArtScout#435::Approve::Buyer" has been successfully completed.

OK

You get a status message that the task is completed.
Click Ok.

You now see the tasklist again and this is empty, means no more tasks todo for the approver.
His job is Finished.

Tasks

No.	Application	Instance	Task	Creator	Priority	Assigned Date
The search did not give any valid result. Please try again.						

Pages 0 of 0 Total: 0 Page Search Text... Match Case

Logout of the BPM portal.

The Supplier task

Login to the BPM Portal with
User : Supplier
Password : Supplier

The Supplier has 2 tasks to complete. One task was here directly because it was initiated from the GUI, the other first went to the Approver and is only now available for the Supplier.

My Tasks | My Collaborations | My Instances | My Alerts | Applications | Dashboard | Profile | Preferences | Favorites

Home » My Tasks

Search Tasks

Application: All | Start Date: Range Field | Select One | Status: Assigned
Version: All | Due Date: Range Field | Select One | Performer: User
Workstep: Select One | Priority: All | Presentation: Portal

Advanced

No Filter | Search | Next Available Task | Reset

No.	Application	Instance	Task	Creator	Priority	Date
1	ArtScout	ArtScout (434)	SupplierDetails#618	admin	Medium	Sep 20, 20
2	ArtScout	ArtScout (435)	SupplierDetails#620	admin	Medium	Sep 20, 20

Pages 1 of 1 | Total: 2 | Page Search Text... | Match Case | Reassign | Make Available | Complete

Click on one of the 'SupplierDetails#***' Tasks.

You now will get the webform which the Supplier will see to fill in his extra article details. Add some numbers. For country, add some valid country code "BE, FR, UK, CA, US". In a later faze we will use a dropdown here and change the web page layout. Click Complete to finish the task.

Home

My Tasks My Collaborations My Instances My Alerts Applications Dashboard Profile Preferences Favorites

Home > My Tasks > Task Details

ArtScout#434::SupplierDetails::Supplier

Instruction: None

Priority: Medium Start Date: Sep 20, 2013 09:40 AM

Article Nr: 45000

Supplier Reference: SPIR003

Description: Fine Liquor

Article Height: 0.50

Article Width: 0.60

Article Length: 1

Country Of Origin: US

Weight Gross: 1.25

Weight Net: 1.15

Url: liquor.com

Complete Save Reset Cancel Reassign

There is now only 1 task left. Do the same for this task.

Logout of the BPM portal. And login again with 'Admin' 'Admin'. Go to 'My Instances' and click on the 'Flow' icon.

Home Management Administration

My Tasks My Collaborations My Instances My Alerts Applications Dashboard Profile Preferences Favorites

