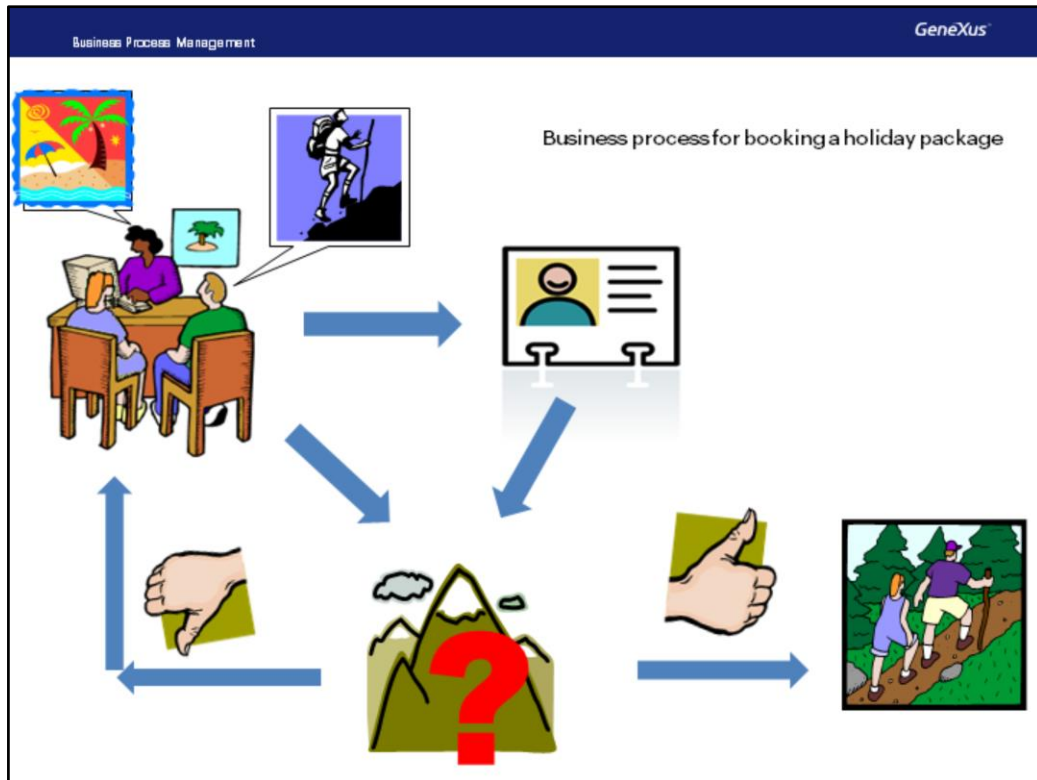


## Business Process Management (BPM)

*GeneXus* 16



Some procedures at the travel agency call for several tasks to be carried out one right after another in a given order. One example is in the case of a reservation for a holiday package at a specific attraction.

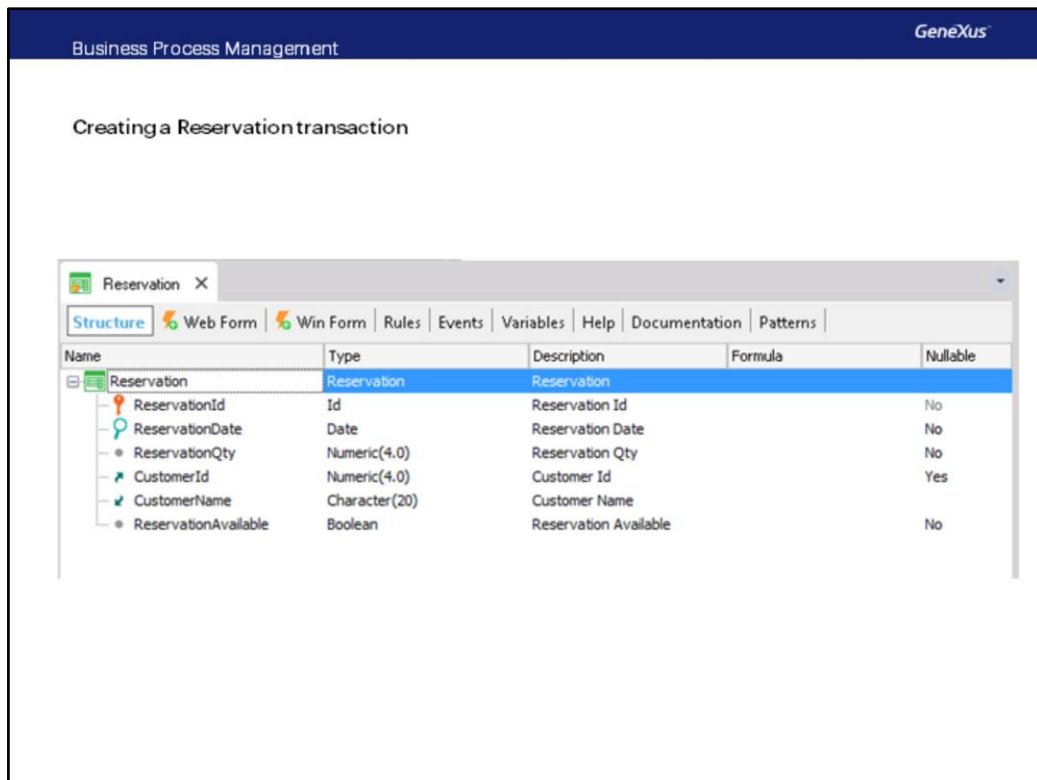
When entering the reservation, we must verify that the individual booking the package is a customer of the agency. If he/she is not, then we must enter him/her as customer.

Having done this, we must verify the existence of packages available for the number of individuals who want to make the trip.

If there are enough seats available the reservation is assigned. Otherwise, the passenger is offered a different package.

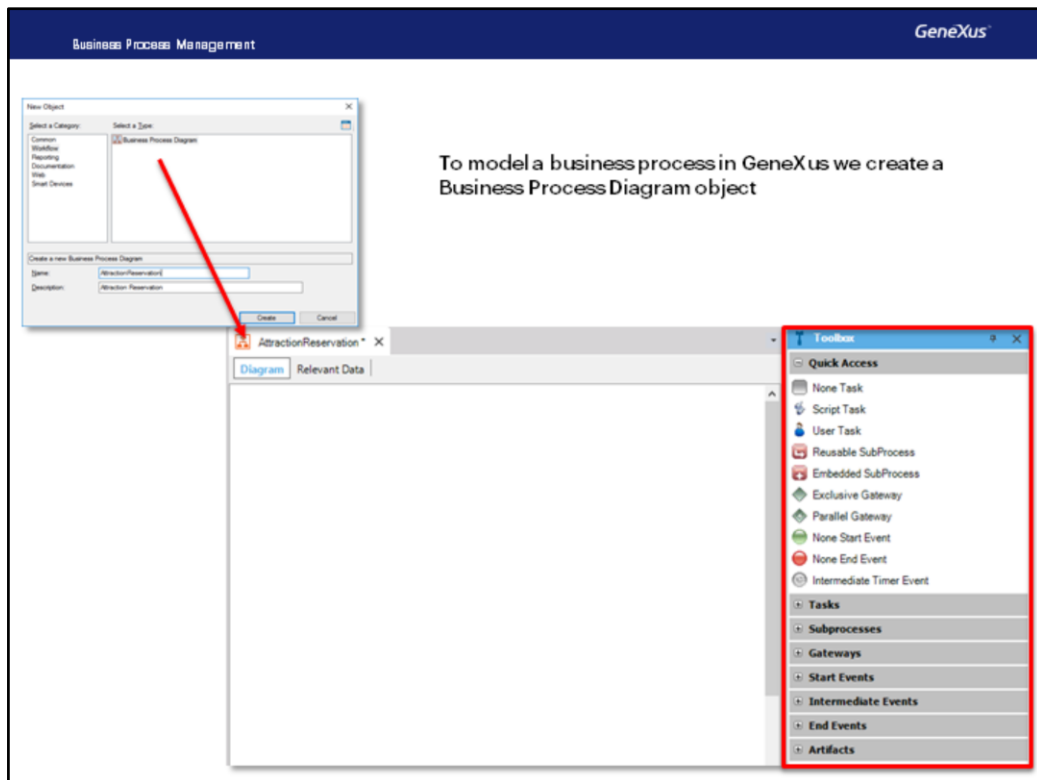
A sequence of steps like the one we just described is what we call business process.

GeneXus enables us to define processes like this one. Let's try it!

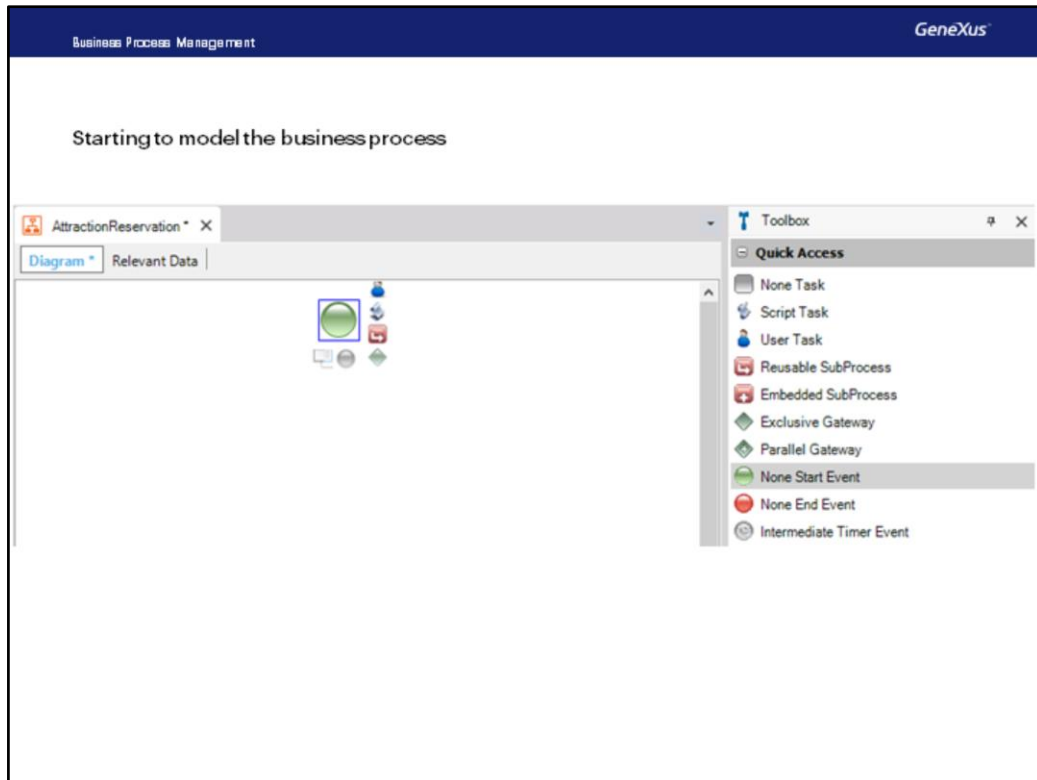


Because we need to record the reservations for holiday packages, what we did in advance was to create a reservations transaction, with attributes to store the identifier, the date of the reservations, the number of passengers who will be using the package, an ID and name of the customer related to the reservation, and ReservationAvailable to mark whether the holiday package is available or not.

