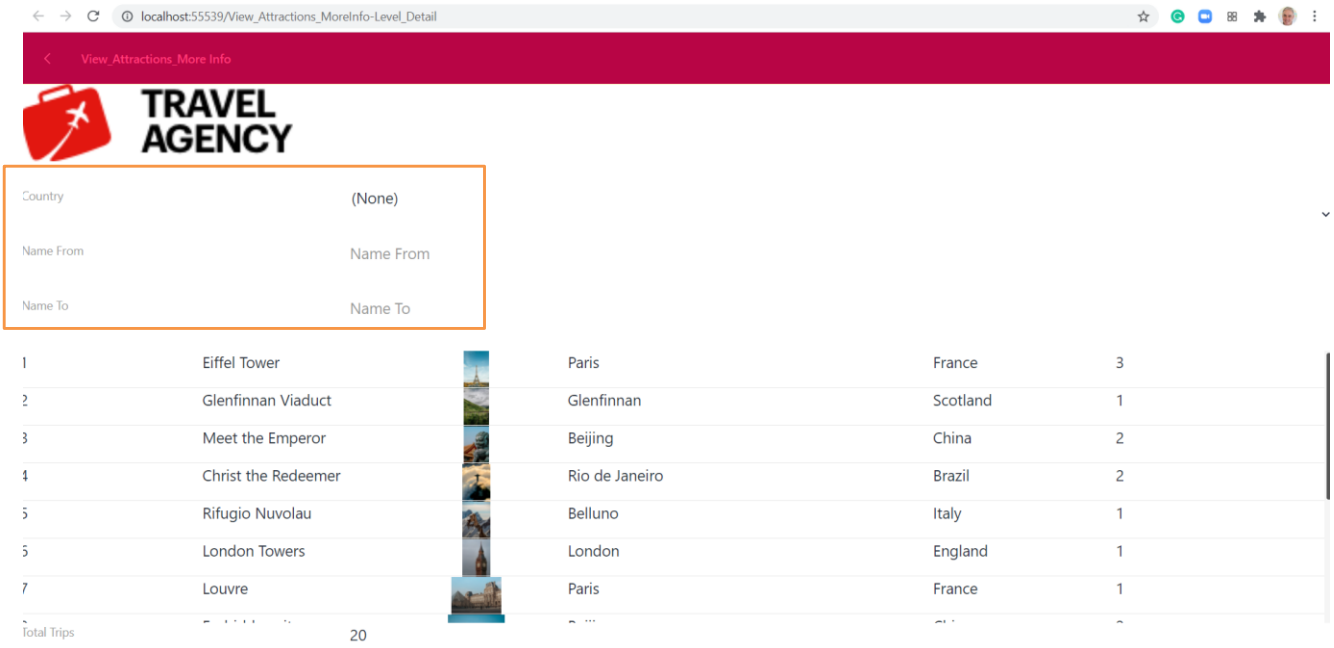
Tables, Tags, and Images

GeneXus™

In this video, we will see how to use tables to organize the information in the form, the options we have for using labels to help us understand the data, and how we can use images and change the way they are shown in the form.

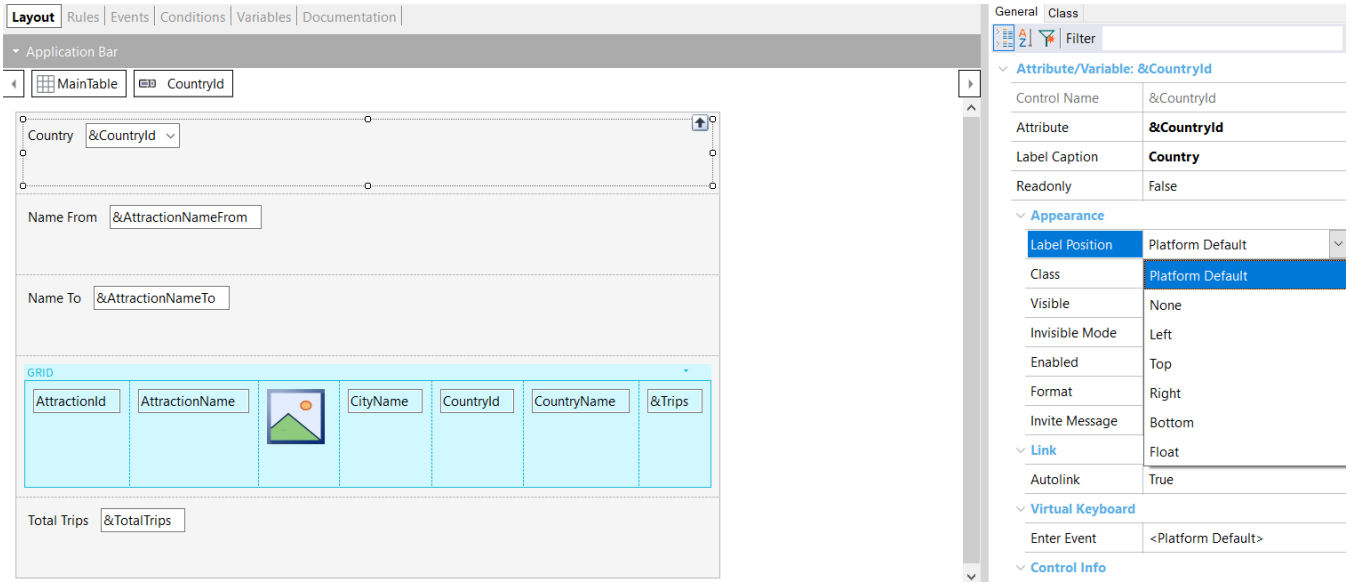
Tags

Let's start by customizing the appearance of the labels associated with the controls in our application.