Home > My Instances

Application: All Version: All

Priority: All Due Date: Search

My Instances

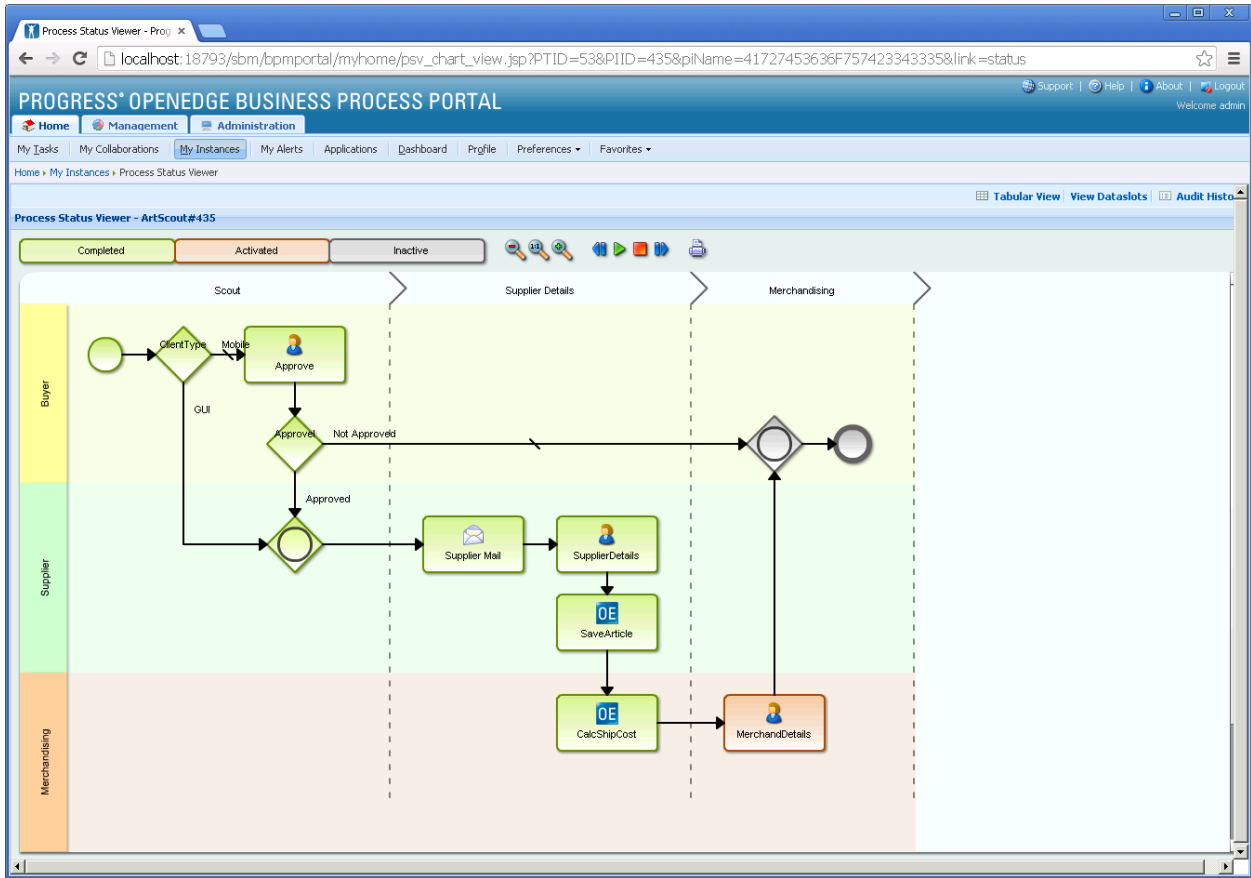
No.	Application	Instance	Instance Priority	Task	Performer	Details	Priority
1	ArtScout	ArtScout (435)	Medium	MerchandDetails	Merchandising		Medium
2	ArtScout	ArtScout (434)	Medium	MerchandDetails	Merchandising		Medium

Pages 1 of 1 Total: 2 Page Search Text... Match Case

We see now that the 'SupplierDetails' task is completed and that the 'SaveArticle' task is also executed.

Also the 'CalcShipCost' is executed. These are background services which were activated from the BPM.

Now the process is waiting for the merchandising to complete their task.



You can check in the GUI application to see if the Supplier data is updated in the database. If your GUI application is still open just press the 'Filter' button to get the latest data from the database.



The Merchandise Task

The merchandising task should be completed via the GUI application. Login in the GUI application with the 'Merchandising' user.

Look for the article which you created before, and update the order amounts.

The screenshot shows a window titled "Buyer and Merchandise Administration". At the top, there are input fields for "Article Number:" and "Status:" with a "Filter" button. Below this is a table with the following data:

Article Number	Article Description	Supplier Name	Status
3	White stuff above 80°	Moonshine Inc	Administere
4	Alco pop Red	Spirits Co.	Administere
4000	eee	Moonshine Inc	GUI
41	bb	Moonshine Inc	Purchase
42	ddcdcdcd cdcd cdcd aaa	Spirits Co.	Administere
43	ggggg	Wines International	Administere
44000	Fine Liquer	Spirits Co.	Administere
45000	Fine Liquer	Spirits Co.	Administere
46000	Brandy High Quality	Spirits Co.	GUI
50	bb	Moonshine Inc	GUI
51	qq	Spirits Co.	GUI
7	dddd	Wines International	New

Below the table are tabs for "Article", "Supplier", and "Merchandising". The "Merchandising" tab is active, showing a form for article 45000. The form includes:

- Article Number: 45000
- Supplier Code: SPIR001 (with a dropdown arrow and "Wines International" selected)
- Default order Amount: 12.00
- Minimum Order Amount: 6.00
- LeadTime: 0.00