For the CustomerId attribute we select the Nullable column with value Yes, to indicate that, at the time of entering a reservation, we might still not have the customer ID for the individual making the reservation.



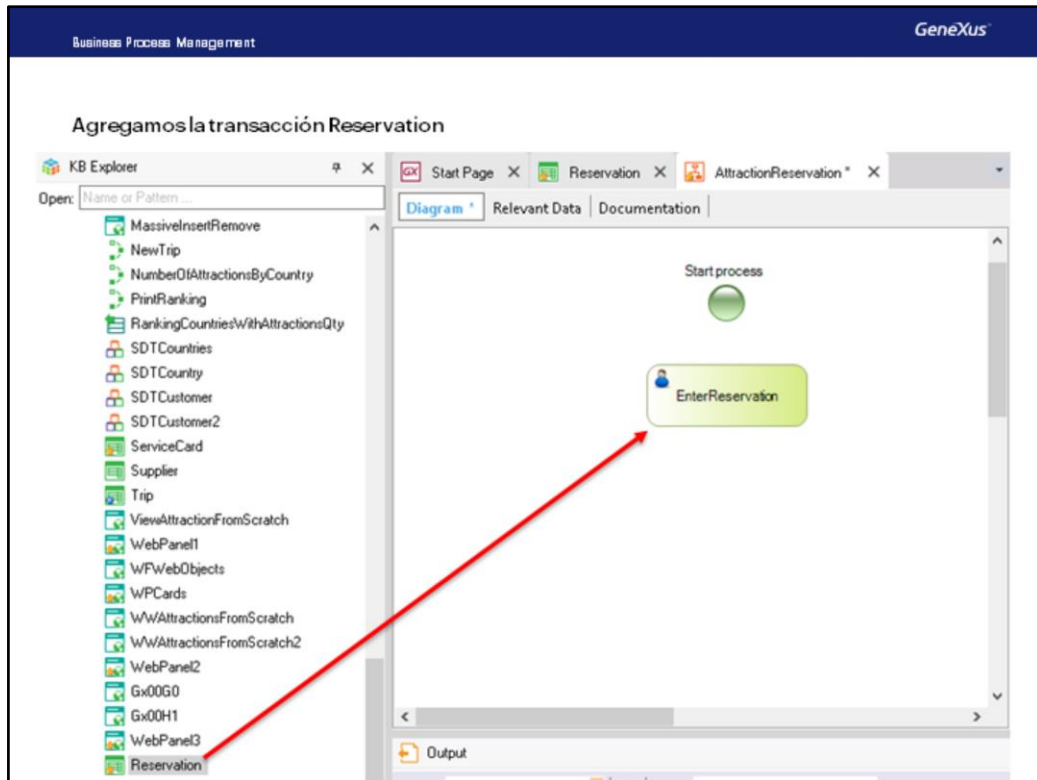
We now go on to create an object of the business process-diagram type. We select File... New... and select Business ProcessDiagram object, which we will call AttractionReservation.



We see a blank page of the diagram we will build. If we open the toolbox we can see that there are a great number of symbols that we can apply.

To mark the start of our business process we drag and drop a NoneStartEvent symbol. We can see that when the symbol is selected, around it a series of symbols are shown, which we could use to continue the diagram. This suggestion saves us a lot of time, since we don't have to go to the toolbox to drag other symbols.

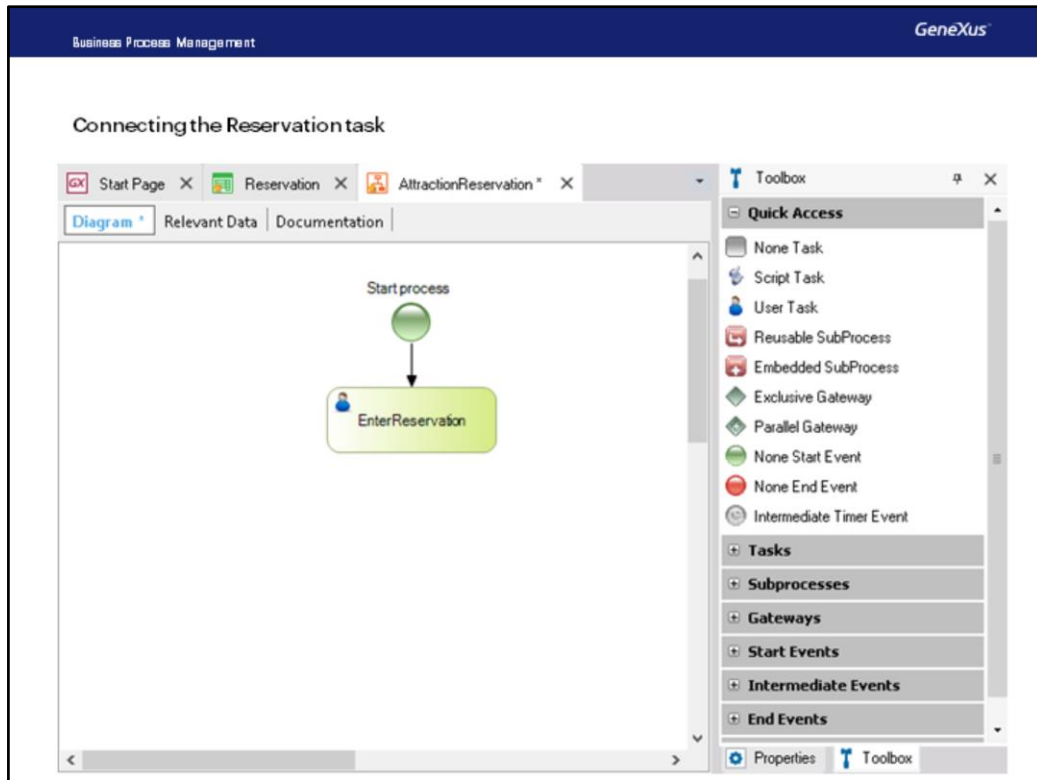
We press F2 on the None Start Event and write Start process.



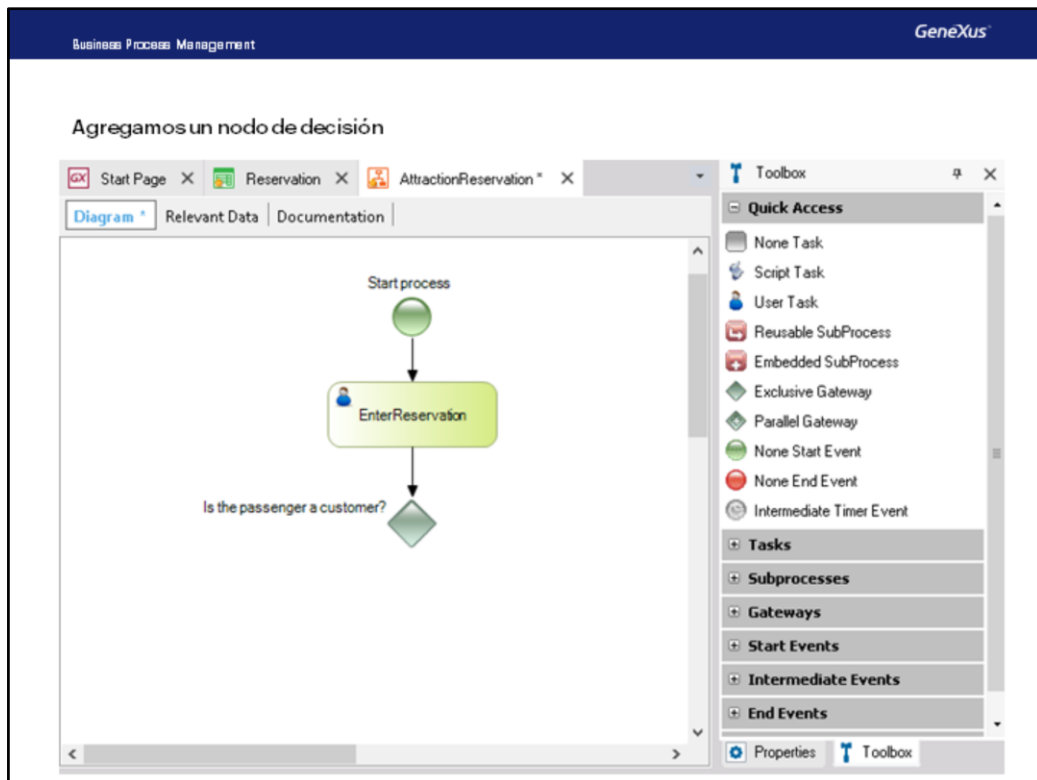
Because our first task will be to enter a reservation, we place the Reservation transaction in the Folder View window and drag it into the diagram.

We can see that a green rectangle with the name Reservation has been created, with the man icon in the upper left. This indicates that an interactive task has been created, since the transaction allows for a user to enter data through it.

We press F2 and change the name to EnterReservation.



To connect the Start node with the transaction, we click on the lower section of the green circle and holding the mouse button we drag until the tip of the arrow reaches the upper border of the transaction rectangle.

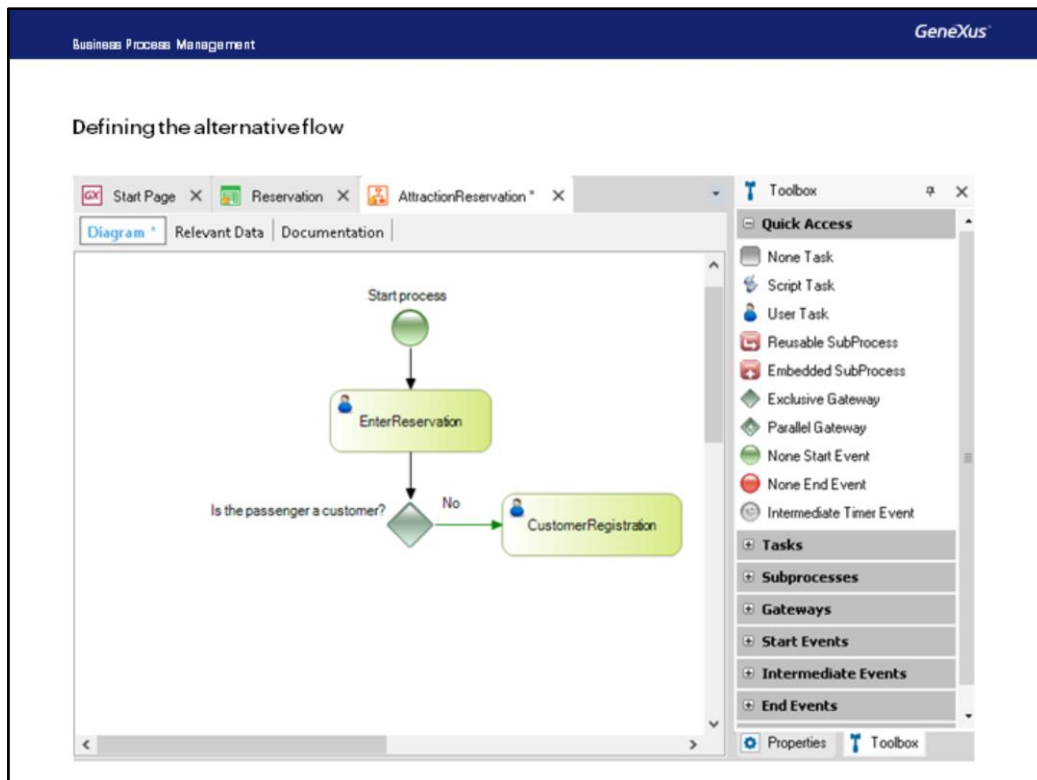


According to the process used at the travel agency, the system must control whether the individual making the reservation is already a customer or not.