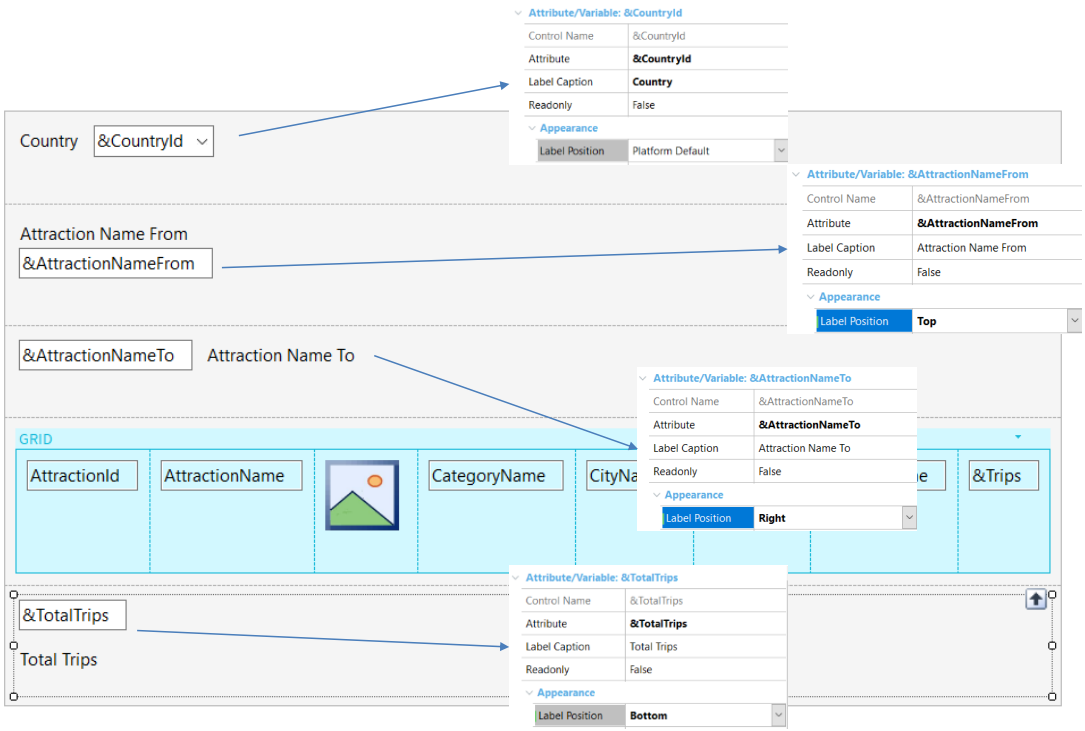
If we run the View_Attractions_MoreInfo panel, we see that the filter labels are displayed by default to the left of the field.

Possible label positions



The position of the label relative to the field can be changed through the Label Position property.
For example, if we select the &CountryId variable, the values of this property can be as follows:

- Platform Default: For Angular, it is the Left value; that is to say, the label will be on the left as we saw. For Android, it is the Top value (that is, above the content) and for iOS it is also to the left.
- None: If we select this value, the tag is not shown. We've already used this when we added the &Trips variable to the grid.
- Left: The tag will be displayed to the left of the content.
- Top: The tag will be displayed above the content.
- Right: The tag will be displayed to the right of the content.
- Bottom: The tag will be displayed below the content.
- Float: This value is only available for native mobile platforms and in this case the label is initially displayed in the position of the attribute or variable content. When the user starts typing, the label moves upwards as if it floated. This only works in editable fields and in numeric or character type fields; in the rest of the types, the value Top is adopted.

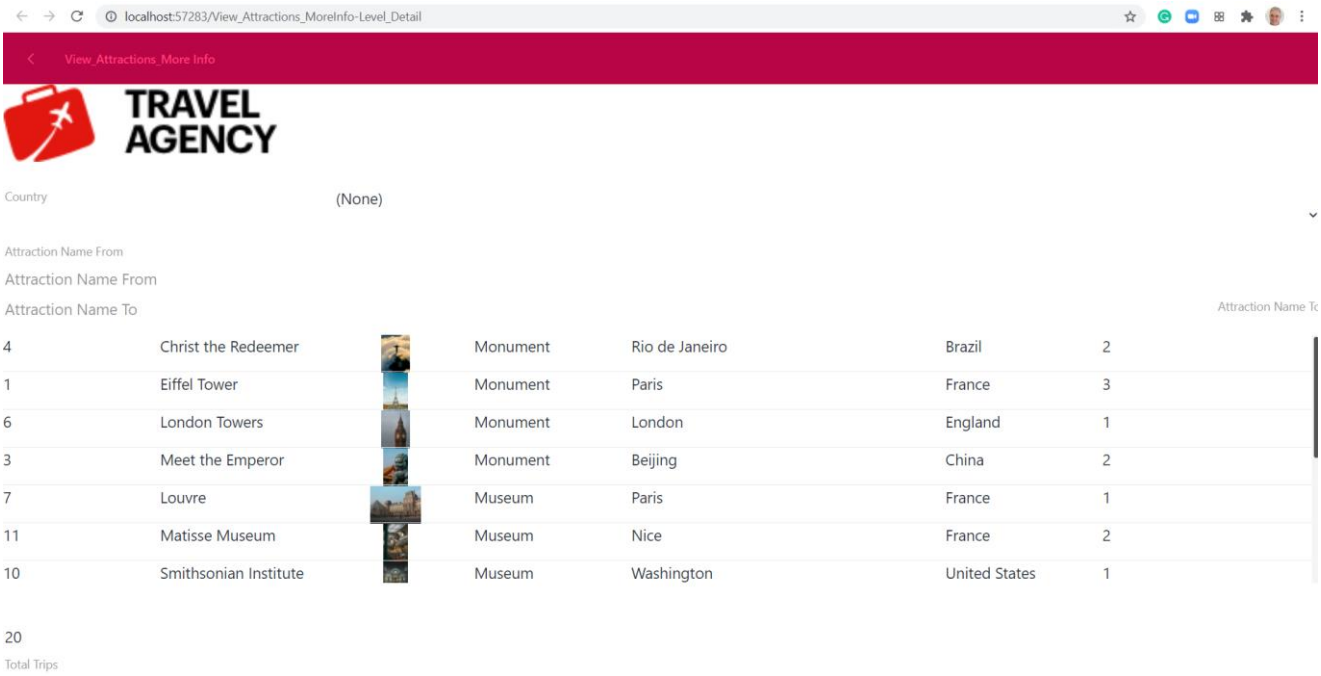


We are going to change the label values of the variables we have on screen.
We leave the Country label of the filter by country with the value PlatformDefault, so it will appear on the left as usual.
To the AttractionNameFrom tag we set the value Top, to the AttractionNameTo tag we set the value Right, and to the total number of trips we set the value Bottom.

Already at design time we can see the changes in the form of the panel.

We run it...