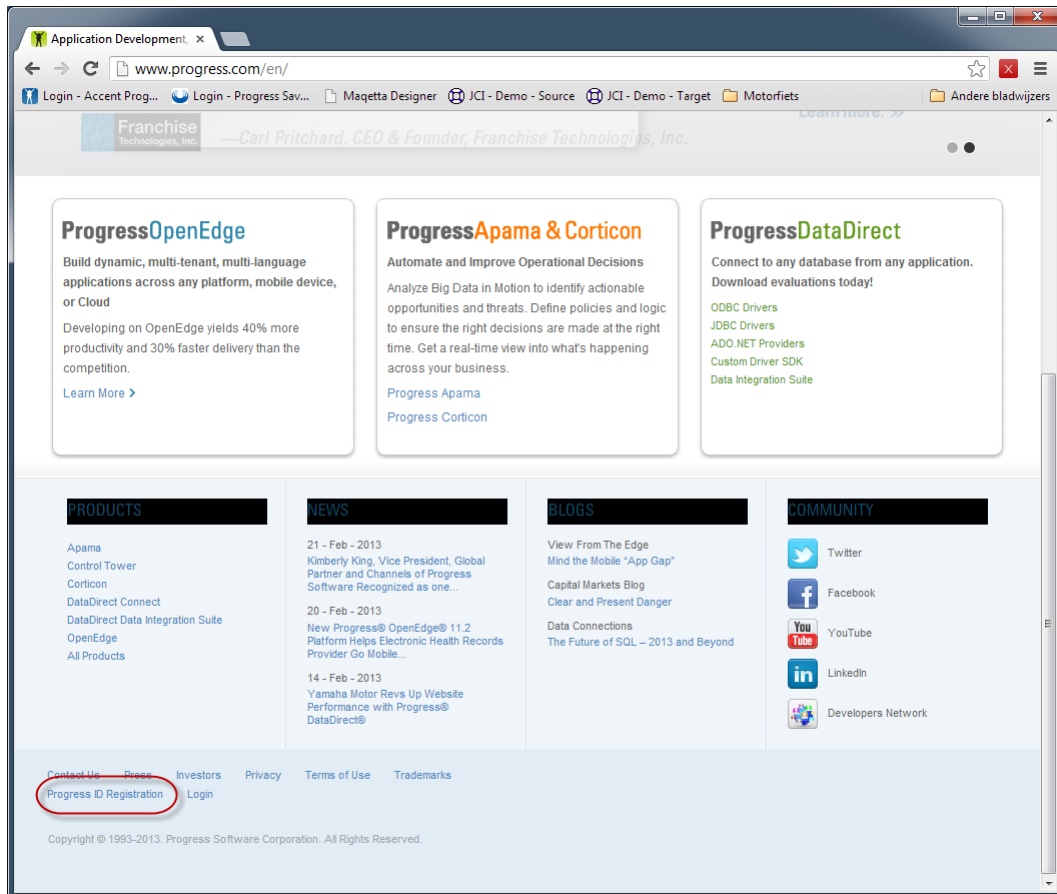
At the bottom of the form, there are two yellow boxes for "Predict Shipping Cost" and "Carticon Shipping Cos", both showing 0.00. There are also "Update", "Save", and "Cancel" buttons. The "Save" button is highlighted with a red box.

Now check again the instances in the BPM portal and you will see that one process is disappeared, means process completed. And the entire article process is ready now.

