

# 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.

localhost:55539/View\_Attractions\_MoreInfo-Level\_Detail








View\_Attractions\_More Info



Country (None)

Name From Name From

Name To Name To

1	Eiffel Tower		Paris	France	3
2	Glenfinnan Viaduct		Glenfinnan	Scotland	1
3	Meet the Emperor		Beijing	China	2
4	Christ the Redeemer		Rio de Janeiro	Brazil	2
5	Rifugio Nuvolau		Belluno	Italy	1
5	London Towers		London	England	1
7	Louvre		Paris	France	1
Total Trips		20			

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 screenshot displays the GeneXus IDE interface. On the left, a form layout is visible with the following elements:

- A dropdown menu for "Country" with the variable "&CountryId".
- Text input fields for "Name From" (variable "&AttractionNameFrom") and "Name To" (variable "&AttractionNameTo").
- A "GRID" containing columns: "AttractionId", "AttractionName", "CityName", "CountryId", "CountryName", and "Trips".
- A "Total Trips" field with variable "&TotalTrips".

On the right, the "Properties" panel is open for the "Attribute/Variable: &CountryId". The "Label Position" property is highlighted and set to "Platform Default". Other visible properties include:

- Control Name: &CountryId
- Attribute: &CountryId
- Label Caption: Country
- ReadOnly: False
- Class: Platform Default
- Visible: None
- Invisible Mode: Left
- Enabled: Top
- Format: Right
- Invite Message: Bottom
- Link: Float
- Autolink: True
- Virtual Keyboard: <Platform Default>

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.

The screenshot displays a design tool interface for a form. A central grid contains several variables: `&AttractionId`, `&AttractionName`, a landscape image, `&CategoryName`, `&CityName`, and `&Trips`. Below the grid is a `&TotalTrips` variable. Above the grid are `&AttractionNameFrom` and `&AttractionNameTo` variables. At the top left is a `&CountryId` variable. Four property panels are open, each corresponding to a variable in the form, with blue arrows indicating the mapping:

- &CountryId:** Control Name: `&CountryId`, Attribute: `&CountryId`, Label Caption: `Country`, Readonly: `False`, Label Position: `Platform Default`.
- &AttractionNameFrom:** Control Name: `&AttractionNameFrom`, Attribute: `&AttractionNameFrom`, Label Caption: `Attraction Name From`, Readonly: `False`, Label Position: `Top`.
- &AttractionNameTo:** Control Name: `&AttractionNameTo`, Attribute: `&AttractionNameTo`, Label Caption: `Attraction Name To`, Readonly: `False`, Label Position: `Right`.
- &TotalTrips:** Control Name: `&TotalTrips`, Attribute: `&TotalTrips`, Label Caption: `Total Trips`, Readonly: `False`, Label Position: `Bottom`.

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

localhost:57283/View\_Attractions\_MoreInfo-Level\_Detail

View\_Attractions\_MoreInfo










Country (None)

Attraction Name From

Attraction Name From

Attraction Name To

4	Christ the Redeemer		Monument	Rio de Janeiro	Brazil	2
1	Eiffel Tower		Monument	Paris	France	3
6	London Towers		Monument	London	England	1
3	Meet the Emperor		Monument	Beijing	China	2
7	Louvre		Museum	Paris	France	1
11	Matisse Museum		Museum	Nice	France	2
10	Smithsonian Institute		Museum	Washington	United States	1

20

Total Trips

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.

localhost:57283/View\_Attractions\_MoreInfo-Level\_Detail

View\_Attractions\_More Info










Country (None)

Attraction Name From

Attraction Name From

Attraction Name To

	Attraction Name From	Image	Attraction Name To	Country	Attraction Name To
4	Christ the Redeemer		Monument	Rio de Janeiro	Brazil 2
1	Eiffel Tower		Monument	Paris	France 3
6	London Towers		Monument	London	England 1
3	Meet the Emperor		Monument	Beijing	China 2
7	Louvre		Museum	Paris	France 1
11	Matisse Museum		Museum	Nice	France 2
10	Smithsonian Institute		Museum	Washington	United States 1

20

Total Trips

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

The screenshot displays a form with a table control named 'MainTable'. The table has 8 columns: Attraction Id, Attraction Name, Attraction Photo, CategoryName, CityName, CountryId, CountryName, and &Trips. The 'Attraction Photo' column contains a small image icon. The 'Country' dropdown is set to '&CountryId'. Below the table, there is a 'Total Trips' label and a corresponding value field.