Testing label positions



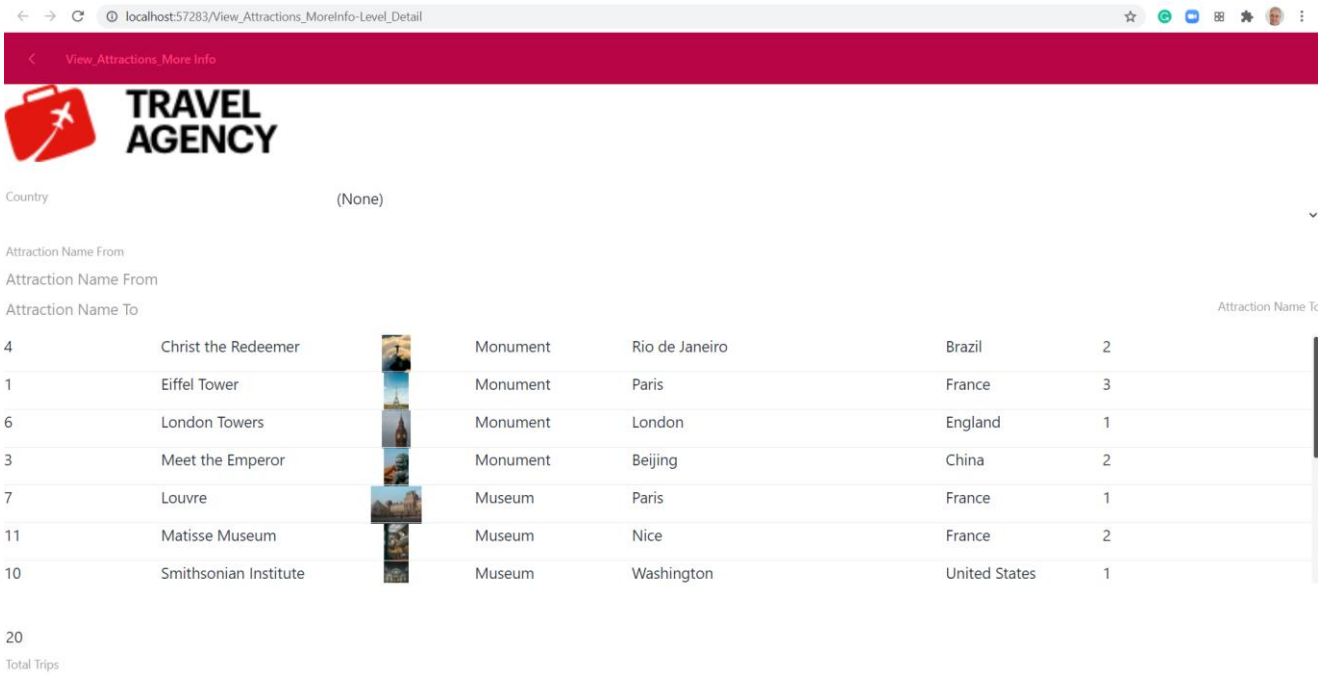
We see how the word Country is displayed on the left, because it is the platform default value; the label Attraction Name From is above the field Attraction Name From because we set it to Top; Attraction Name To is to the right of the field Attraction Name To because we set it to Right. As it had the entire screen width available, it is well aligned to the right, and the label of the total trips value is below the total, because we set the value Bottom.

Depending on the application and the information we want to display, we will choose one value or another for this property.

This property can also be set if we want to add titles to the values shown in the grid, but in this case each grid cell will have its own label.

Tables

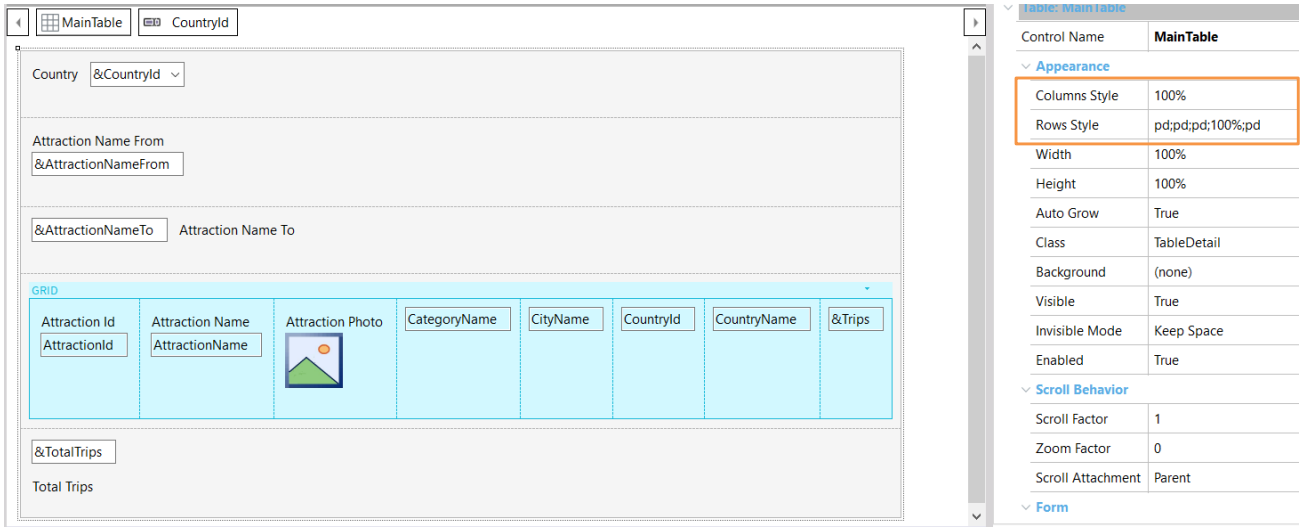
We have already used tables to improve the appearance of a panel in our application. Let’s go back to this topic to understand the use of this control a little better.



When running the View_Attractions_MoreInfo panel, we can see that the country filter selector is on the far right of the screen. If we press it, the Dynamic combo box takes the entire width of the screen.

We can also see that the label we set with right alignment was positioned on the right side of the screen, far away from the AttractionNameFrom field where we entered the filter values.

