Practice exercises
GeneXus BPM Suite Course

GeneXus™ 17

December 2020

TABLE OF CONTENTS

To work on the following practice exercises, open GeneXus, create a KB and import the file GXBPMSuiteGX17_Practices.xpz containing all the necessary objects for this practice session.

MODULE I

PRACTICE EXERCISE 1. PURCHASE PROCESS

A clerk creates a purchase order and should receive a notice every 30 seconds, to remind him/her that a purchase order is pending to be sent. Once the clerk finishes the task, the system should not send any more notices and the process should continue.

After the order is defined, it must be authorized by both the Administration Manager and the Purchasing Manager. If they both approve the order, then the corresponding purchase order is automatically generated, and later issued (printed) by the clerk. If any authorization is denied, then the process must be terminated at that point.

1.1. BASIC PROCESS MODELING

In this section, the purchase process is modeled. For the time being you should ignore the deadlines related to tasks.

1. In GeneXus, create an object of Business Process Diagram type called Purchases and model the purchase process according to the description provided in the previous item. When associating objects with components from the diagram, use the following GeneXus objects:

Name	Type	Description
Request	Transaction	Purchase order
AuthorizationManager	Web panel	Authorization from Administration Manager
AuthorizationWarehouse	Web panel	Authorization from Purchasing Manager
GenerateOrder	Procedure	Generates a purchase order document based on an order made
Emission	Web panel	For printing purchase order

2. Use the *Request* transaction to enter purchase orders. With the Subject Rule property of the task, change the process subject to the following: 'Purchase Order No.: ' + &RequestId

3. Use the *AuthorizationManager* web panel for the Administration Manager's authorization, and the *AuthorizationWarehouse* web panel for the Purchasing Manager's authorization. Press F2 on the tasks to modify their names.

4. To model conditions analyze the logic of the *AuthorizationManager* and *AuthorizationWarehouse* web panels.

5. Use the *GenerateOrder* procedure to generate the purchase order from the order.

6. Use the *Emission* Web Panel to print the order.

7. Create 2 query Web Panels (*Consult Application*) for authorizations of the *Purchase* process and 2 preview Web Panels (*Preview Application*).

> **Note:** A **User Task** has the property **Consult Application** where you can associate a Web Panel that allows you to query from the process history what was the action performed in that step of the process. The **Preview Application** property, on the other hand, allows you to associate a panel so that the end user can assess whether he or she is the right person to take the task by viewing the information it contains without having to take it.

7.1. You need to create a Web Panel called *Consult Application* that receives the RequestId attribute as a parameter and must display:

    a. The order ID (RequestId)

    b. The order date (RequestDate)

    c. A grid with columns that include ItemId, ItemDescription, RequestLineQuantity and RequestLineAmount

    d. The order total (RequestTotalAmount)

    e. A textblock that displays `'The request has been approved'` if the order was approved and `'The request has NOT been approved'` otherwise, depending on the value of the attribute RequestAuthorizationManager or RequestAuthorizationWarehouse, as appropriate.

7.2. Create a web panel called *Preview Application* to display the information necessary to picture an idea of the work to be done in that task. In this case, only the order to be assessed is required; that is to say, to implement it we select SaveAs in the ConsultApplication Web Panel and then remove the textblock that indicates if the order was approved or not, deleting the event that loaded this textblock.

The GeneXus objects for both applications are associated from the tasks' properties.

8. After the diagram has been created, save it and execute it with the *Run* option.

1.2. WORKING WITH THE INBOX

1. Try out some of the executions in the process created by authorizing and rejecting purchase orders.

2. Try viewing the authorization tasks prior to their execution, through the "Preview" option in the Inbox located in the "More Actions" section. (*Preview Application*)

3. Access the *Outbox and My Processes*.

1.3. HISTORY

1. Open the process history.

2. Check the values in conditionals.

3. Select the authoritation tasks and press the Query button to view the decision made. (*Consult Application*)

4. Check the status of tasks and their duration. To do this, columns in the grid must be enabled.

5. Check if batch tasks are executed by the *WFENGINE* user (Workflow engine).

PRACTICE EXERCISE 2. PC REPAIR PROCESS

A technician makes a preliminary assessment of the problem and a diagnosis to indicate whether it is necessary to buy spare parts to do the repair. If parts must be acquired, then the Purchase process described must be followed. If parts must be acquired and the process is rejected, the PC must be returned to the owner.

The repair stage involves two technicians: one who does the repair and the other who tests the PC. They enter comments on the record regarding the repair or test they do in order to make communications easier. The testing technician must decide whether the repair was done correctly and if so, he sends the PC to the administrative office for it to be returned to the owner. If the testing fails, the PC is returned to the technician who repaired it so that the issue can be corrected.