To represent a decision on the diagram, we go to the toolbar, click on Gateways, drag an Exclusive Gateway node over the diagram and link it from the Reservation transaction. With F2 we write the question "Is the passenger a customer?" and then clicking on it we position the text.

This type of node assesses a condition, and depending on the results, it makes the flow continue downstream (which would be the normal route) or to the right of the symbol (the alternative route).



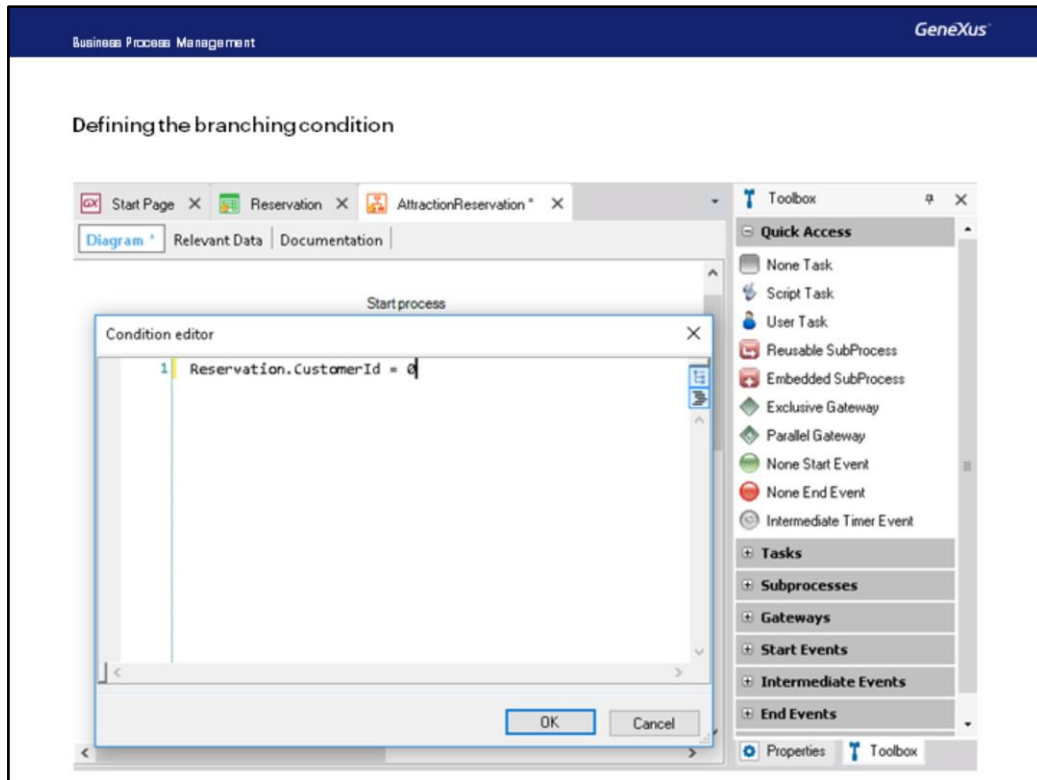


In our example we must define the condition for the Gateway which will make the flow to either go along the regular path or through the alternative path in order to add a customer.

In other words: if the individual making the reservation of the holiday package is not a customer of the company, then the Customer transaction must be invoked to add the new customer.

To begin, we drag the Customer transaction to the diagram and connect it from the Exclusive Gateway node. With F2 we change the task's description to CustomerRegistration.

We then select the connection and with F2 we add the label "No".

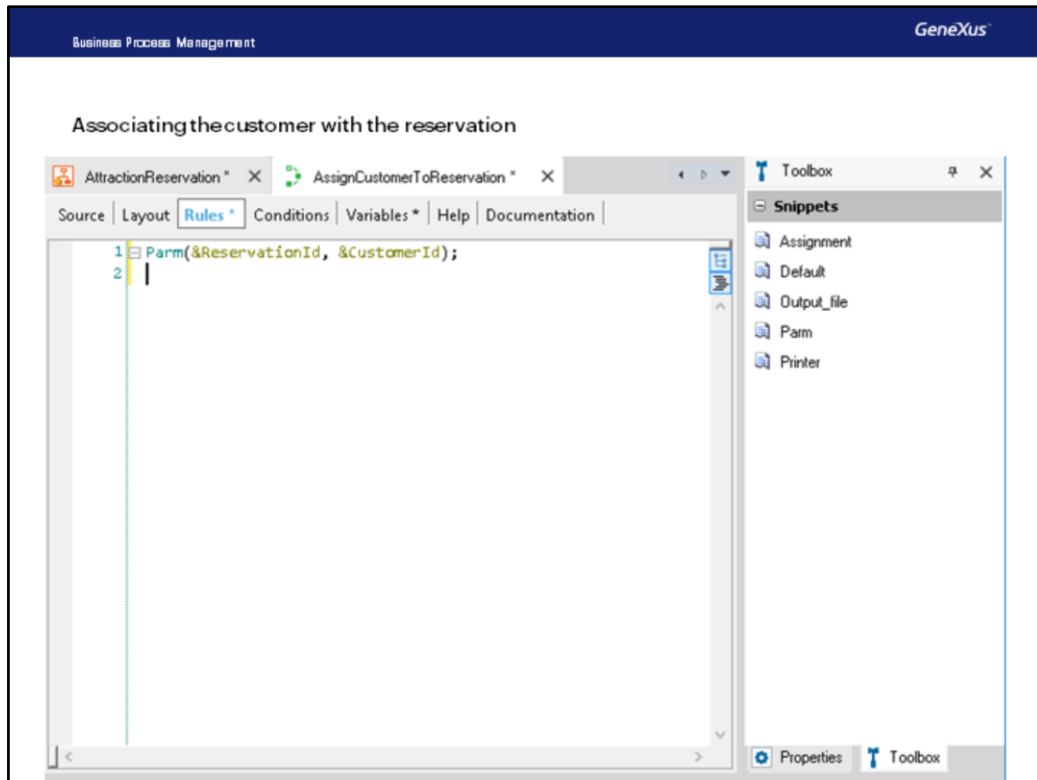


To complete the definition of the decision we have to add the condition that will, from the decision node, cause the process flow continue to the right invoking the Customer transaction, or go downwards to the following task.

In our case, the path to the right would be the alternative flow, while the downward path would be the usual flow, that is, when the individual making the reservation is already a customer of the travel agency.

To define the condition that will cause the diverging paths, we double click on the green arrow that links the Gateway with the Customer transaction, and we will see that a window of the **conditions editor** opens up.

We write the following: `Reservation.CustomerId = 0`, to indicate that the flow must take that route if the CustomerId attribute had value 0 when the reservation was entered.

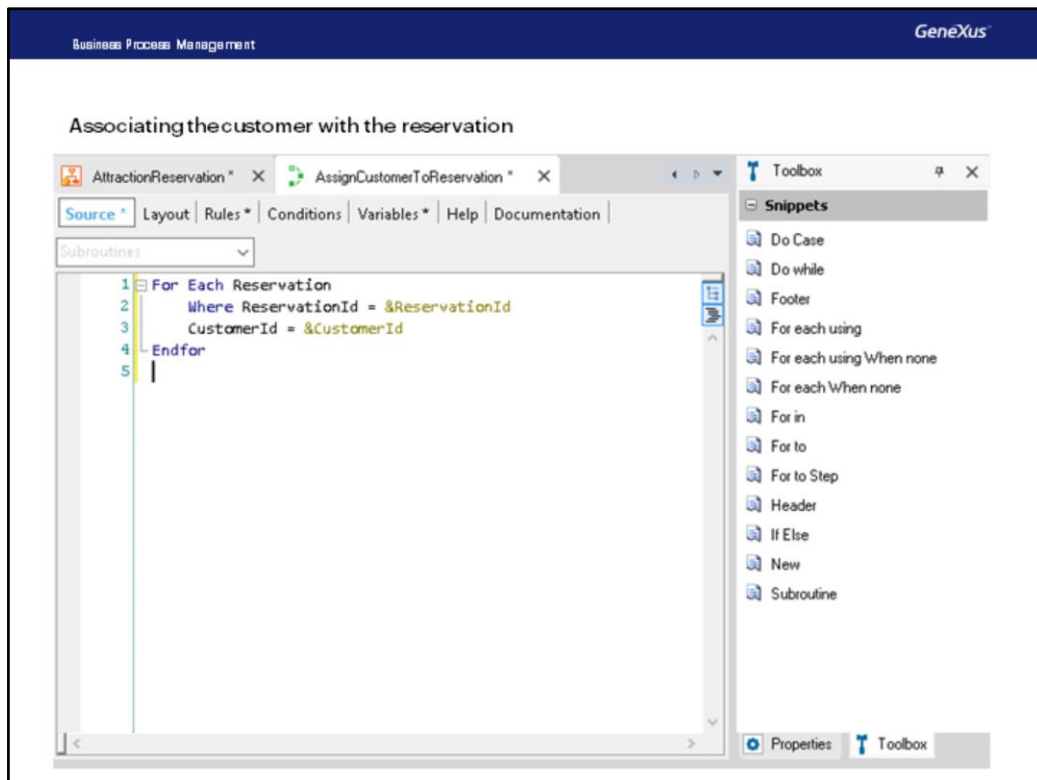


Following this alternative path the Customer transaction will open for us to add, as a customer, the individual making the reservation.

Once we confirm the entry we must associate the customer we just created with the reservation.