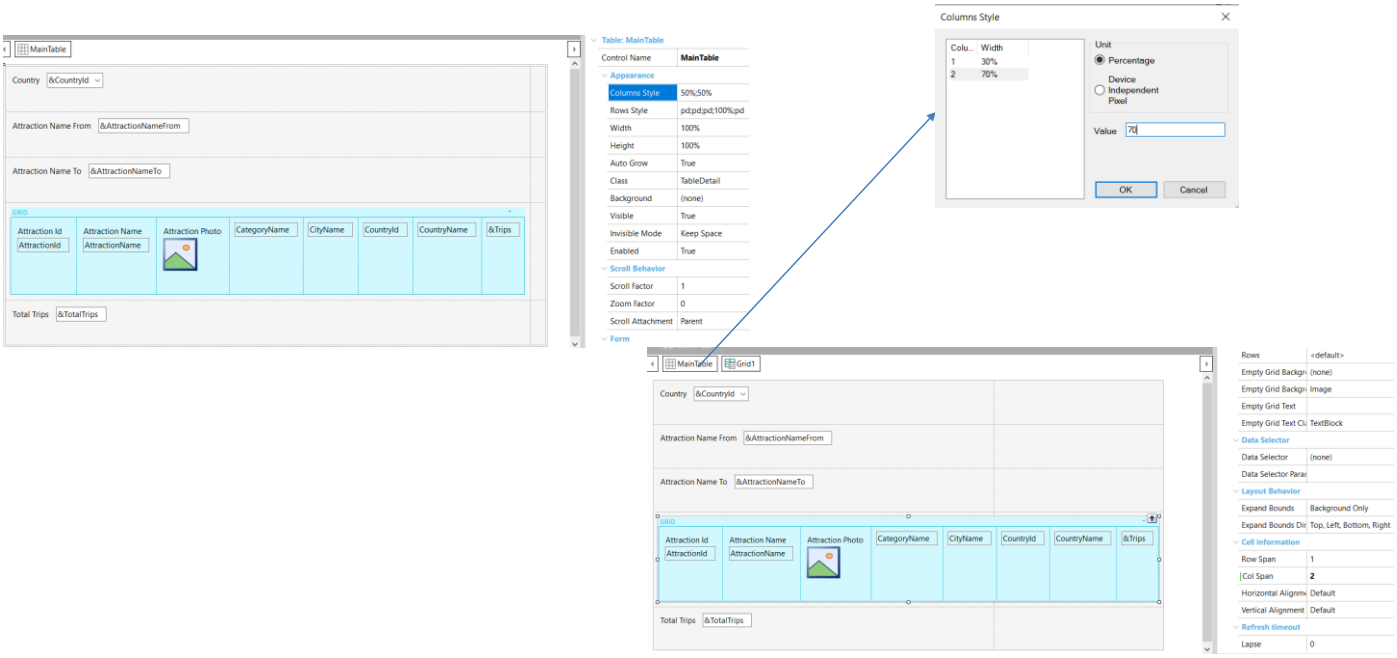
Columns Style and Rows Style properties of a table



If we click on any control in the panel form, we see that they all belong to a table called MainTable.

If we go to its properties, we see that the Column Style property has a single value of 100%, which implies that the table has a single column.
The Row Style property has 5 values, corresponding to the 5 rows of the table, all with platform default values except for the fourth row that corresponds to the grid.

Adjusting the columns of a table

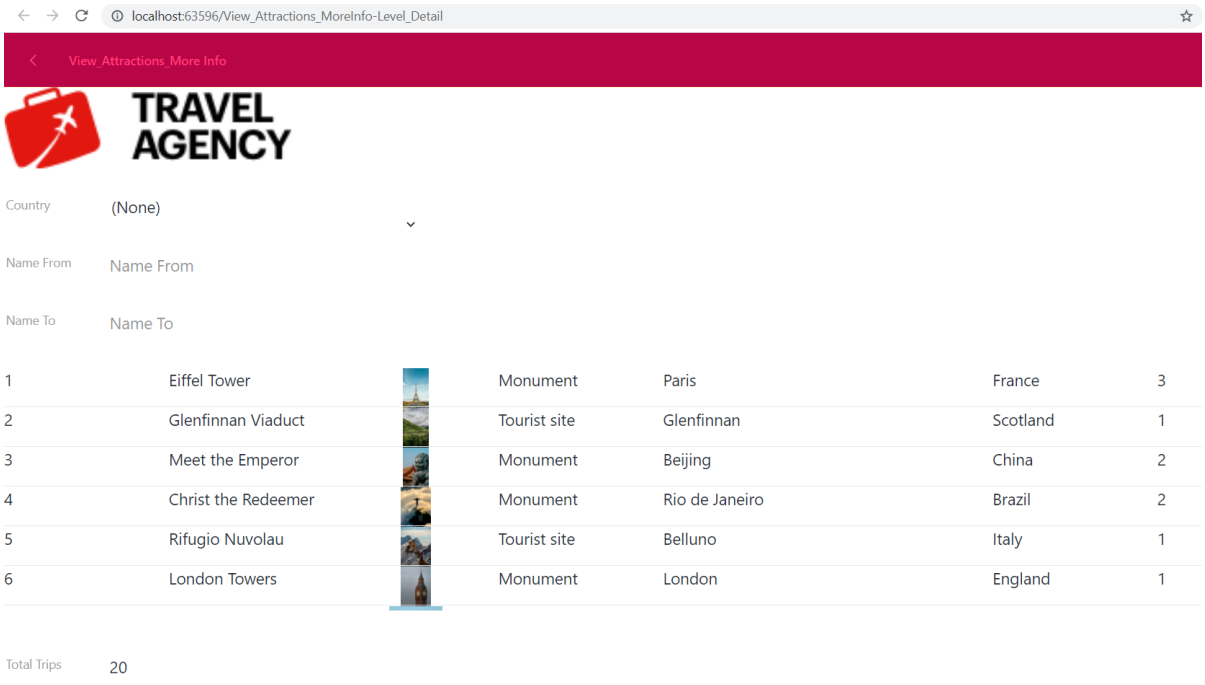


Let's improve this by adding a column to the Main Table.
We select the MainTable, right-click, and select Insert Column After. Note that the new column has been placed on the right.

In the Main Table properties, we confirm that it now has 2 columns, and that each one takes up 50% of the available space.
We can also see that the grid is in the left column but we want it to take up both columns, because the table we added should only affect the controls in the fixed part.