2.1. ADVANCED PROCESS MODELING

To develop the Repair process, use the following objects:

Name	Type	Description
RepairRequest	Transaction	Information about the repair
RepairComments	Transaction	Enables the entry of comments
DeliveryPC	Web panel	To deliver the PC

1. Use the *RepairRequest* transaction for the initial evaluation task.

2. Define the condition that evaluates whether the purchase of parts is necessary or not. To do so, open the *RepairRequest* transaction to see where that information is stored.

3. Use a sub-process to link the Repair process to the Purchase process.

4. In the case where the repair requires the purchase of parts, define the condition that assesses the result of the purchase. What relevant data may be re-used to express such condition?

5. Use the *RepairComments* transaction for the repair and testing tasks. Note that technicians must make a decision at their own discretion. Think about which would be the most appropriate route type in this case.

6. Use the *DeliveryPC* Web Panel to model the delivery to owner.

2.2 EXECUTION

1. Execute the processes again, trying to initiate the purchase process from the repair process.

2. Verify that everything works fine, and go over the process history.

3. From the Process Manager verify if the values of the Relevant Data are correct.

MODULE II

PRACTICE EXERCISE 3. WORKFLOW DATA TYPES

By means of the Workflow API, modifications will be made in the behavior of some tasks of the business process.

3.1. BASIC WORKFLOW DATA TYPES

Make the following changes to the purchases diagram:

a. Change the implementation of the Administration Manager's assessment so that a condition assessment procedure is used. The procedure must use the value loaded in the *RequestAuthorizationManager* attribute and return a value that is coordinated with the values set in the routes coming out of the conditional.

b. Change the assessment implementation of the Purchasing Manager in a way similar to the exercise described above, using a conditions assessment procedure. In this case, instead of reading an attribute from the database, define Relevant Data called *AUTHWAREHOUSE* and depending on the result of that relevant data, the path to follow will be defined. For this to work, you must modify the *AuthorizationWarehouse* web panel used in the authorization and then change the code so that instead of saving the decision, as it happens now, it will be saved in the relevant data, and this value is evaluated in the condition.

> **Note:** These exercises are only meant to practice the use of Workflow Data Types. The solution as it was before these changes is undoubtedly the best one because it implies less programming by the developer and the problem is solved correctly.

3.2. ADVANCED WORKFLOW DATA TYPES (COMPUTER RECEPTION PROCESS)

When a new customer arrives at the shop, the receptionist receives the customer, and in the case of a repair job, the customer is assigned to the corresponding assistant.

The assistant receives the PC and starts the Repair process implemented in the previous practice exercises.

1. Model the computer Reception process in the *Business Process Diagram*.

2. Develop the application used by the receptionist, according to the following guidelines:

   - The receptionist DOES NOT use the Inbox, so the process must be created outside the Inbox.

   - Use the *Reception* web panel as a guide to implement the application.

   - To create the process outside the *Inbox*, check the video titled "*Starting a process from a GeneXus object using the Workflow API*" and/or read the documentation on Workflow Data Types in the GeneXus wiki.

   - Change the subject of the process with the name of the customer entered at Reception.

   - To assign service to a user:

     - Obtain the organizational model.

     - Load users of the organizational model in the sub-file.

   > **Note:** So far only the user "WFADMINISTRATOR" is available and you can work with it, but for this exercise, if you wish, you can enter more users from the Management Console of the Workflow client.

   - With the user selected, use the pre-assignment operation for the customer service task.

3. In the customer service task use the *EntryRepair* transaction of the Repair process and connect the two processes so that access to the information entered in the Reception process is possible from the Repair process.

PRACTICE EXERCISE 4. DEADLINE AND WORK WITH DOCUMENTS

4.1. DEADLINE OF PURCHASE PROCESS

1. Modify the Purchases process to include in it the deadlines for defining purchase orders. Remember to make sure that the timer will not interrupt the task when the deadline expires.

2. Once the time (30 seconds) has elapsed, a reminder task must be sent to the user.

3. After the reminder is executed, an Event Gateway associated with another timer will ensure that the message is repeated every 30 seconds until the purchase order creation task is completed. At that time, a signal must be sent to end the automatic reminder pattern.

4. Execute the apwftimerscheduler.exe *30* program by command line to start controlling the deadlines.

5. Execute the processes to verify that everything works correctly.

> **Optional:** Another option to solve the requirement is through a single intermediate timer of Cycle type. What changes should you make in the diagram?

4.2. WORK WITH DOCUMENTS

1. Modify the repair process to allow the technician to attach a photo of the repair performed; this is a required option.

2. The technician in charge of testing should see the image of the repair.

3. Remember that when you are going to work with documents you must first define them at the knowledge base level. (Preferences -> Workflow -> Documents.)

4. Execute the process and do not forget to configure in the client the use of documents and test the modifications made.