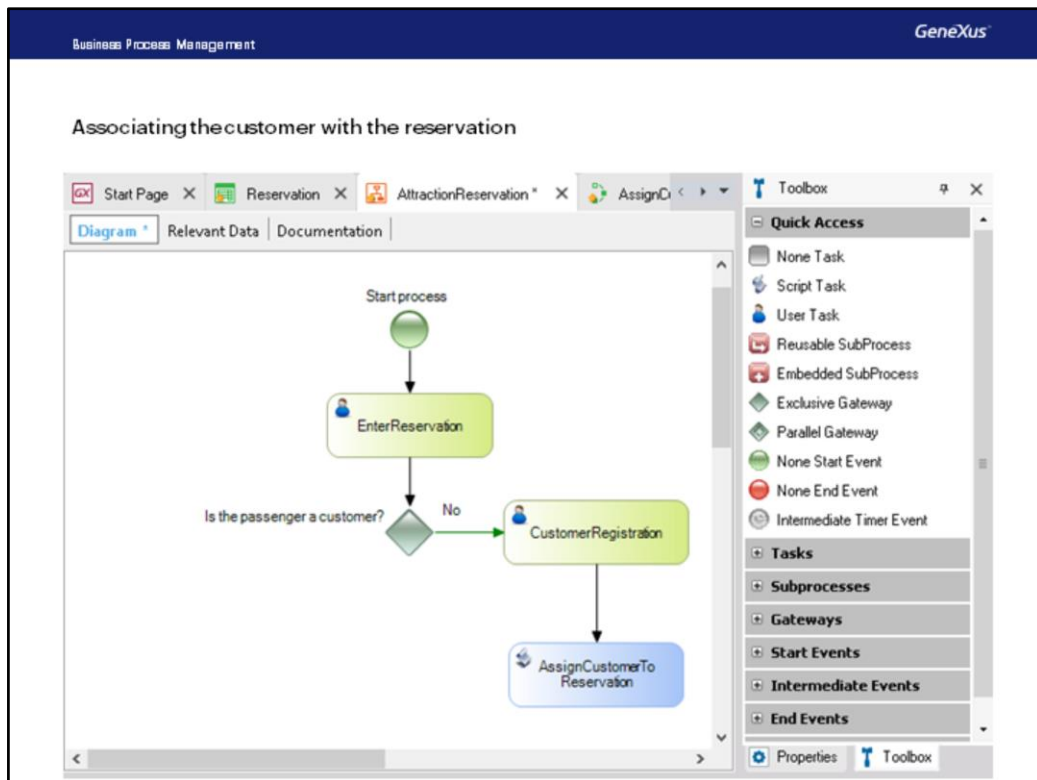
To do this we must create a procedure... which we will call AssignCustomerToReservation, and in its rules section we will write a parm rule with the parameters &ReservationId and &CustomerId...

These variables store the identifier of the reservation we created at first and the customer identifier that we want to associate with that reservation.



In the source we write ... For each... WhereReservationId equal... to the ReservationId variable we receive by parameter. Next, to CustomerId we assign the value of the &CustomerId variable, and then close the For Each.

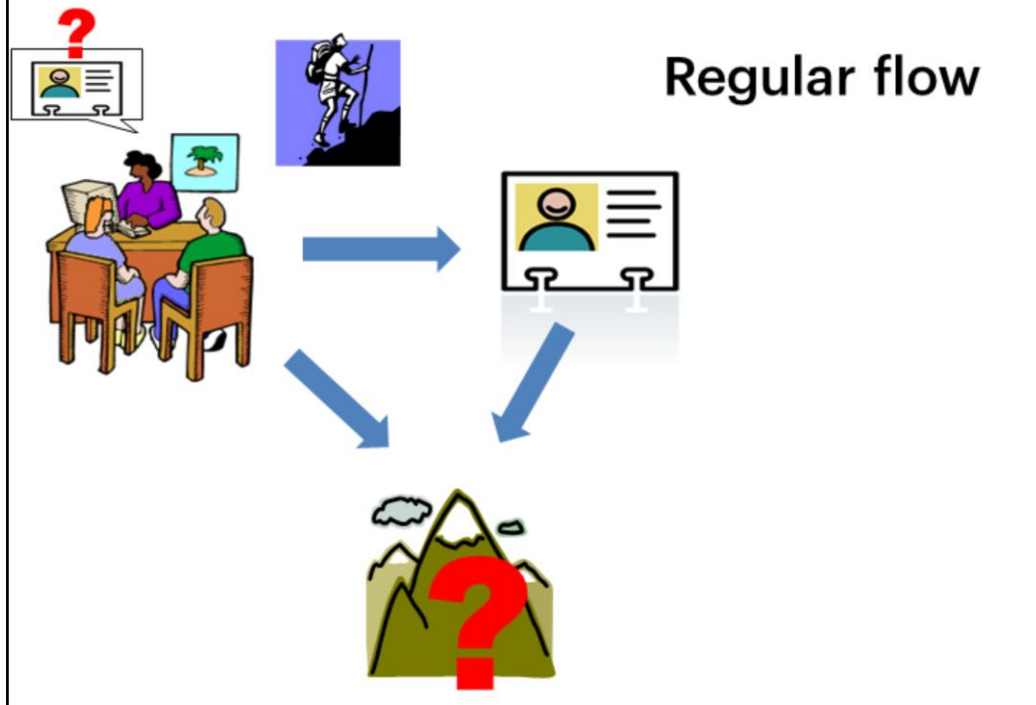
This is how we assign the customer just created to the reservation that was entered before.



We now drag the procedure we just created to the diagram. We can see that a blue rectangle was generated, meaning that the task created is of the non interactive type, also called batch type.

This is because the procedure is executed without human intervention, so there is no interaction with the user who enters data, as is the case with transactions.

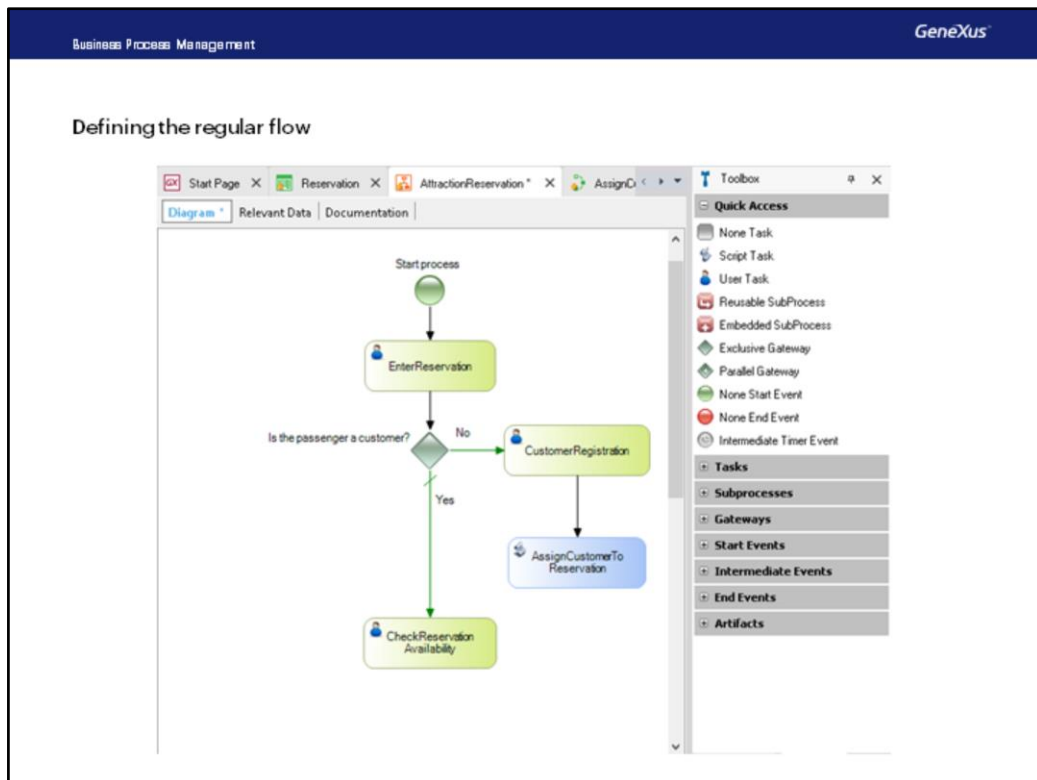
And lastly, we connect the AssignCustomerToReservation procedure from the Customer transaction....



So far we have described the tasks to be done when the alternative flow is executed, that is, adding a new customer and assigning it to the reservation.

We now add the activity that will be done if the flow is the normal one, when it is not necessary to invoke the customer transaction.

At this point, the agency's employee must verify availability for the reservation (like checking flights, availability of the package requested for specific dates, among other things) and then instruct the system as to whether the reservation may be completed or not.



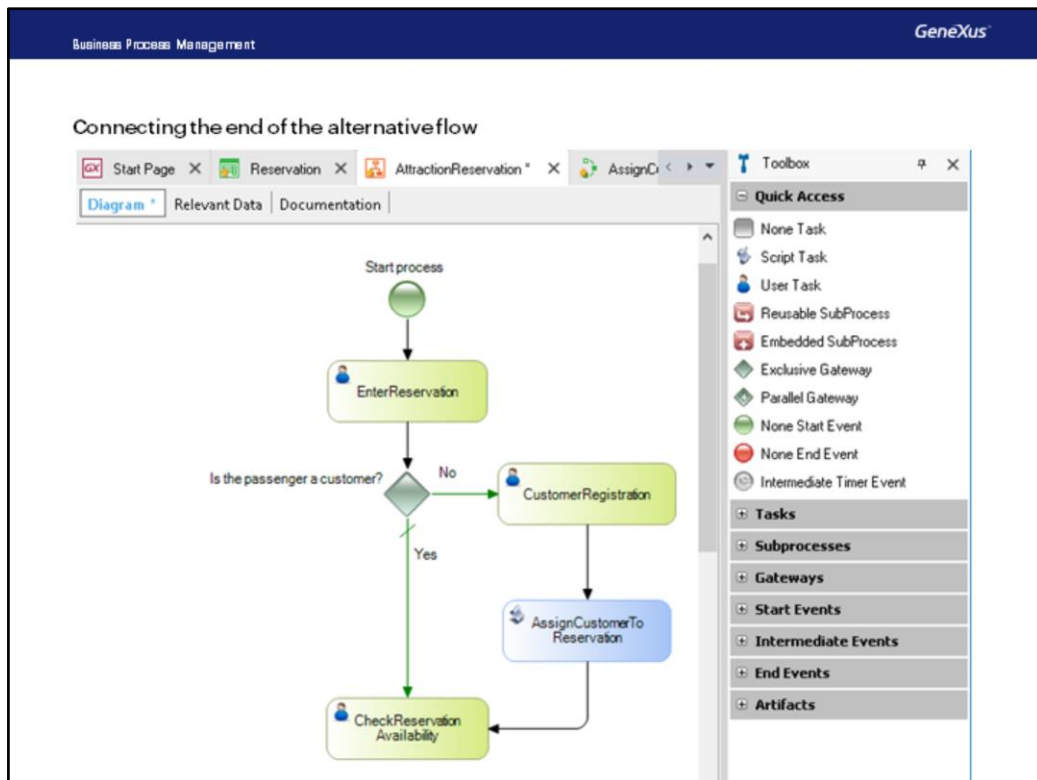
To enter this data we must have a screen where we will confirm or cancel the reservation. To do this, we can open the reservation transaction again and mark it as available or not, with the ReservationAvailable attribute.

To do so, we place the reservation transaction in the KB Explorer window, drag it to the diagram and connect it from the Gateway.

We see that an activity was created under the name Reservation1, so we press F2 and change its name to CheckReservationAvailability. Note that we are not changing the name of the transaction here, but rather the activity of the diagram that invokes the Reservation transaction.