The properties panel on the right shows the following settings for 'Table: MainTable':

Control Name	MainTable
<b>Appearance</b>	
Columns Style	100%
Rows Style	pd;pd;pd;100%;pd
Width	100%
Height	100%
Auto Grow	True
Class	TableDetail
Background	(none)
Visible	True
Invisible Mode	Keep Space
Enabled	True
<b>Scroll Behavior</b>	
Scroll Factor	1
Zoom Factor	0
Scroll Attachment	Parent
<b>Form</b>	

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\*.

\* Attention: if you have been playing the videos in GeneXus since the beginning of the course, in this Panel you will have 4 more rows, which were used for the text blocks and tables of the titles. In this case, use the row numbers corresponding to your Panel.

## Adjusting the columns of a table

The image illustrates the process of adjusting the columns of a table in GeneXus. It shows three main components:

- Table Properties:** A list of properties for the 'MainTable' control, including 'Columns Style' (50%:50%), 'Width' (100%), and 'Height' (100%).
- Columns Style Dialog:** A dialog box showing the configuration for two columns: 'Col 1' with a width of 30% and 'Col 2' with a width of 70%.
- Table Control:** A visual representation of the table with columns for 'Attraction Id', 'Attraction Name', 'Attraction Photo', 'CategoryName', 'CityName', 'CountryId', 'CountryName', and '&Trips'.
- Cell Information Properties:** A list of properties for the grid, including 'Col Span' set to 2, indicating that the grid should span both columns.

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.

localhost:63596/View\_Attractions\_MoreInfo-Level\_Detail







View\_Attractions\_More Info



Country (None) ▾

Name From Name From

Name To Name To

1	Eiffel Tower		Monument	Paris	France	3
2	Glenfinnan Viaduct		Tourist site	Glenfinnan	Scotland	1
3	Meet the Emperor		Monument	Beijing	China	2
4	Christ the Redeemer		Monument	Rio de Janeiro	Brazil	2
5	Rifugio Nuvolau		Tourist site	Belluno	Italy	1
6	London Towers		Monument	London	England	1

Total Trips 20

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

The screenshot shows the GeneXus IDE interface. At the top, there are tabs for 'Layout', 'Rules', 'Events', 'Conditions', 'Variables', and 'Documentation'. Below these is the 'Application Bar' with 'MainTable' and 'AttractionDescription' buttons. The main workspace displays a form layout with a table containing the following structure:

Visiting the	AttractionName
	Category Name <input type="text" value="CategoryName"/>
	City Name <input type="text" value="CityName"/>
	Country Name <input type="text" value="CountryName"/>

Below the table is a text area labeled 'AttractionDescription'. To the right, the 'Columns Style' dialog is open, showing a table with the following data:

Colu...	Width
1	30%
2	10dip
3	70%

The dialog also shows 'Unit' set to 'Percentage', 'Device' set to 'Independent Pixel', and 'Value' set to '70'. Below the dialog, a diagram illustrates the column widths: 30% (112 DIPs), 10 DIPs, and 70% (262 DIPs), totaling 384 DIPs.