MODULE III

PRACTICE EXERCISE 5. PROCESS EXECUTION AND MONITORING

5.1. MANAGEMENT CONSOLE

1. Add in GeneXus the "Clerk" and "Chief" roles from Preferences – Workflow – Roles.

2. Open the purchases diagram and assign the following roles:

   - Start symbol: Clerk
   - Request: Clerk
   - Authorizations: Manager
   - Issuance: Clerk

3. So far, the Workflow client was executed in prototype mode. Now, to test security we will change it to Standard so that it will request us to log in.

   To do so, change the Environment's "Execution Mode" property, and set it as "StandardClient."

4. Run the purchase process.

   To enter with the *Administrator* user, the information to be used is:

   - User: WFADMINISTRATOR
   - Password: WFADMINISTRATOR (both in capital letters).

5. From the *Management Console* create two new users: *user1* and *user2*. To *user1* assign the Clerk role and to *user2* assign the Manager role.

6. Now, using only *user1* and *user2*, execute the Purchase process. You will have to enter *user1* first and create the process. And then enter *user2* for authorizations, etc. Execute the *Outbox* and *My Processes.*

5.2.   PROCESS MANAGER

1. From the *Process Manager* abort and suspend some processes, and verify that the related tasks are not in the *Inbox.* Resume the processes and verify if they are again the *Inbox.*

2. Notice the different values of relevant data and status of tasks.

3. Enable and disable the process definitions and note what happens with the instances of processes and tasks. Remember that, by default, only enabled processes in the *Process Definition* applicative are shown. To view disabled processes you must change the corresponding filter in the first column of the grid.

5.3. ORGANIZATIONAL UNITS

Using organizational units, solve the problem that implies that tasks at a branch should not be viewed by the other branches.

- Define the organizational unit for branches.

- Assign organizational units to the users in each branch.

- Enter GXflow as one of these users, create new processes and verify that the organizational unit is correctly assigned.

- Access the *Process Manager* and verify that the processes have the organizational units assigned.

5.4. PROCESS ANALYSIS AND PERFORMANCE

Log in as Administrator to GXflow's client and go to the statistics section. Look at the different analysis options at the process and task level and analyze each one of them.

OPTIONAL

PRACTICE EXERCISE 6. MULTI-INSTANCE

Suppose that the purchase process must be modified so that the order may be authorized only by the Purchasing Manager. Also, authorizations must be done on an item-by-item basis.

For example, for the following purchase order:

Qty.	Item
1	Notebook
2	Monitor
1	Printer

The Purchasing Manager will have three tasks in his Inbox: one to authorize the three notebooks, one to authorize the two monitors, and third one to authorize the printer purchase. The Administration Manager's authorization task no longer exists.

On the other hand, the purchase order generated will only contain the items authorized by the Purchasing Manager.

Make the changes necessary in the purchase process in order to reflect this new reality.

Note: To develop these modifications in the application, we will use several properties of the activities together with the Workflow Data Types.

1. In the second level of the *Request* transaction, add an attribute for representing the result of the authorization of each item in the purchase order.

2. In the Purchasing Manager Approval task, in the group of properties *Advanced Properties->Looping*, modify the *Loop Type* property with the *Multi-Instance* value and the *Ordering* property with the *Parallel* value. This means that at the time of creating, the activities created will be more than just one; there will be several parallel Approval activities.

3. To indicate the number of tasks that will be created, we can use the *Expression Rule* property, also located in the *Looping* group. This property allows specifying the arithmetic expression that will determine the number of instances that will be created for the task involved. This expression may include constants, relevant data or attributes of the transactions present in the diagram. For the specific case of this exercise, we could define, in the Request transaction, a formula attribute with the items of the order and then use it to define the number of authorization tasks to be created.

4. Define relevant data called Items of the numeric type 4 and with the *Dimension* property with value *Vector*. Each ItemId in the purchase order will be loaded in this relevant data.

5. Create a procedure to receive as parameter the ID of the purchase order. This procedure must save all the IDs of the items in the order in the Items relevant data (use the *SetValue* operation of the W*orkflowApplicationData* data type). In the *After Trn* event of the *Request* transaction add the call corresponding to this procedure.

6. In the Purchasing Manager's Authorization web panel, make the following changes:

    a. Obtain the *Items* relevant data.

    b. Each of the workitems associated with the Approval task has an index to tell it apart from the rest of the workitems in the same Approval task. So, use the *Index* property to obtain the workitem's index.

    c. Using the GetValue operation of the *Items* relevant data and the workitem's index, recover the ID of the item corresponding to be approved in this task.

d. Don't forget to set a condition to filter by ItemId.

7. Implement a procedure to update the attribute added in Step 1, depending on whether the line has been authorized or not. This procedure must be invoked in the events of the *Authorize* or *Reject* buttons in the Purchasing Manager's Authorization web panel.

8. Modify the *GenerateOrder* procedure, so that only the lines approved are loaded in the Purchase Order.

9. Test the process at runtime; you should get as many Approval tasks as lines in the order. Upon execution of each of these tasks, there should only be a line with a single item in the order. Lastly, in the Issuance task, only approved items should appear.