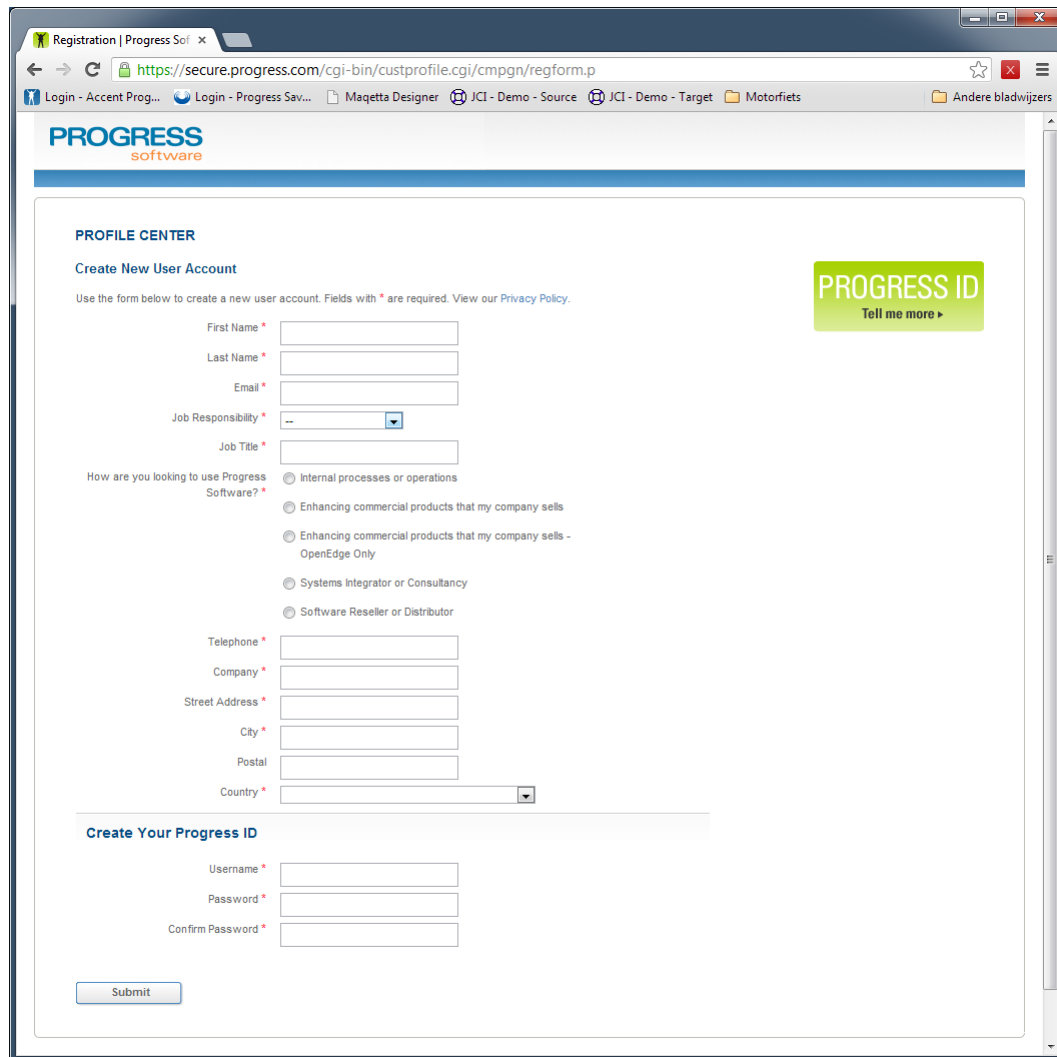
Appendix A – Getting a Progress login

If you don't have a Progress login yet, follow these steps to get one.

1. Go to <http://www.progress.com>, scroll to the bottom of the page click **Progress ID Registration**.



2. Complete the form and click **Submit**



The screenshot shows a web browser window with the URL <https://secure.progress.com/cgi-bin/custprofile.cgi/cmpgn/regform.p>. The page is titled "PROGRESS software" and "PROFILE CENTER". The main heading is "Create New User Account". Below this, there is a sub-heading "Create Your Progress ID" and a "Submit" button.

Use the form below to create a new user account. Fields with * are required. View our [Privacy Policy](#).

PROGRESS ID
Tell me more >

First Name *

Last Name *

Email *

Job Responsibility * --

Job Title *

How are you looking to use Progress Software? *

- Internal processes or operations
- Enhancing commercial products that my company sells
- Enhancing commercial products that my company sells - OpenEdge Only
- Systems Integrator or Consultancy
- Software Reseller or Distributor

Telephone *

Company *

Street Address *

City *

Postal

Country *

Create Your Progress ID

Username *

Password *

Confirm Password *

You will receive an email with your Progress Login Credentials.
Your Mobile App Builder evaluation period of 60-days will start automatically when you log on to the Mobile App Builder for the first time.

Appendix B – Gui Application Manual

The screenshot shows the 'Buyer and Merchandise Administration' application window. At the top, there are search filters for 'Article Number' and 'Status', along with a 'Filter' button. Below this is a table listing articles with columns for Article Number, Article Description, Supplier Name, and Status. Article 6, 'Raspberry Pi single-board computer', is highlighted in blue. Below the table are three tabs: 'Article', 'Supplier', and 'Merchandising'. The 'Article' tab is active, showing a detailed view for article 6. It includes fields for Article Number (6), Supplier Code (RASP001), Supplier Reference (123123), and Description (Raspberry Pi single-board computer). The supplier name 'The Raspberry Pi Foundation' is displayed next to the code. At the bottom, there is a 'Predict Shipping Cost' button showing '\$ 30', and 'New', 'Update', 'Save', and 'Cancel' buttons.