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 screenshot shows the GeneXus IDE interface. At the top, there are tabs for 'Layout', 'Rules', 'Events', 'Conditions', 'Variables', and 'Documentation'. Below these is the 'Application Bar' with 'MainTable' and 'AttractionDescription' buttons. The main workspace displays a layout for 'Visiting the' attraction, with fields for 'AttractionName', 'Category Name', 'City Name', and 'Country Name'. A 'Columns Style' dialog box is open, showing a table with three columns and their widths: 30%, 10dip, and 70%. The dialog also has options for 'Unit' (Percentage, Device, Independent, Pixel) and a 'Value' field set to 70. Below the dialog, a diagram illustrates the width distribution: 30% (112 DIP) for the first column, 10 DIP for the second column, and 70% (262 DIP) for the third column, totaling 384 DIPs.

Colu...	Width
1	30%
2	10dip
3	70%

Unit
<input checked="" type="radio"/> Percentage
<input type="radio"/> Device
<input type="radio"/> Independent
<input type="radio"/> Pixel

Value:

OK Cancel

Diagram: 30% (112 DIP) | 10 DIP | 70% (262 DIP) | Total: 384 DIPs

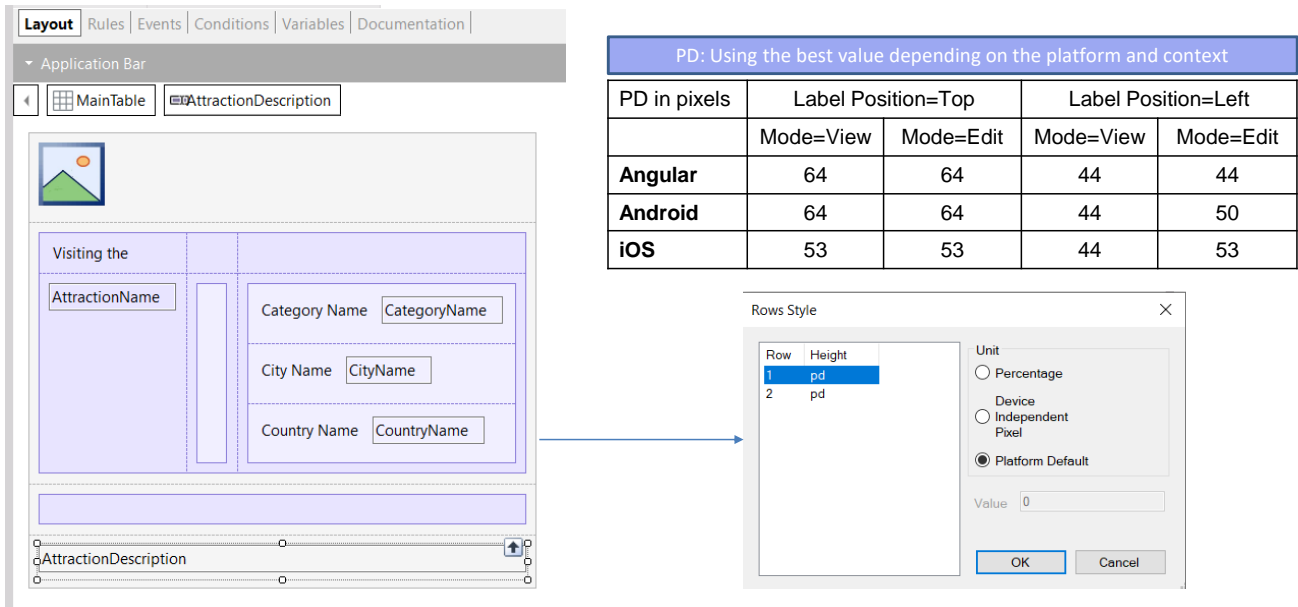
The Device Independent Pixel unit corresponds to the abstraction of 1 pixel; later on, an application converts it to physical pixels, which allow scaling to different screen sizes. Each platform has a different number of pixels; for example, in a web application generated in Angular, 1 DIP is 1 pixel, 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
<b>Angular</b>	64	64	44	44
<b>Android</b>	64	64	44	50
<b>iOS</b>	53	53	44	53

Rows Style

Row	Height
1	pd
2	pd

Unit

Percentage

Device

Independent Pixel

Platform Default

Value: 0

OK Cancel

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.