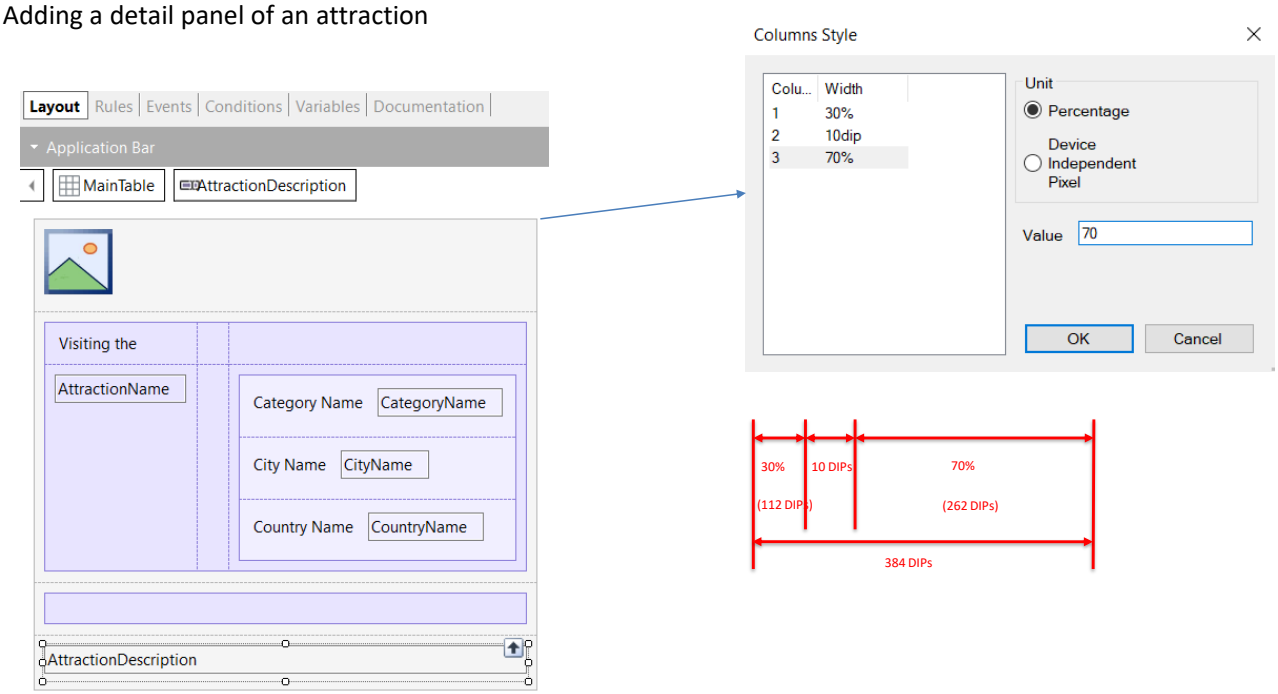
To make the grid take both columns, we go to its properties and in the Cell information group*, in the Col span property, we set the value 2; the grid now takes the entire width of the form.
We go back to the table properties and adjust the value of the columns to 30% and 70%.
This is the opportunity to leave the default position values in the labels, adjust their descriptions, and execute.

* Note: if you are developing with any GeneXus version from v17 U11 up to v18 U4 (both included), the Cell information property group will not be available in Grids. In that case, create a Table, place the Grid inside it, and set the Col Span property of that Table to achieve the same result.



It looks much better and if we click on the Country combo box, we can see that now it has a reasonable width.

Adding a detail panel of an attraction



As we continue developing the application, we create a panel to see the details of an attraction and name it View_Attraction_Details. In the rules, we add a Parm rule with the AttractionId attribute as input.

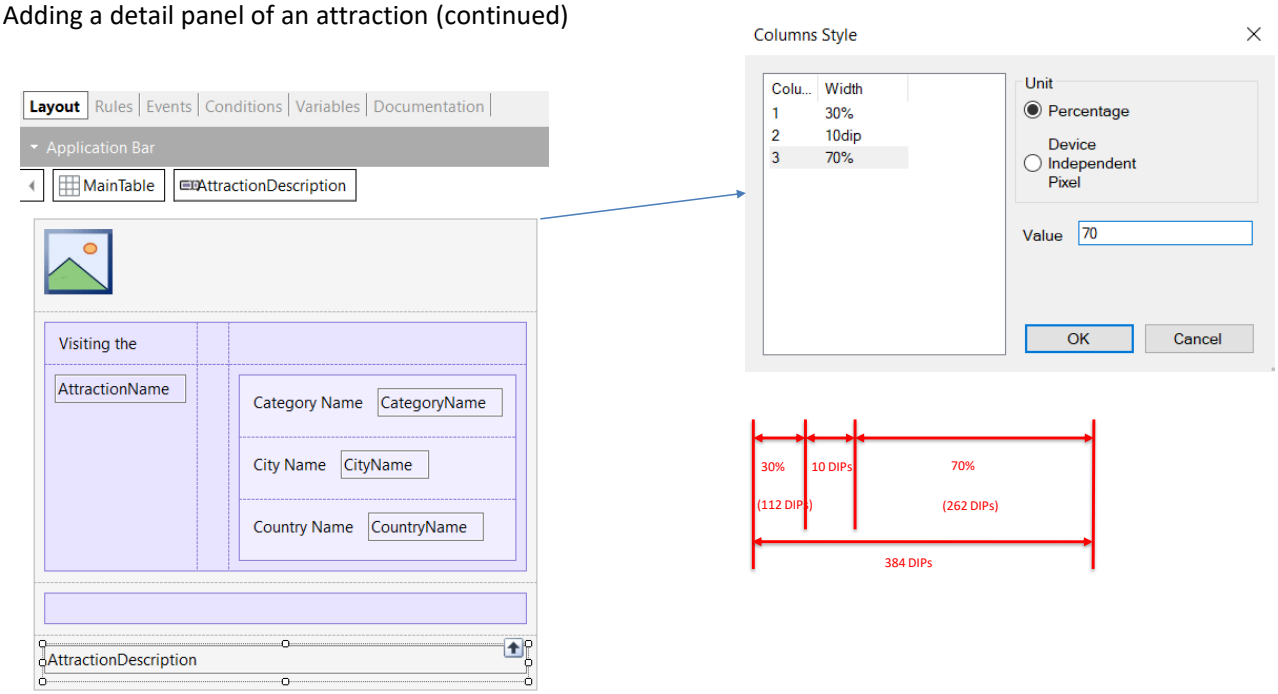
Then in the form, we drag the AttractionPhoto attribute and set its Label position property to None.

Below the photo we add a table; inside it we add a textblock, and in its Caption property we type the text “Visiting the.”
Below it in the table we add the attribute AttractionName.
We select both controls and in the Horizontal Alignment property we choose Left.

We right-click on the textblock and select Insert Column After twice to create 2 columns.
Now we drag a table and insert it well to the right in the AttractionName row.
Inside this table we include the attributes CategoryName, CountryName, and CountryCity.