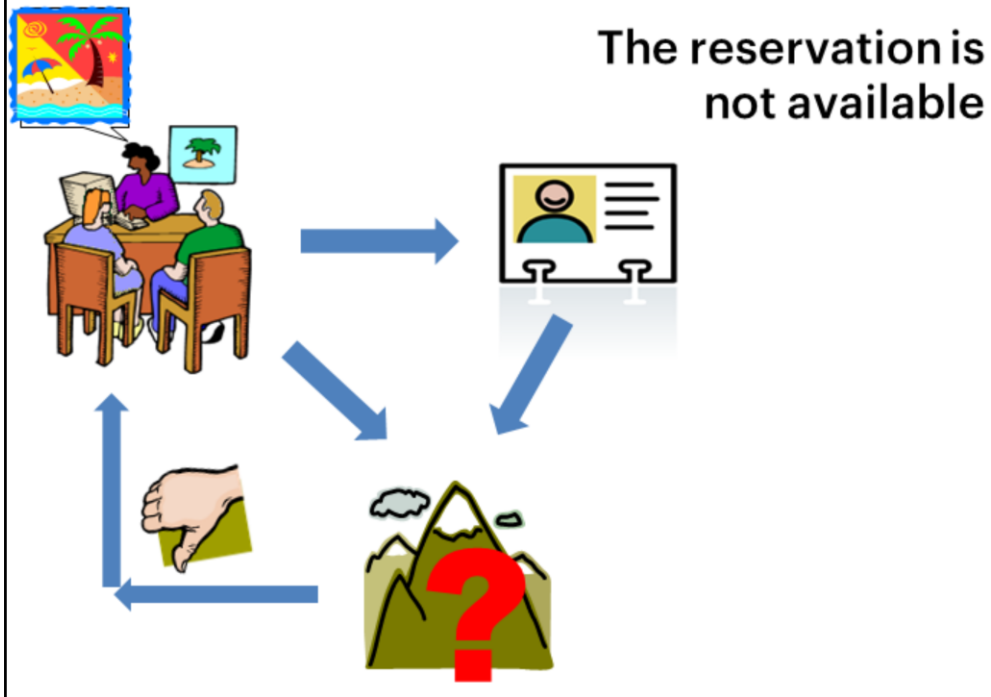
When we inserted the Gateway that allowed us to decide, we said that the flow to the right aimed at the Customer transaction was the alternative flow and that the flow downwards was the regular flow, when the individual making the reservation was already a customer.

To indicate that the regular flow is the one going downwards we select the connection and in the Properties window we set the ConditionType property with the Default value. Note that, on the diagram, the flow is shown with a green line across it. We press F2 and label it with the text "Yes".



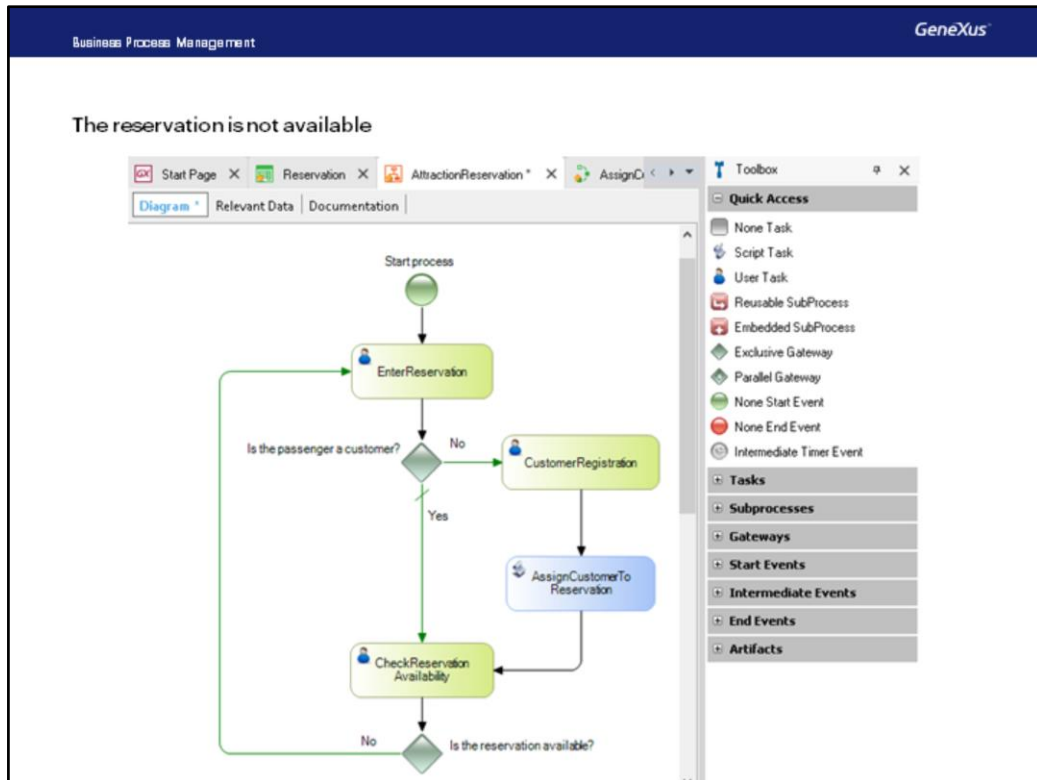
We can now connect the AssignCustomerToReservation task to the CheckReservationAvailability task, because once the customer assignment is completed, we must also continue by verifying availability for the reservation.





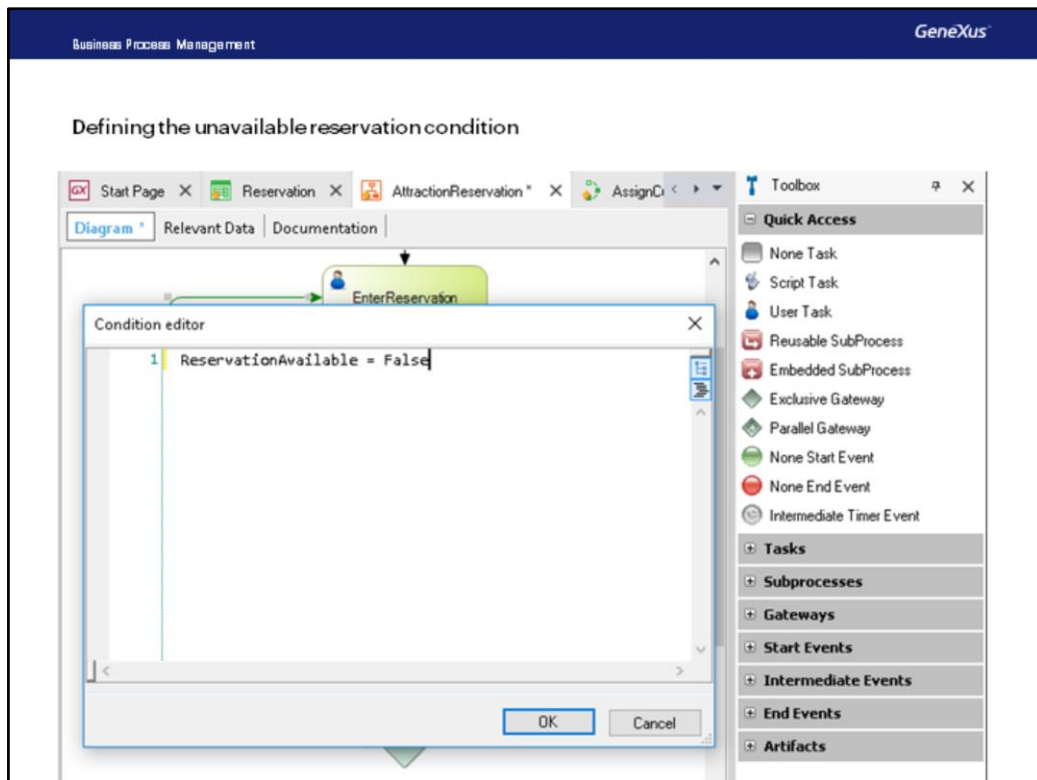
After the completion of the ReservationAvailability task we must assess the value we set for the ReservationAvailable attribute.

If the checkbox is not checked, this means that the reservation is not available and we must offer the customer a different package.



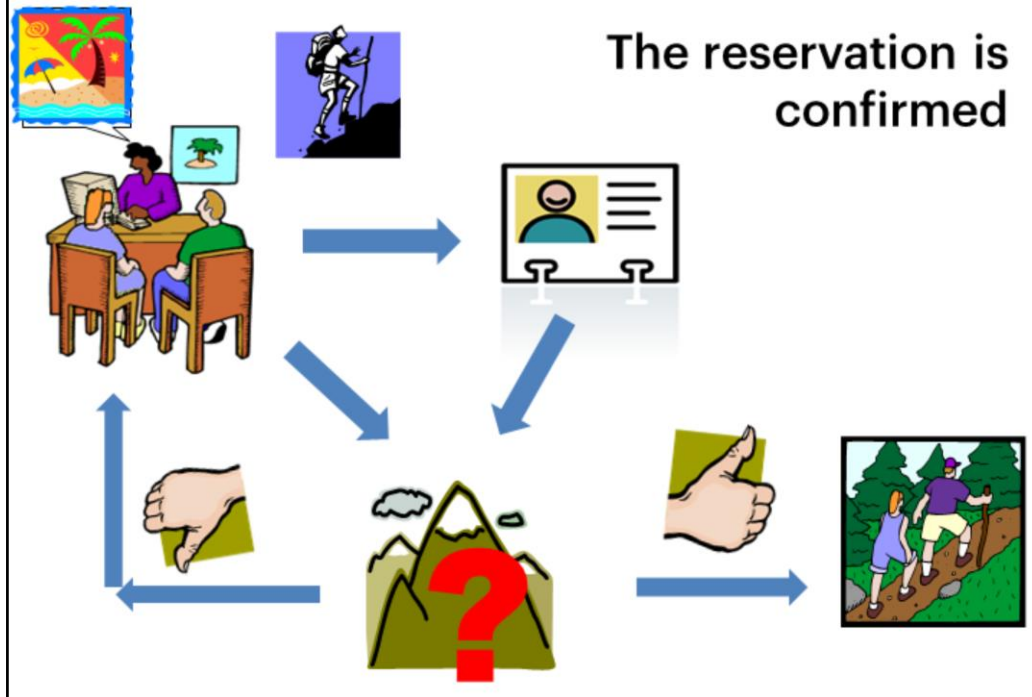
To show this, we insert from the toolbar an Exclusive Gateway and connect it from the ReservationAvailability task. We change its description to "Is the reservation available?"

We then connect its alternative flow (which we draw to the left), to the EnterReservation task, to allow for the addition of a new reservation. We label the connection with "No".



Now we must enter the condition necessary for the alternative flow to take place, that is, when the reservation is not available.

To indicate this on the diagram, we double-click on the connection that is oriented to the left and write `ReservationAvailable=False`.