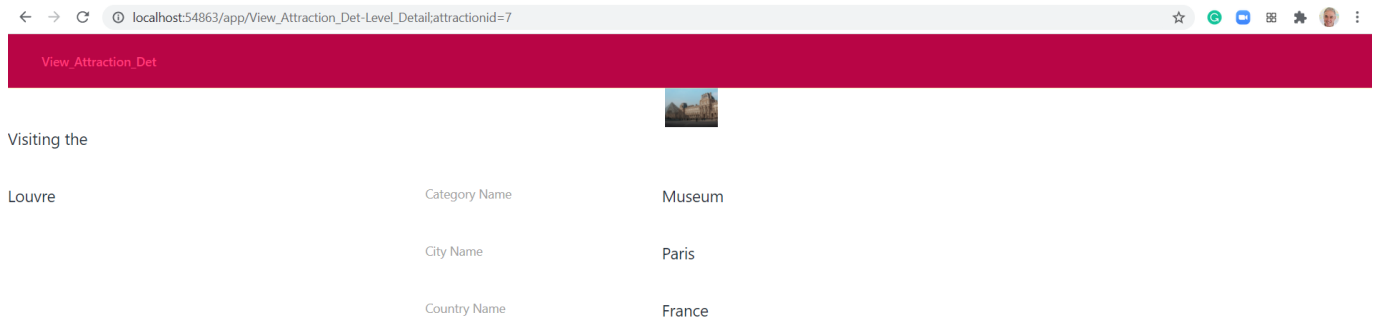
Remember that in Angular 1 DIP is 1 pixel, but in Android or iOS the number of pixels per DIP depends on the device and the orientation.

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



Visit the palace of French kings to admire some of the world's finest art. The Louvre holds many of Western Civilization's most famous masterpieces, including the Mona Lisa by Leonardo da Vinci, and is one of the top things to do in Paris. A large number of the museum's paintings were owned by the various kings who lived in the Louvre when it was a royal residence, other pieces were acquired through France's.

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

The image shows the GeneXus IDE interface for editing a layout. The top navigation bar includes 'Layout', 'Rules', 'Events', 'Conditions', 'Variables', and 'Documentation'. Below it is the 'Application Bar' with 'MainTable' and 'Table1' buttons. The main workspace displays a layout with a header 'Visiting the' and a table with columns 'AttractionName', 'Category Name', 'City Name', and 'Country Name'. Below the table is a row for 'AttractionDescription'. A blue arrow points from the table to the first 'Rows Style' dialog box. This dialog shows a table with 'Row' and 'Height' columns, where all rows have 'pd' as the height. The 'Unit' section has 'Platform Default' selected. A second 'Rows Style' dialog box is shown below, with a blue arrow pointing to it from the first. This dialog shows the 'Height' column with '400dip' for row 1, '10dip' for row 3, and 'pd' for rows 2 and 4. The 'Value' field is set to 400.

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.

View\_Attraction\_Det



Visiting the

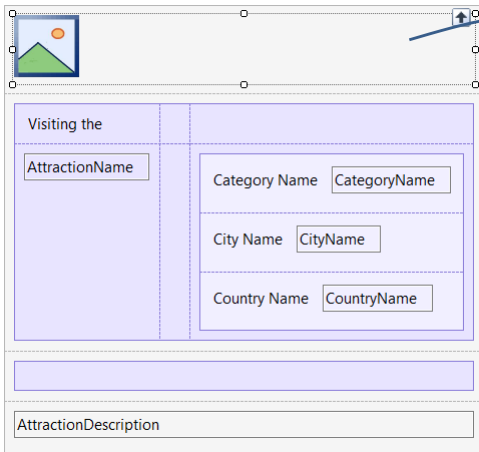
Louvre

Category Name	Museum
City Name	Paris
Country Name	France

Visit the palace of French kings to admire some of the world's finest art. The Louvre holds many of Western Civilization's most famous masterpieces, including the Mona Lisa by Leonardo da Vinci, and is one of the top things to do in Paris. A large number of the museum's paintings were owned by the various kings who lived in the Louvre when it was a royal residence, other pieces were acquired through France's.

