## Turning the business process diagram into a functional application

In previous videos we have modeled the air ticket reservation process for a Travel Agency by means of a business process diagram, and we also saw a brief introduction to the BPMN standard, as well as the ways in which GeneXus facilitates the building of diagrams compliant with such standard.



Here, we will turn the model into a functional application using GeneXus objects associated with the symbols in our diagram.

Let's start by the process diagram we defined for ticket reservations.

The first TicketReservation activity is interactive and its function is to record the ticket reservation.

Page

To accomplish this we could use a transaction object with the following structure:

**GeneXus**<sup>™</sup>

age.

Reservation transaction object

	win Form   Kul	es   Events   Variables   He	lp   Documenta	tion   Pattern
Name	Туре	Description	Formula	Nullable
Reservation	Reservation	Reservation		
- P ReservationId	Id	Reservation Id		No
	Date	Reservation Date		No
····  ReservationQty	Numeric(4.0)	Reservation Qty		No
- 🛃 CustomerId	Id	Customer Id		Yes
CustomerName	Name	Customer Name		
ReservationAvailable	Boolean	Reservation Available		No

Here we see that the ReservationId attribute is the transaction's identifier, and also, we have the ReservationAvailable attribute to indicate whether the reservation is available or not.

In order for the screen of the Reservation transaction to be opened when the TicketReservation task is executed, we will associate the task to the transaction object. We will be performing these assignments from a GeneXus KB, so we will need to import the diagrams we created with the Business Process Modeler.

We execute the Business Process Modeler, then open the GXBPMSuiteCourse KB, select Knowledge Manager and there we select Export. We check Export All and then Export.

GXBPMSuiteCourse - GeneXus Business Process Modeler 15	- 0	×
: <u>File Edit View K</u> nowledge Manager <u>Wi</u> ndow <u>Tools</u> <u>H</u> elp		
🚳 🖸 Start Page X 🔝 FlightTicketReservation [Read-only] X 🎲 Export Objects X 🗸 🖉 Properties		7 ×
P     Export Objects		$\sim$
e Export Ele Name:		
2 C:\Models\GXBPMSuteCourse\GXBPMSuteCoursexpz Export		
Options Cancel		
Add Knowledge Base/Version/Environment properties to list		
⊠ Egpot Al		
Add		
References		
Qear		
Categories Toolbox		
Image: Second		
C/Models/GXBPMSuiteCourse		

We now execute GeneXus and open the GXBMPSuiteCourse KB that we had created previously, where we had created several objects.

We select Knowledge Manager and then Import. We search for the file we had created from the Business Process Modeler and press Load.

👰 🐼 Start Page 🗙 🌏 Import Objects 🗙				*	Properties		<del>7</del> ×
Disses type the same of a file to import or releast it:					📜 🦺 🌾   Filte	er	×
C:\Models\GXBPMSuiteCourse\GXBPMSuiteCourse xpz			💕 🗠	bad	<ul> <li>Options: Impo</li> </ul>	rt Options	
HI					Import Type	All O	bjects
Objects in File		Import Status			Automatic Bac	kup True	
> ☑ 🛃 Business Process Diagram (4)	Import >>				Language Tran	slatio Upda	te ~
	Cancel				Redefine Extern	nal Pre True	
	001001				Show Propertie	es Errc False	
	Options				Allow Create P	arent True	
					Allow Create N	1odul True	
					∨ Themes		
					Theme Impo	ort Be Incre	mental Integration
					Class Confli	cts Use F	rom Export
Select / deselect all			Clear Sta	tus			
🐼 0 Errors 🛕 0 Warnings 😒 0 Success							
Output				ΦX			
Snow : Import/Export			<b>◆</b> Au	loscroii			
Processing file 'C:\Models\GXBPMSuiteCours	==== e\GXBPMSuiteCourse	e.xpz'		^			
Checking Import ReferencesFinished							
Reading import file objectsFinished							
Import File Load Success							
				~	Properties	Toolbox	Responsive Sizes
C:\Models\GXBPMSuiteCourse m://C:\Models\GXBPMSuiteCourse	/				operades	3/0/0	/3 4/4 .

In this case we are using an export file with extension .xpz to take diagrams from the GXBPM to GeneXus. Another possibility could have been to send, from the GXBPM to the KB, to a GXServer and then do a Create KB from Server from GeneXus.



Now we open the FlightTicketReservation object with Ctrl-O.

We select the task TicketReservation and go to its properties. We click on the Web Object property and press the button with three dots.

In the dialog box we click on the button reading "Web Object" and select the Reservation transaction prior to pressing Ok.