We are going to adjust the properties of the tables' columns and rows to display the contents properly.
We select the table below the photo and in its Column Styles property we choose 30%, 10 DIPs (**D**evice **I**ndependent **P**ixel) and 70%.

Adding a detail panel of an attraction (continued)



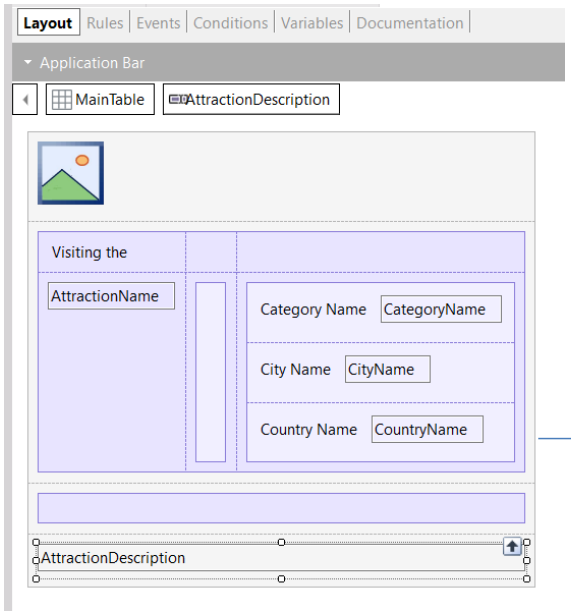
The Device Independent Pixel unit corresponds to the abstraction of 1 pixel; later on, an application converts it to physical pixels, which allow scaling to different screen sizes. Each platform has a different number of pixels; for example, in a web application generated in Angular, 1 DIP is 1 pixel, and in native mobile applications it depends on the device.

The maximum width of the columns of a grid is always 384 DIPs.
Percentages are relative to the value resulting from subtracting the fixed values from the total width (in DIPs).

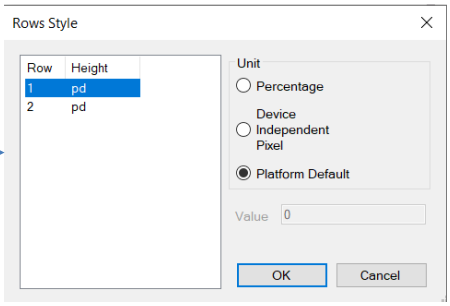
In this example, there are three columns: the first with 30%, the second with a fixed value of 10 DIPs and the third with 70%. The values that the first and third columns will take are obtained by applying those percentages to the value resulting from subtracting the sum of the fixed values (here only one, 10 DIPs) from the width of the table. The second column with 10 DIPs is used as a blank space between the first and the third column.

For this reason, the space available for the first and third columns is: 384 – 10 = 374 DIPs; therefore, the first column will have 30% of 374 which equals 112 DIPs and the third the remaining 70%, which are 262 DIPs.

Adjusting the rows of a table



PD: Using the best value depending on the platform and context				
PD in pixels	Label Position=Top		Label Position=Left	
	Mode=View	Mode=Edit	Mode=View	Mode=Edit
Angular	64	64	44	44
Android	64	64	44	50
iOS	53	53	44	53



For the rows, we leave the PD (Platform Default) values which correspond to: “Using the best value depending on the platform and context.”

This value varies with the platform and, for the same platform, it also depends on the cell’s content. For example, if the field has a label or not, and if it does, if the label will be displayed on top or to the left, depending on the value of the Label Position property, or if we are in a layout in Edit mode or in View mode (we will see this later).

For example, for Angular or Android, with Label Position = Top, it corresponds to 64 DIPs, while on iOS to 53 DIPs.

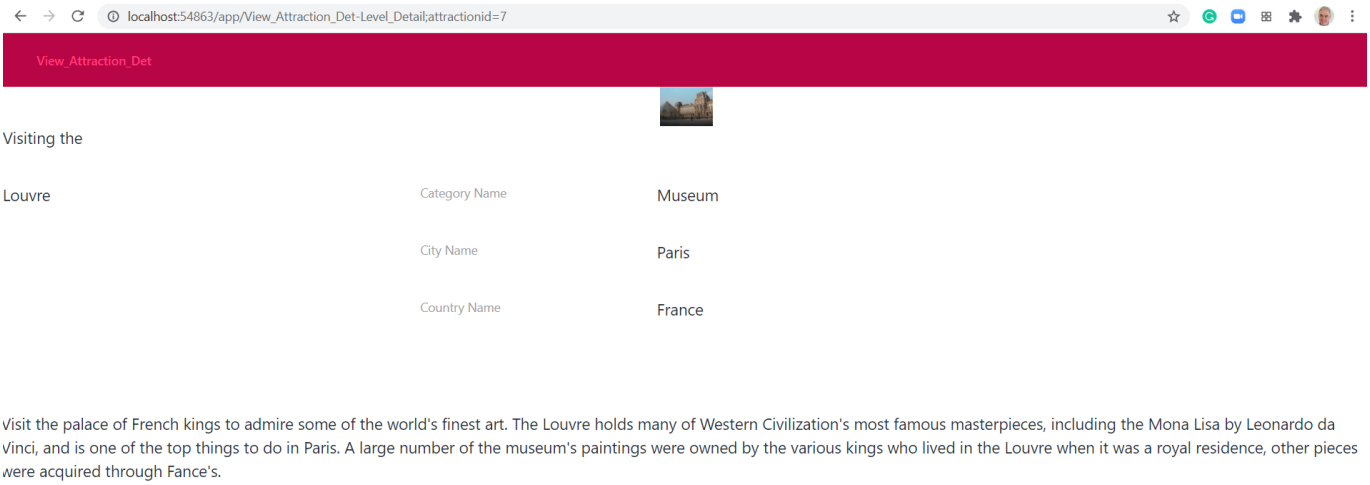
Remember that in Angular 1 DIP is 1 pixel, but in Android or iOS the number of pixels per DIP depends on the device and the orientation.

In order to invoke the panel we built, we go to the View_Attractions_MoreInfo panel and program the Tap event of the AttractionName attribute to invoke the View_Attraction_Detail panel, sending the AttractionId by parameter.

Images

Now let’s see how we can change the way an image is displayed.