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

## Image scaling

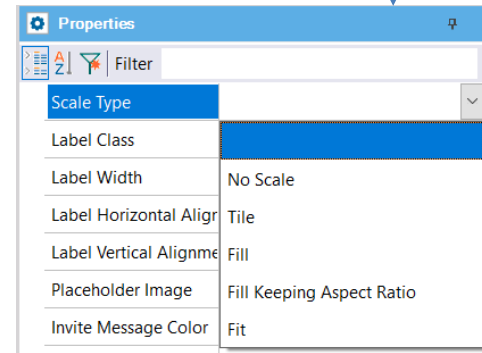
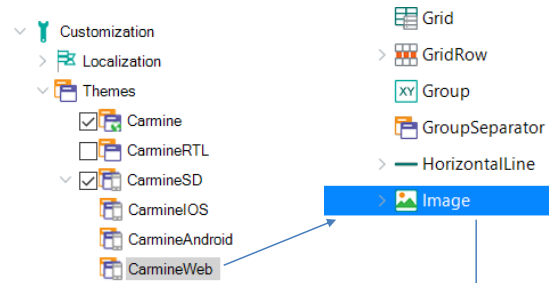


Attribute/Variable: **AttractionPhoto**

Attribute	AttractionPhoto
Readonly	True

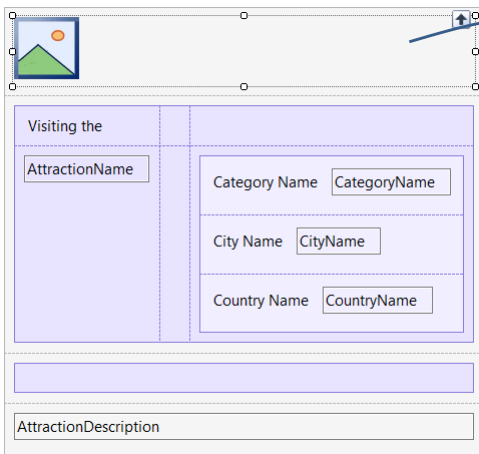
Appearance

Label Position	None
Class	Image



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



▼ Attribute/Variable: **AttractionPhoto**

Attribute	<b>AttractionPhoto</b>
Label Caption	
Readonly	True

▼ Appearance

Label Position	Platform Default
<b>Class</b>	<b>myImage</b>

```

TravelAgency_AngularCourse x
Tokens Styles Documentation |
1  styles Base {
2    @import GeneXusUnanimio.UnanimioWeb;
3
4  0 references
5  .myImage {
6    gx-content-mode: fill-keeping-aspect;
7  }
8  }

```

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

<b>No Scale</b>	Keeps the original size of the image, regardless of the size of the control area.
<b>Tile</b>	The image is not scaled. It is repeated horizontally and vertically to fill the control size.
<b>Fill</b>	The image is scaled in width and height to fill the whole size of the control area.
<b>Fill Keeping Aspect Ratio</b>	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.
<b>Fit</b>	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
<ul style="list-style-type: none"><li>• Platform Default</li><li>• None</li><li>• Right</li><li>• Left</li><li>• Top</li><li>• Bottom</li><li>• Float</li></ul>	<ul style="list-style-type: none"><li>• Columns Styles</li><li>• Row Styles</li><li>• Measure Units</li></ul>	<ul style="list-style-type: none"><li>• No Scale</li><li>• Tile</li><li>• Fill</li><li>• Fill Keeping Aspect Ratio</li><li>• Fit</li></ul>

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

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

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