Fi	le Edit View Layout Insert Buil	d Knowledge Manager Window Tool	s Test Help				
•	三日 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	🚦 🛗 🕍 🕨 🕨 🕼 🖬	- Release - 📮 Tests Bu	ild E	Build - 🔬 🖡		
Ħ	🐼 Start Page 🗙 🔝 Validate	Reservation 🗙 🚠 FlightTicketReservati	onDynForms 🗙 🔚 Flight < >	$\overline{}$	Properties	ą	×
Se la constante de la constant	Diagram * Relevant Data Doc	cumentation			📜 🛃 🌾 Filter		$\times$
Т. Бр				✓ Task: Ticket Reser	vation	^	
orer	New reservation	2			Name	Ticket Reservation	
	request	Ticket Reservation			Task Metadata Id	13	
					Task Metadata GUI	b9d37456-023c-49b0-b1f2-	·
	Application Declaration X			×	Туре	User	
	c c	Application: Reservation			Web Object	Reservation .	
				. 1	Object	(none)	
		Data Mappings	Delevent dete		Visible in history	True	
		ReservationId	(none)		Roles		
					Subject rule		
					✓ Advanced Prope	erties	
					Optional routing	Branch	
			OK Cancel		Request commer	False	
					Work with docum	False	
					e 1 1	r 1	

We can see that the Data Mapping sector shows that the transaction's primary key is the ReservationId attribute. We then run the mouse over the RelevantData column, where we will see a button and an arrow.

We press the button to the left of the arrow and we will see that ReservationId has been selected.

Application Declaration	×
Application: Reservation	
Data Mappings	
Primary key attribute	Relevant data
ReservationId	(none)
	(none)
	ReservationId
	Customerld
	OK Cancel

This means that ReservationId will be the name of relevant data in our process.

Relevant data is global data that is known throughout a process. We can compare it to a global variable for it is the mechanism used by Workflow for passing information between tasks.

When we associate a transaction to an interactive task GeneXus automatically creates relevant data with the same name as the transaction's identifier attribute.

We press OK, and we will have established a relation between the task and the GeneXus object, in addition to a mapping between the ReservationId attribute and the &ReservationId relevant data.

2 Filter	Ť
Task: Ticket Reserv	vation
Name	Ticket Reservation
Task Metadata Id	13
Task Metadata GUIE	b9d37456-023c-49b0-b1f2
Туре	User
Web Object	Reservation
Object	(none)
and the second	

This means that, every time that the TicketReservation task is instanced, the Reservation transaction will be executed because it is the application associated with the task.

In this modeling process we first created the interactive task and then associated it to the application. Instead of creating the task, we could have dragged the diagram from the View Folder directly to the Reservation transaction.



When we do this we can see that an interactive task is automatically created with the same name as the transaction, and its Web Object property reads: Reservation, because the GeneXus object was automatically associated with the task.

If we press the button with the three dots we will also see that the relevant data was created and that the primary key attribute in the transaction was automatically associated with it.

Start Page 🗙 <u>ह</u> Validate Reservation	on 🗙 🔝 FlightTicketReservationDynFor	ms 🗙 🔝 FlightTicketRese < >	$\overline{}$	Properties	<del>7</del>	×
i <b>gram *</b> Relevant Data   Documentat	ion			📜 🛃 🌾 Filter		$\times$
			^	✓ Task: Reservation		^
New reservation				Name	Reservation	
request				Task Metadata Id	67	
				Task Metadata GUI	db0f70b4-f7c6-4265-86bd	
	No 🔒			Туре	User	
Is the passenge customer?	er a Record	Customer		Web Object	Reservation .	
	Ĭ			Object	(none)	
	Application Declaration	>	<	Visible in history	True	
				Roles		
9	Application: Reservation			Subject rule		
Reservation	Data Mappings	1		✓ Advanced Prope	erties	
	Primary key attribute ReservationId	Relevant data ReservationId		Optional routing	Branch	
				Request commer	False	
				Work with docun	r False	
				Send e-mail	False	
		OK Cancel	11	Strong synchroni	False	
	Validate Reservation			Skip predecessor	True	-

This relevant data will have the same name and type as the transaction's primary key.

If the primary key includes more than one attribute then relevant data will be created for each attribute that is part of the primary key.

In the Relevant Data tab of the FlightTicketReservation process we can see the relevant data defined.

<b>1</b>	🐼 Start Page 🗙 👪 FlightTic	ketReservation * ×		- 1	Properties		<del>7</del> )	×
BExp	Diagram * Relevant Data * D	Documentation		>	🛔 🛃 🌾 🛛 Filter			×
lorer	Name	Туре	Is parameter	~	RelevantData: Rese	ervationId		
†4T	🖃 🚷 Relevant Data				Name	ReservationId		
	<ul> <li>ReservationId</li> </ul>	Numeric(6.0)		۰.	Туре	Numeric		
					ls parameter	True		$\sim$
					Parameter access	In/Out		
					Length	6		
					Value			
					ls constant	False		
					Dimension	Scalar		
					✓ Event Handling			
					On data change			

This data appears under the ampersand symbol to indicate that it consists of variables.

When we refer to the relevant data ReservationId we always do it by using the ampersand symbol as prefix in order to set it apart from the ReservationId attribute, as well as in the case of variables in GeneXus objects.

One more thing to bear in mind when we drag a transaction object to the diagram is whether is has a Parm rule defined where the value of the primary key is received in a variable.

Here we have the Country transaction to which the Work With pattern was applied and consequently a Parm rule was added to receive the &CountryId variable as parameter.



The dragging of the transaction to the diagram creates the relevant data, but if the names of variables in the Parm rule match those of the attributes in the primary key then there will be a mapping between the relevant data and the variables, so that **when the transaction is** executed, the variables take the values stored in the relevant data at the time.

In the next video, Automation Part 2, we will continue to associate GeneXus objects with the tasks in our ticket reservation diagram.