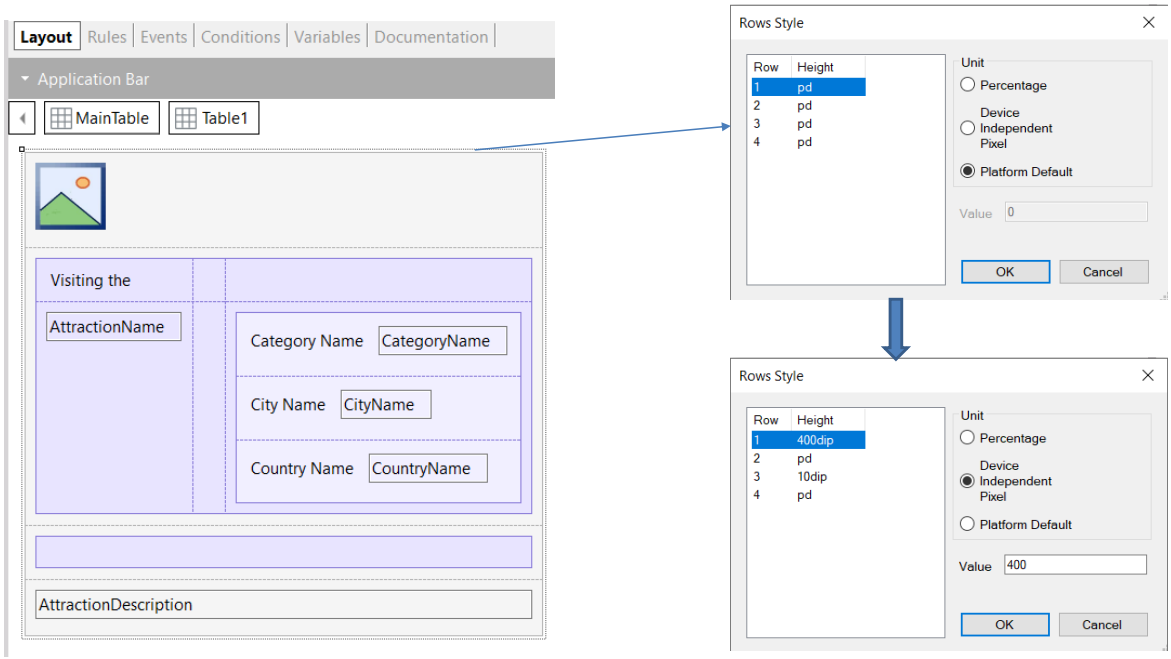
Image scaling



Before we continue, let's run it to see what we have done so far. We run the panel View_Attractions_MoreInfo...
We click on the Louvre and the panel we have just built opens... It doesn't look so bad, but the image is too small.

First of all, let's check the values of the Rows Style property of the Main Table.

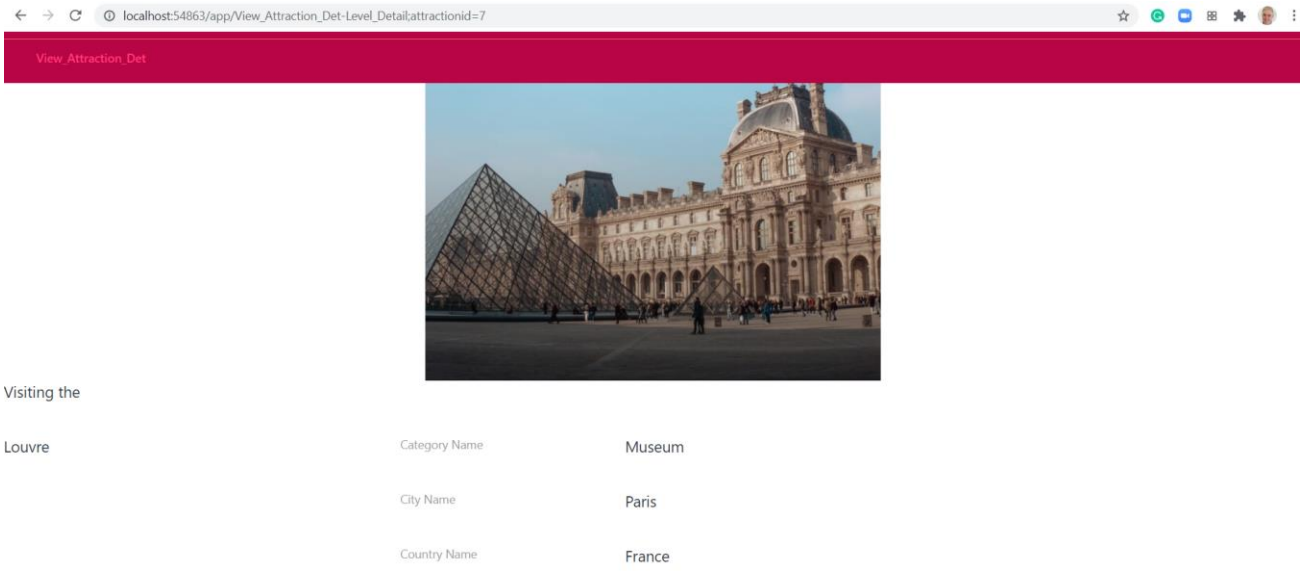
Image scaling



Note that for all the rows we leave the PD value. This means that if we generate in Angular, according to the table we saw, the image is only 64 pixels high, that's why it looks so small.

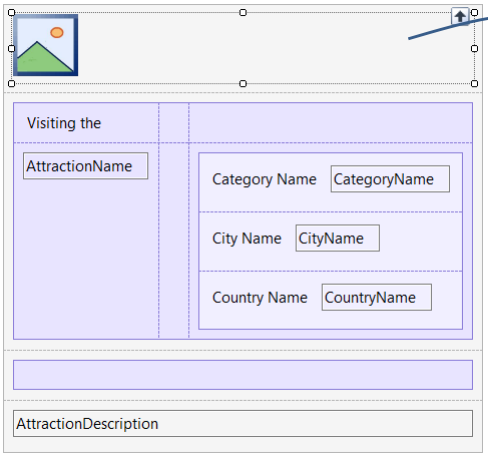
We give it the value 400 DIPs (which in Angular would correspond to 400 pixels) and to the third row we give the value 10 DIPs, so that it is a separator between the main data and the description of the attraction.

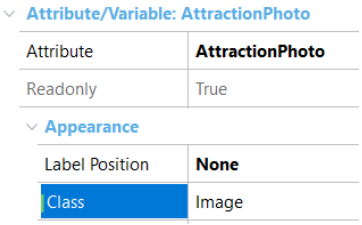
Let's try it.