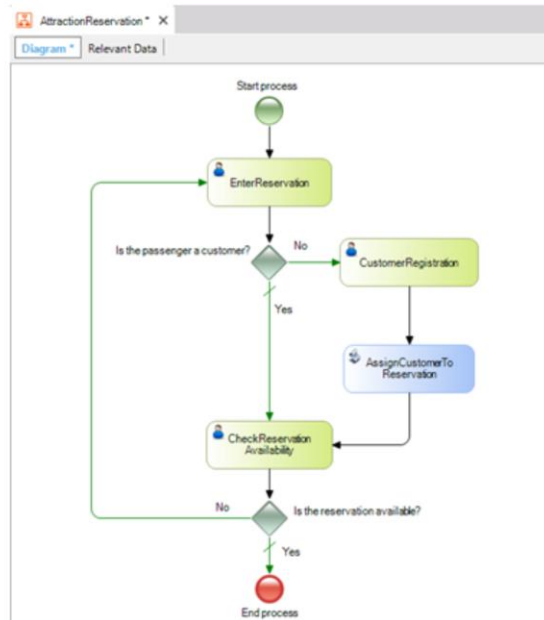


To this point we have analyzed everything that occurs in the case of having to offer an alternative for the customer to travel, since we could not confirm the reservation.

What is left to do now is to take into account the case where a reservation is indeed confirmed.

In such case we will not have any more tasks and the process comes to an end.

## Defining the end of the process



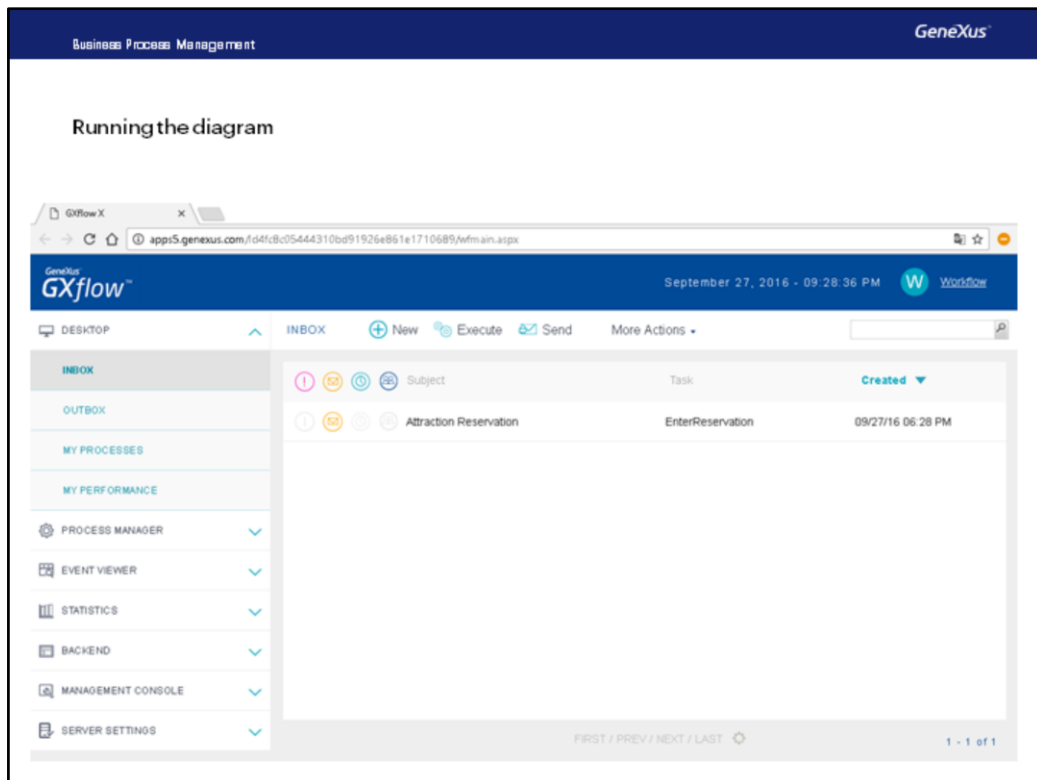
To indicate that we want to end the diagram, we insert, from the toolbar, a NoneEndEvent symbol and connect it from the Gateway. We assign it the text "End process".

This connection downwards is the Gateway's regular flow, so when the reservation is confirmed, the process will be ended.

To indicate this we select the connection and in the Properties window we insert its ConditionType property with the default value. Finally, we assign to the connection the text "Yes".

We have now completed the diagram for our business process.

To test its operation, we will now execute it.



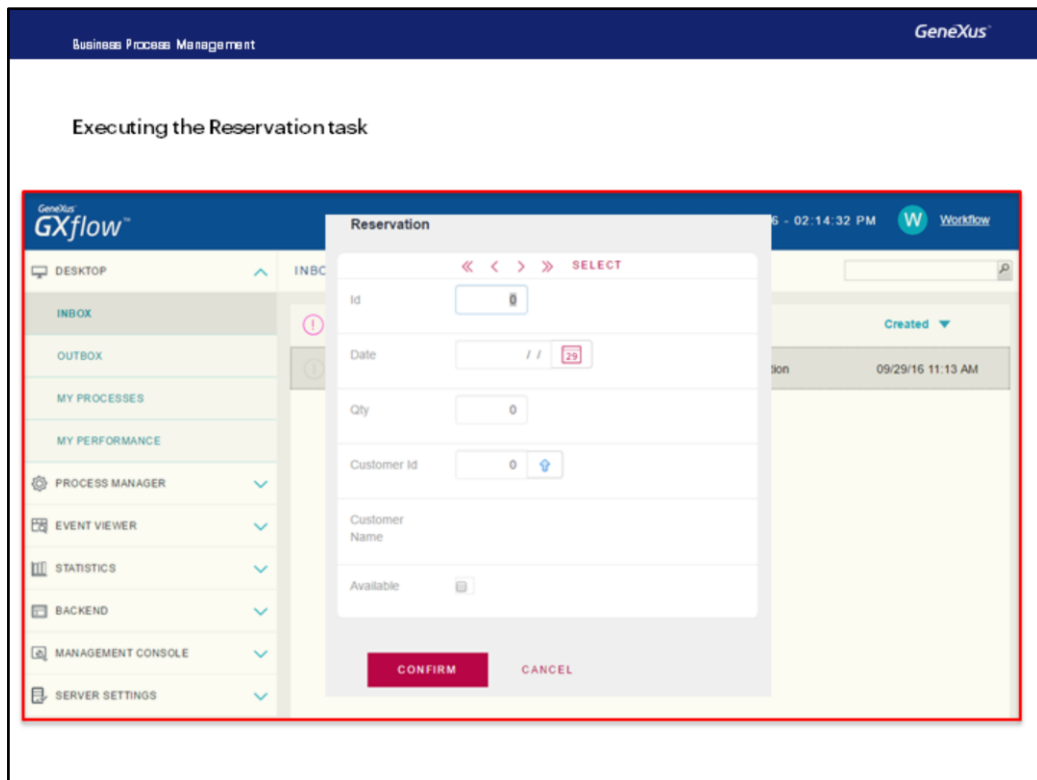
We right-click on the tab with the name of the diagram and select Run...

In the impact analysis, GeneXus informs that the Reservation table will be created, which was expected, so we proceed to reorganize and go on with the execution.

We can see that a screen is opened with the GXflow Client, which has the appearance of an email inbox where for each user we see the tasks pending, according to the process defined.

If we look at this inbox, we can see a closed envelope and in the Subject column we read: AttractionReservation. This tells us that we have defined this process and in the Tasks column we have the EnterReservation task pending.

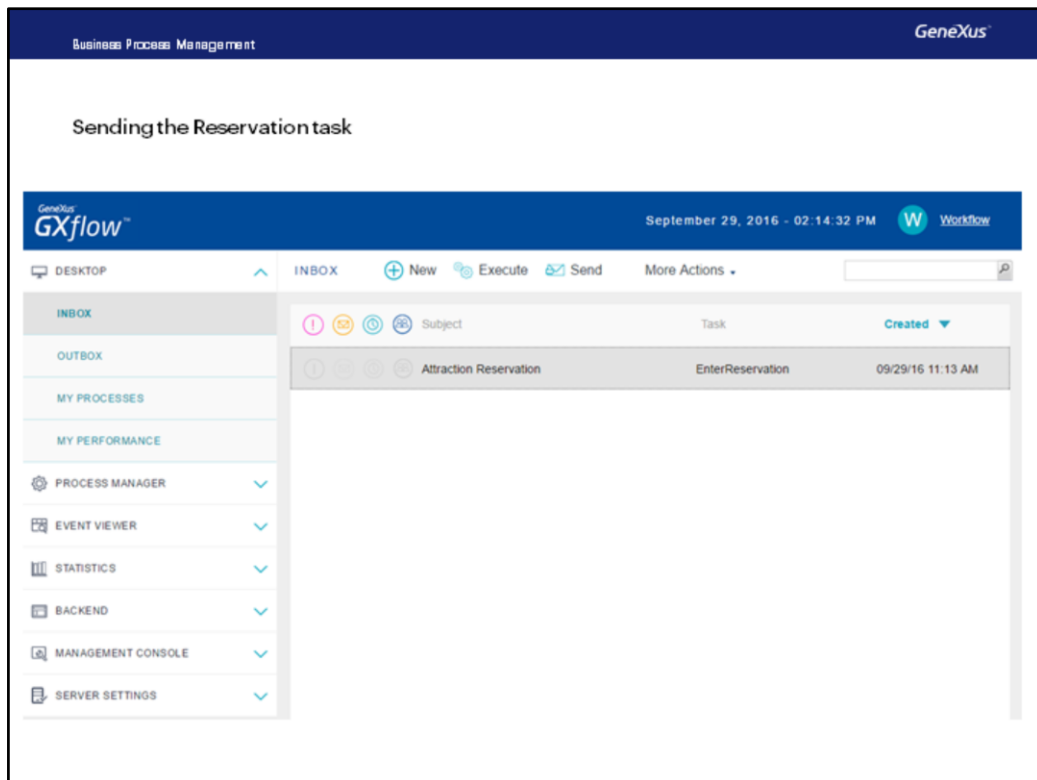
To execute it we select it and then press the Execute button or otherwise double-click on the task...



The Reservation transaction is then executed, which is the GeneXus object which is associated to the Enter Reservation task, since to create this task we dragged and dropped the transaction on the diagram.

We leave the ID blank because it is autonumbered, enter the date ... the number of passengers which is 2... and leave the value of CustomerId in 0, because the person making the reservation still isn't a customer of the agency.

We then press Confirm.



Now the task no longer has a closed envelope, as it is open, with a symbol showing that the task has been performed.

To go on to the following task we press Send.