Article Number	Article Description	Supplier Name	Status
1	Smartphone Ascend P6	Huawei Technologies Co.	Purchase
2	Smartphone Ascend P2	Huawei Technologies Co.	Purchase
3	Quartz Kitchen	DuPont	Purchase
4	Video Conferencing KX-VC600EX kit	Sanyo Electric	Purchase
5	Documentcamera - Avervision CP130	Sanyo Electric	Purchase
6	Raspberry Pi single-board computer	The Raspberry Pi Foundation	New
7	NIKE AIR PEGASUS+ 30	Nike Inc.	New
8	The Da Vinci code	Doubleday	New

It is possible to filter the articles by article number and by status.

On the article tab the buyer can add new articles.

It is required to use a unique Article number for the new article. The supplier code has to be chosen from a list of suppliers. The suppliers reference is the code that is used by the supplier.

The description is a mandatory field

Buyer and Merchandise Administration

Article Number: Status:

Article Number	Article Description	Supplier Name	Status
1	Smartphone Ascend P6	Huawei Technologies Co.	Purchase
2	Smartphone Ascend P2	Huawei Technologies Co.	Purchase
3	Quartz Kitchen	DuPont	Purchase
4	Video Conferencing KX-VC600EX kit	Sanyo Electric	Purchase
5	Documentcamera - Avervision CP130	Sanyo Electric	Purchase
6	Raspberry Pi single-board computer	The Raspberry Pi Foundation	New
7	NIKE AIR PEGASUS+ 30	Nike Inc.	New
8	The Da Vinci code	Doubleday	New

Article Supplier Merchandising

Supplier

Article Number:

Supplier Code: Huawei Technologies Co.

Country of Origin:

Net Weight: Gross Weight:

Length: Width: Height:

Website URL:

On this screen the supplier can validate the new articles. The suppliers has to choose the country where the article comes from. He also has to fill in the net weight and gross weight. The weight is measured in pounds. The dimensions of the article are measured in inches. The website URL is an optional field to enter the website of the supplier.

Buyer and Merchandise Administration

Article Number: Status:

Article Number	Article Description	Supplier Name	Status
1	Smartphone Ascend P6	Huawei Technologies Co.	Purchase
2	Smartphone Ascend P2	Huawei Technologies Co.	Purchase
3	Quartz Kitchen	DuPont	Purchase
4	Video Conferencing KX-VC600EX kit	Sanyo Electric	Purchase
5	Documentcamera - Aversion CP130	Sanyo Electric	Purchase
6	Raspberry Pi single-board computer	The Raspberry Pi Foundation	New
7	NIKE AIR PEGASUS+ 30	Nike Inc.	New
8	The Da Vinci code	Doubleday	New

Article Supplier Merchandising

Merchandising

Article Number:

Supplier Code: Huawei Technologies Co.

Default order Amount: Minimum Order Amount:

LeadTime:

Predict Shipping Cost

The Merchandiser can enter the default order and the minimum amount that has to be ordered. He can also enter the Lead time which is measured in days. The predict shipping costs calculates how much it will approximately cost to ship the product.



Progress Software

Progress Software Corporation (NASDAQ: PRGS) is a global software company that simplifies the development, deployment and management of business applications on premise or on any Cloud, on any platform and on any device with minimal IT complexity and low total cost of ownership.

Worldwide Headquarters

Progress Software Corporation, 14 Oak Park, Bedford, MA 01730 USA

Tel: +1 781 280-4000 Fax: +1 781 280-4095 On the Web at: www.progress.com

For regional international office locations and contact information, please refer to the Web page below: www.progress.com/worldwide

Progress and [ALL PRODUCT NAMES LISTED IN PUBLICATION] are trademarks or registered trademarks of Progress Software Corporation or one of its affiliates or subsidiaries in the U.S. and other countries. Any other marks contained herein may be trademarks of their respective owners. Specifications subject to change without notice.

© 2013 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.