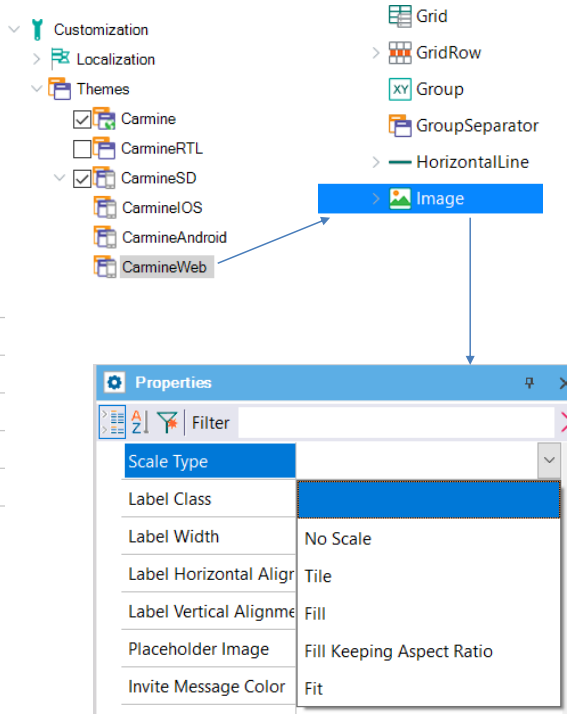


It looks much better now! The image was enlarged to fit the available space without losing proportions.

Image scaling

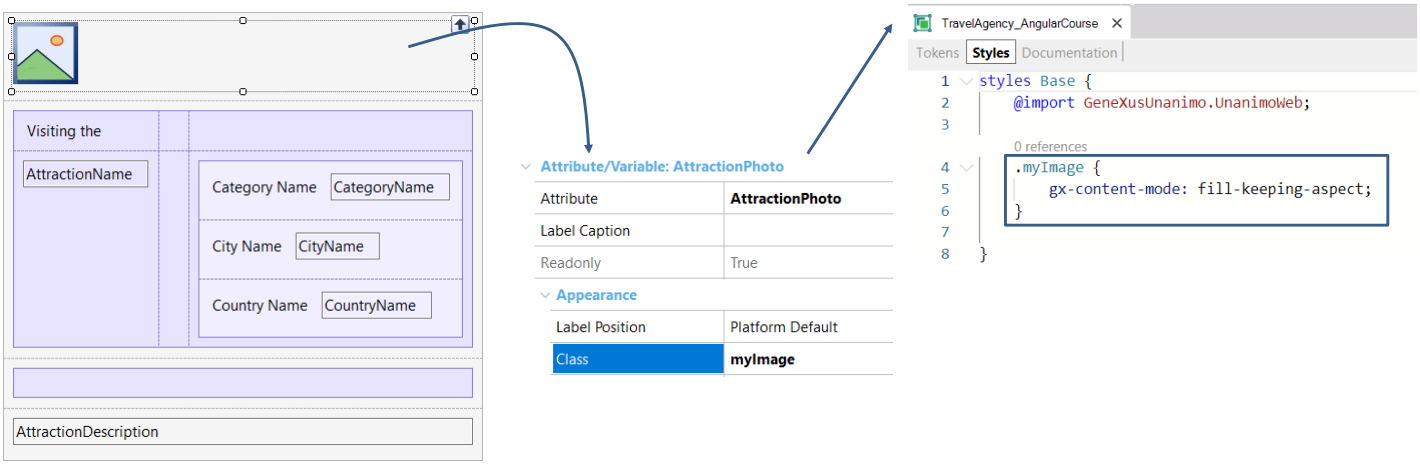






In GeneXus 17, it is possible to use the Scale Type property of the class associated with the Image control to configure how the image adjusts to the size of its container in the applications. To do so, we go to Customization, Themes, CarmineSD, and double-click on CarmineWeb; then we select the Image class to set the value in the Scale Type property.

Image scaling



To configure this in GeneXus 18, we will use in the Design System Object a property called *gx-content-mode*, applied to the control class. Later on, we'll talk about how to work with a Design System Object.

Possible values of the Image Scale Type property

No Scale	Keeps the original size of the image, regardless of the size of the control area.
Tile	The image is not scaled. It is repeated horizontally and vertically to fill the control size.
Fill	The image is scaled in width and height to fill the whole size of the control area.
Fill Keeping Aspect Ratio	The image's width and height are increased or decreased to fill the entire size of the control area, but keeping the aspect of the image. For example, if the image size is 100x200, and the control size is 50x50, then the image size is converted to 50x100.
Fit	The image scales in width and height to see it at all, keeping the aspect of the image. For example, if the image is 100x200, and the control is 50x50, then the image is converted to 25x50. This is the default value.

The values that define image scaling are as follows:

No Scale: The original size of the image will be kept regardless of the size of the area where the control is located.
Tile: In this case, the images are not scaled and the image will be repeated to fill the control area.
Fill: The image height and length will be scaled to take the area where the control is located.
Fill Keeping Aspect Ratio: It is similar to Fill but the image will always keep the same width and height proportions; in this way, the image will not be skewed. For example, if the image size is 100x200 px and the control area size is 50x50, the image will be converted to 50x100 px.
Fit: The image height and width will be scaled so that it can be fully displayed, keeping its proportions and aspect ratio. For example, if the image size is 100x200 px and the control area size is 50x50, the image will be converted to 25x50.

The default value is Fit, so in this example, without changing the property in the class, the image took the 400 DIPs (400 pixels) we assigned to it and was proportionally enlarged until it looked right.

In summary

Label Position	Tables	Image Scaling
• Platform Default • None • Right • Left • Top • Bottom • Float	• Columns Styles • Row Styles • Measure Units	• No Scale • Tile • Fill • Fill Keeping Aspect Ratio • Fit

To sum up what we have seen in this video:

- We saw the various options for Label Position: Platform Default, None, Right, Left, Top, Bottom, and Float. The last one is only valid for mobile devices.
- Regarding Tables, we saw how to specify the styles of the columns and rows. We talked about measurement units DIPs, Percentages, and the PD—Platform Default—value for Rows.
- Finally, we talked about image scaling and its different values: No Scale, Tile, Fill, Fill Keeping Aspect Ratio, and Fit.

This helped us to improve the look and feel of the travel agency application.

In future videos, we will continue to develop other features to complete the application.

GeneXus™

training.genexus.com
wiki.genexus.com
training.genexus.com/certifications