Business Process Management

GeneXus

Next task: Customer trn

GeneXus  
GXflow

September 29, 2016 - 03:09:00 PM

Workflow

DESKTOP

INBOX

New

Execute

Send

More Actions

INBOX

OUTBOX

MY PROCESSES

MY PERFORMANCE

PROCESS MANAGER

EVENT VIEWER

STATISTICS

BACKEND

MANAGEMENT CONSOLE

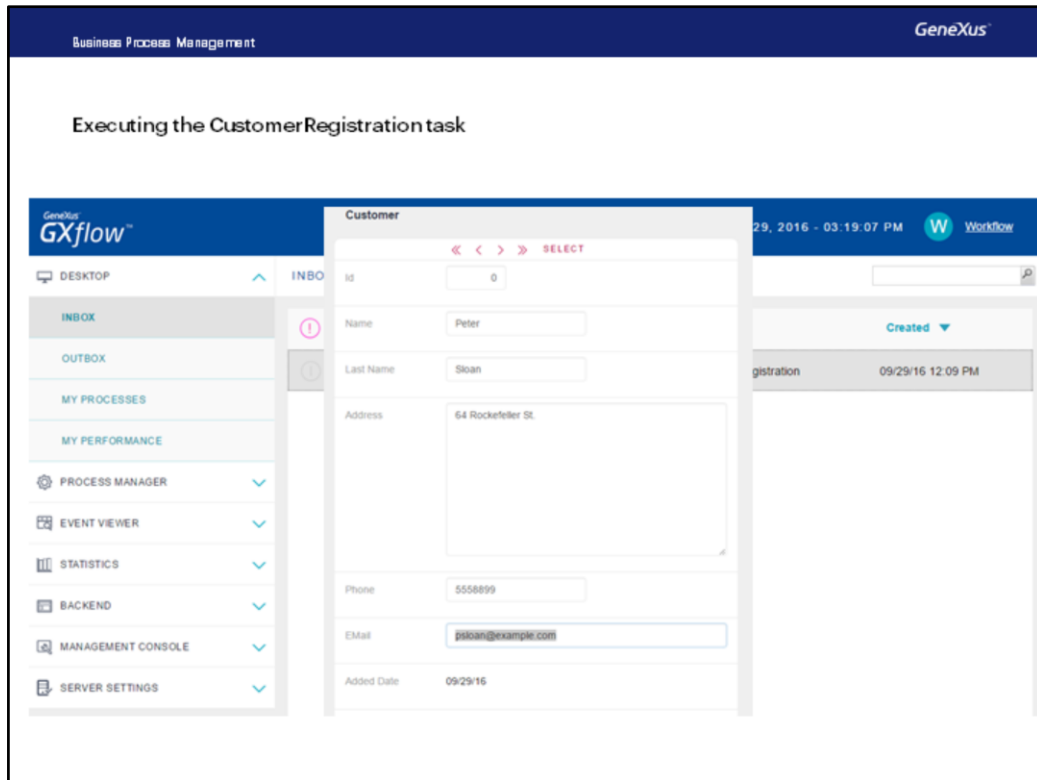
SERVER SETTINGS

Subject	Task	Created
Attraction Reservation	CustomerRegistration	09/29/16 12:09 PM

FIRST / PREV / NEXT / LAST

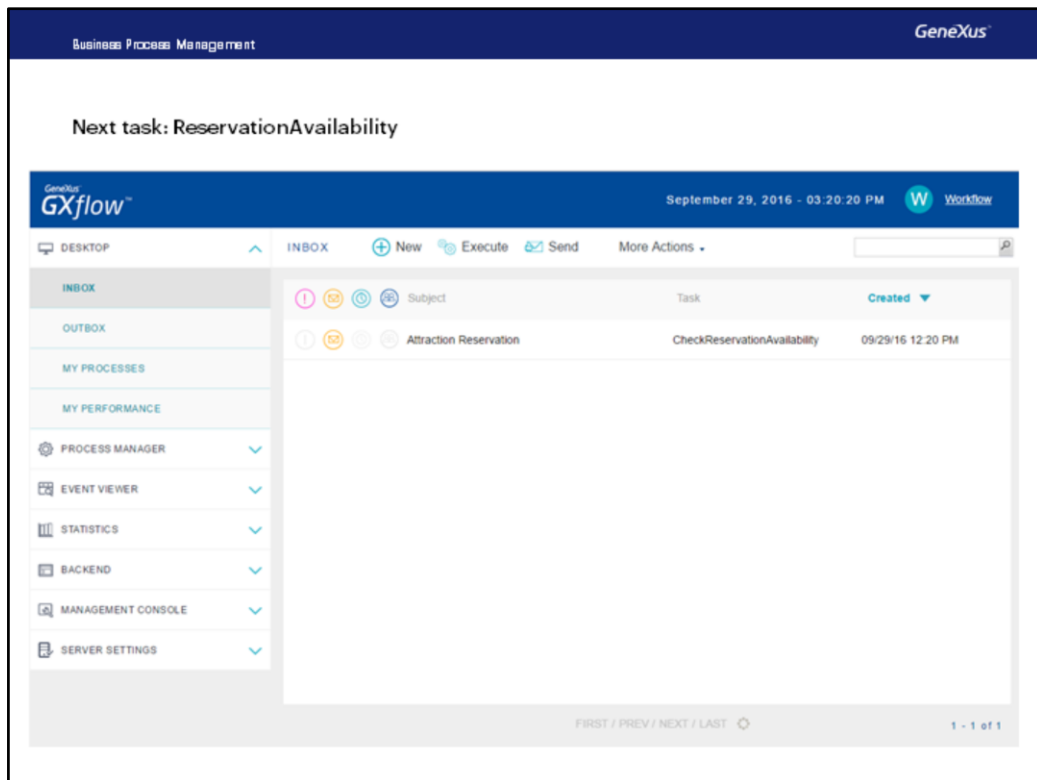
1 - 1 of 1

The task we have pending execution is the CustomerRegistration, as we did not enter customer ID in the reservation, so the diagram's flow will continue to the right, opening the Customer transaction so we can add the individual as a new customer.



We execute the task, the Customer transaction screen opens up, and we enter the customer.

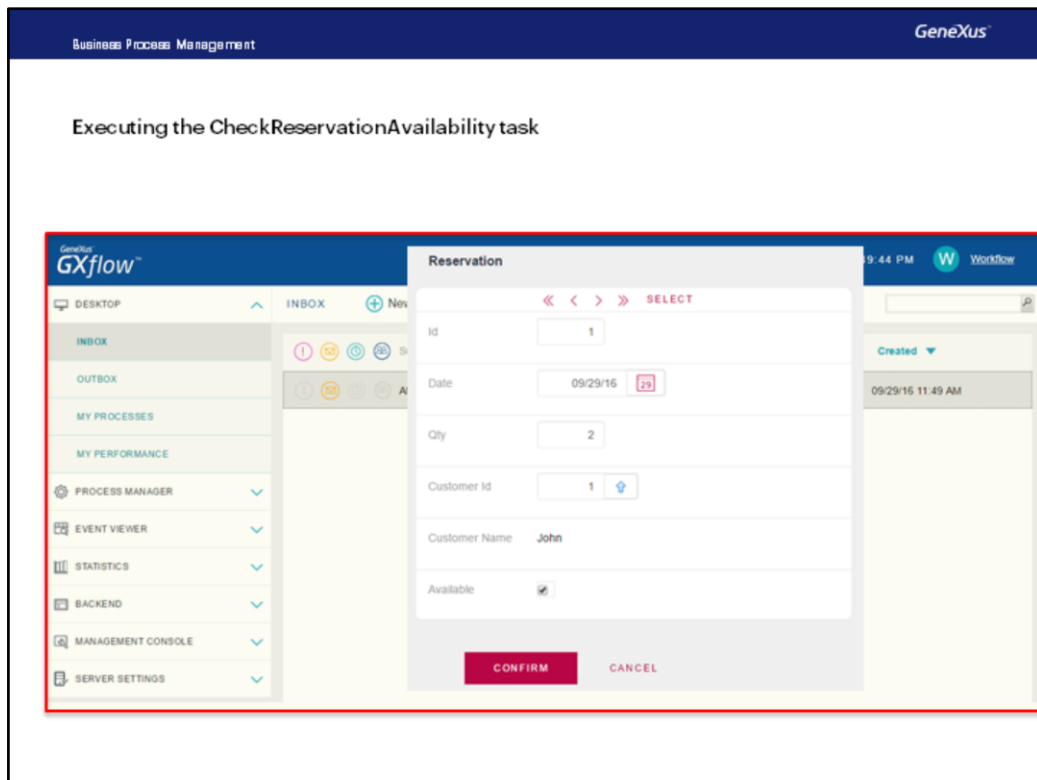
We confirm and then close the window... We now press the Send button to end that task and execute the following one.



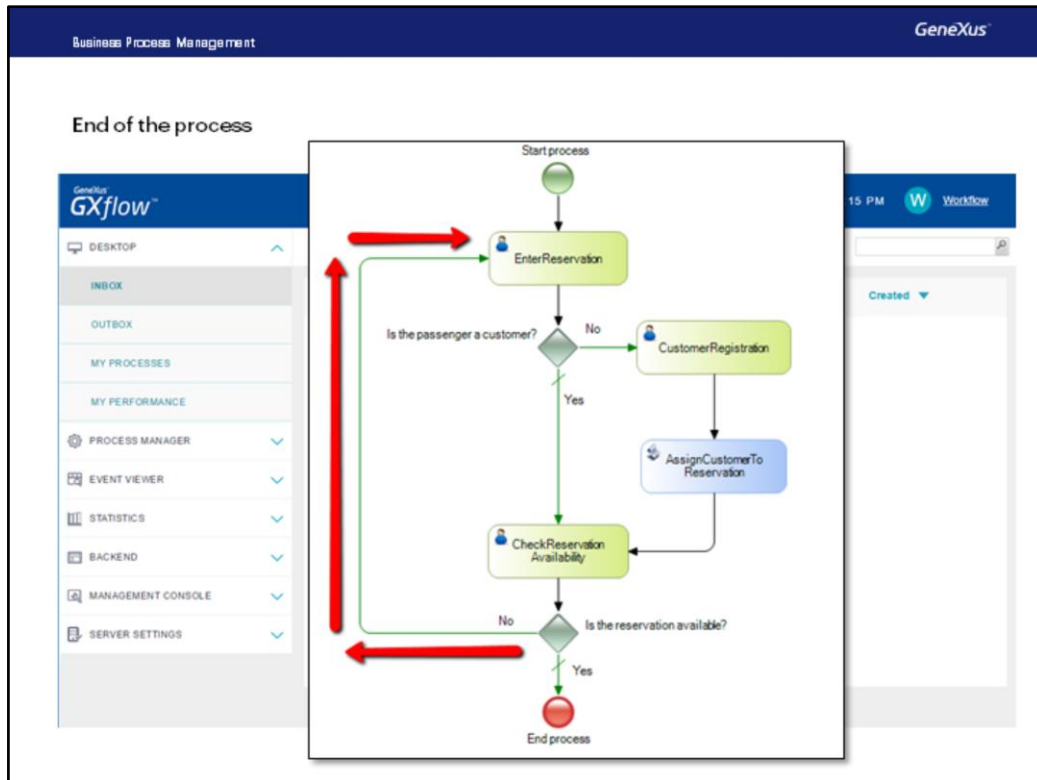
We can see that the following task is the one called CheckReservationAvailability.

Recalling the process, the Customer task invoked a task called AssignCustomerToReservation, which invoked a procedure to assign the customer identifier that was just entered in the reservation.

Since the AssignCustomerToReservation task is not interactive, it did not appear in the inbox when we ended the Customer task. Instead, we saw the following task to do, that is: ReservationAvailability.

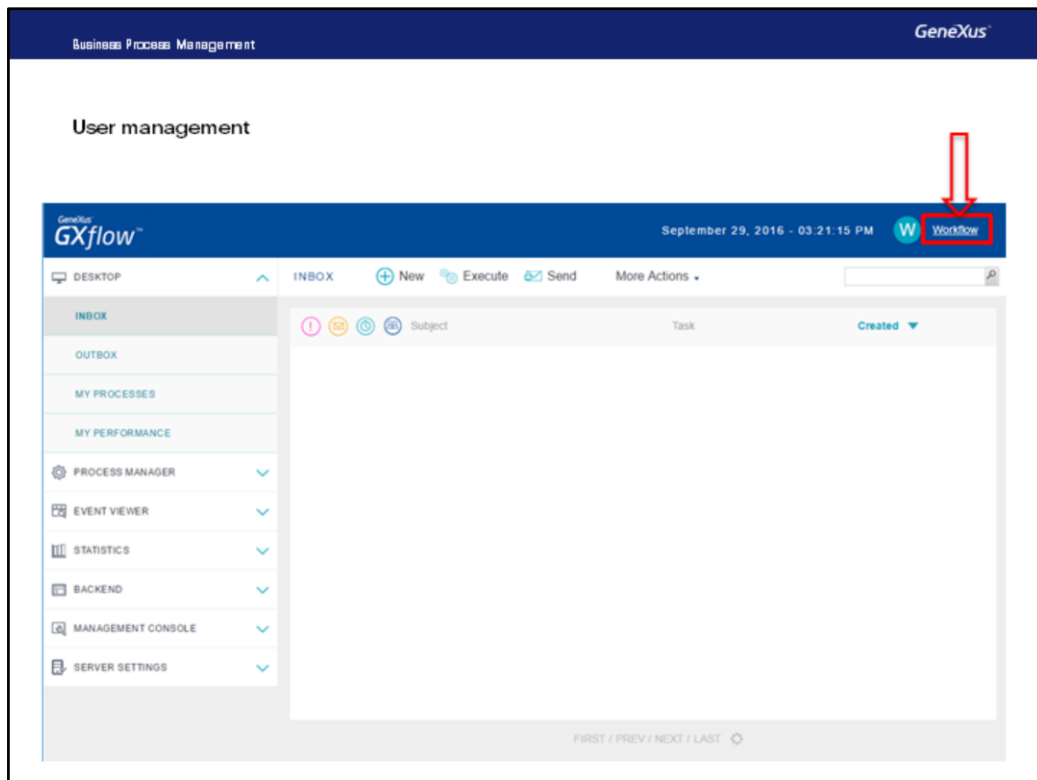


If we execute the CheckReservationAvailability task, by double-clicking on it, we can see that the form for the Reservation transaction is displayed, for us to enter whether or not the reservation is available. Let's suppose it is, so we select the checkbox and press Confirm.



We now see the inbox is empty. This means that there are no more tasks pending execution and we are at the end of the process.

Note that if we hadn't selected the checkbox, the workflow would have taken us again to the Reservation transaction, at the beginning of the diagram.



Something we did not mention is that all these tasks were executed by the same user.

This is so while we develop and test the process, but in reality, different users will be using it and will execute the different tasks, according to their profiles in the organization.

Business Process Management GeneXus

### Viewing process history

GeneXus GXflow™ September 29, 2016 - 03:23:26 PM Workflow

DESKTOP MY PROCESSES History Comments

INBOX

OUTBOX

**MY PROCESSES**

MY PERFORMANCE

PROCESS MANAGER

EVENT VIEWER

STATISTICS

BACKEND

MANAGEMENT CONSOLE

SERVER SETTINGS

	Subject	Process	State	Created	Ended
1	Attraction Res...	AttractionRese...	completed	09/29/16 12:08 PM	09/29/16 12:21 PM
2	Attraction Rese...	AttractionRese...	aborted	09/29/16 11:46 AM	09/29/16 12:08 PM
3	Attraction Rese...	AttractionRese...	aborted	09/29/16 11:13 AM	09/29/16 11:46 AM

FIRST / PREV / NEXT / LAST 1 - 3 of 3

Another interesting thing is that we can see the process history, and which pathways in the diagram were used during the execution of the process.

To see this history log, we select My Processes in the Navigator window. We are shown the process we have just executed (Attraction Reservation), which appears in Completed status.

Business Process Management

GeneXus

Viewing process history

GeneXus  
GXflow™

September 29, 2016 - 03:25:30 PM

Workflow

DESKTOP

INBOX

OUTBOX

MY PROCESSES

MY PERFORMANCE

PROCESS ANALYTICS

EVENT VIEWER

STATISTICS

BACKEND

MANAGEMENT

SERVER SETTINGS

HISTORY

Comments Consult Conditionals Optionals More Actions

Step	State	User	Created	Ended
Start process	completed		09/29/16 12:08 PM	09/29/16 12:08 PM
EnterReservation	completed	Workflow Administrator	09/29/16 12:08 PM	09/29/16 12:08 PM
Is the passenger a customer?	completed		09/29/16 12:08 PM	09/29/16 12:08 PM
CustomerRegistration	completed	Workflow Administrator	09/29/16 12:09 PM	09/29/16 12:20 PM
AssignCustomerToReservation	completed	Workflow Engine	09/29/16 12:20 PM	09/29/16 12:20 PM
CheckReservationAvailability	completed	Workflow Administrator	09/29/16 12:20 PM	09/29/16 12:21 PM
Is the reservation available?	completed		09/29/16 12:21 PM	09/29/16 12:21 PM

FIRST / PREV / NEXT / LAST

1 - 8 of 8

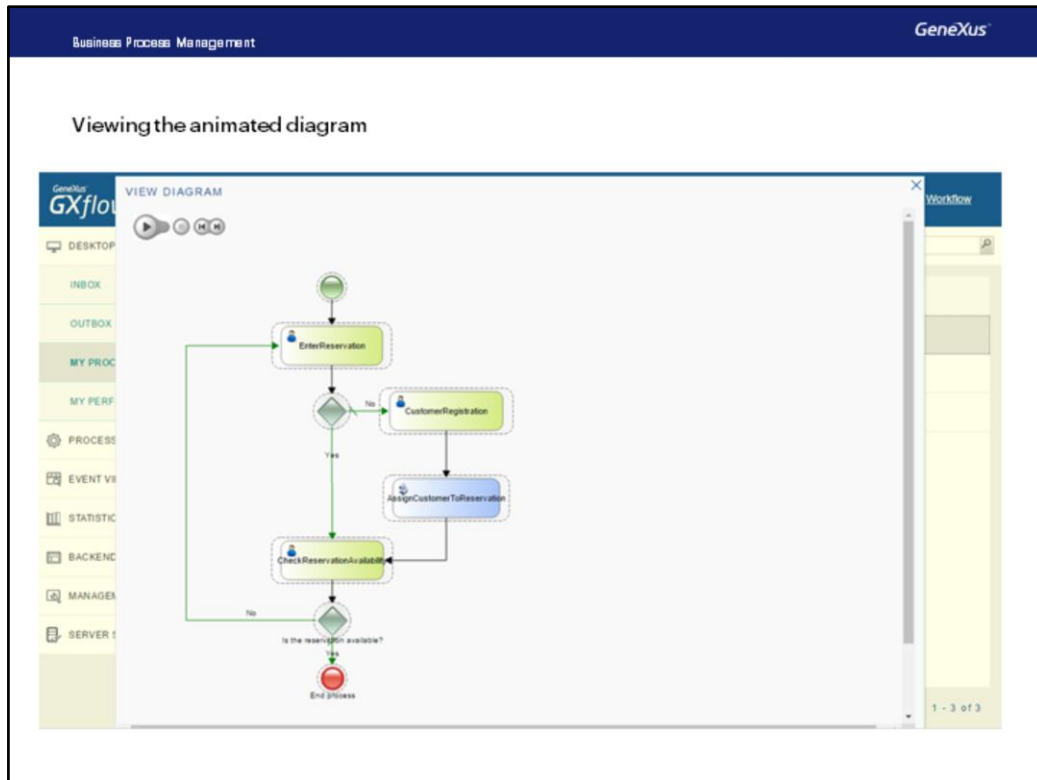
FIRST / PREV / NEXT / LAST

1 - 3 of 3

We double-click on it, and see that a window opens showing us the process history.

In this window we see all the tasks that have been executed.

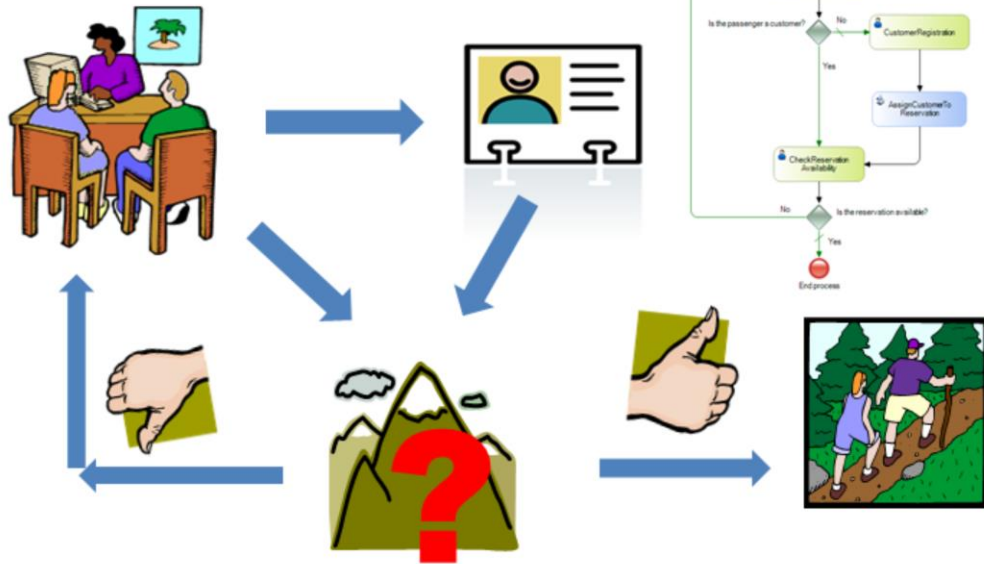




If we go to More Actions, View Diagram, we can see the animated history.

We press Play ...

## Business Process Management (BPM)



We have therefore seen that GeneXus enables us to model business processes in an intuitive manner, automatically solving the object that must be executed in each case, according to the definition we made through the diagram.

# GeneXus™

**The power of doing.**

Videos

Documentation

Certifications

[training.genexus.com](http://training.genexus.com)

[wiki.genexus.com](http://wiki.genexus.com)

[training.genexus.com/certifications](http://training.genexus.com/certifications)