Multiple Selection in Grid and Commands to Process the Selected Lines

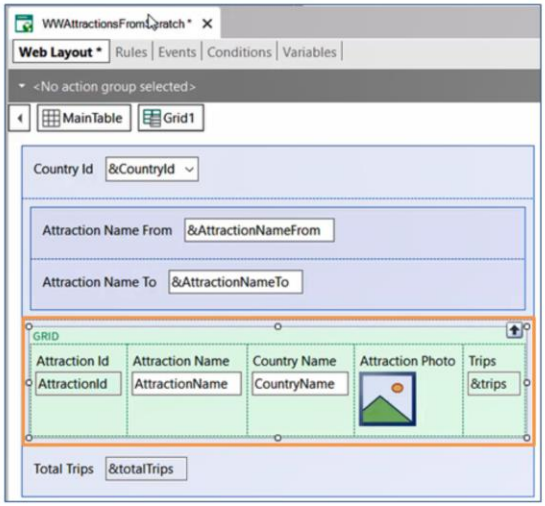
GeneXus™

AttractionId	AttractionName	CountryId	CityId	CategoryId
1	Louvre Museum	2	1	1
2	The Great Wall	3	1	2
3	Eiffel Tower	2	1	2
4	Forbidden City	3	1	2
5	Christ the Redeemer	1	2	2

On several occasions we will need to work with a set of previously selected elements, in order to perform some action with them.
In a grid we can display many elements, and we may want to select only some of them.
Next, we will see how we can perform a multiple selection of elements in a simple way, and how to run through those elements and then process them.

For this example, we will use the application developed for a travel agency.

Web Panel WWAttractionFromScratch

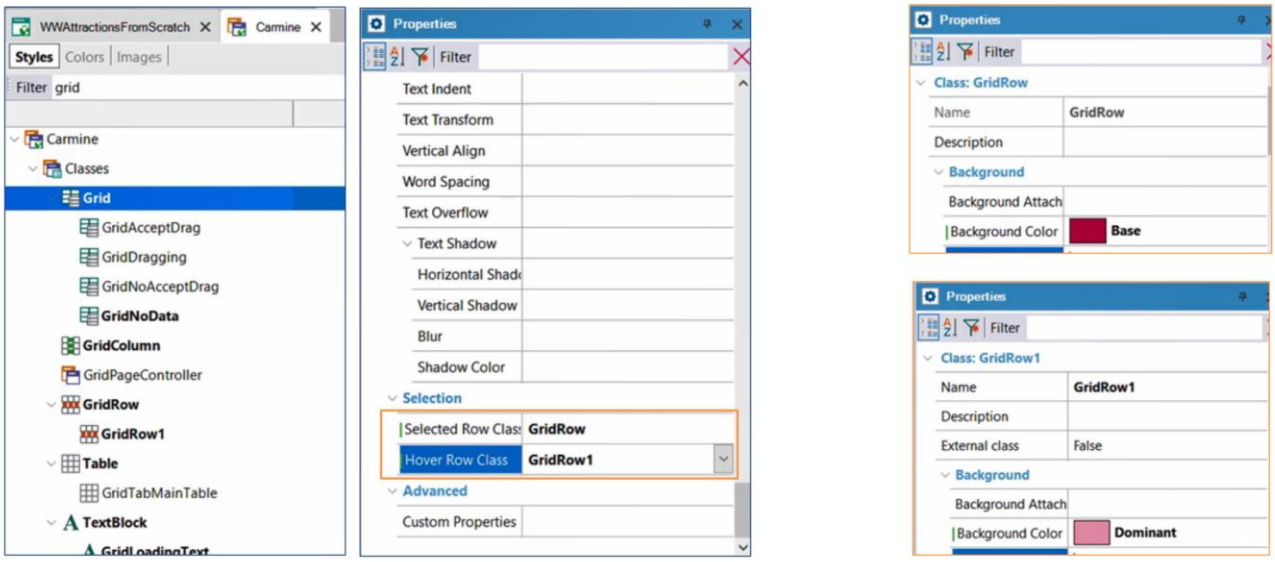


In our KB we have a Web panel that lists all the attractions entered. Suppose that from the list we need to select only one attraction to perform some action with it later; to do so, it is necessary to be able to mark that row of the grid as selected.

In a Web application, if we want to be able to mark a grid row as selected, we go to the grid properties and set the AllowSelection property to True.

After doing it, the Allow Hovering property is enabled. It will allow the rows to be marked with a color when hovering over them, so we leave it set to True. We will see it next.

Changing properties in the Web Panel theme



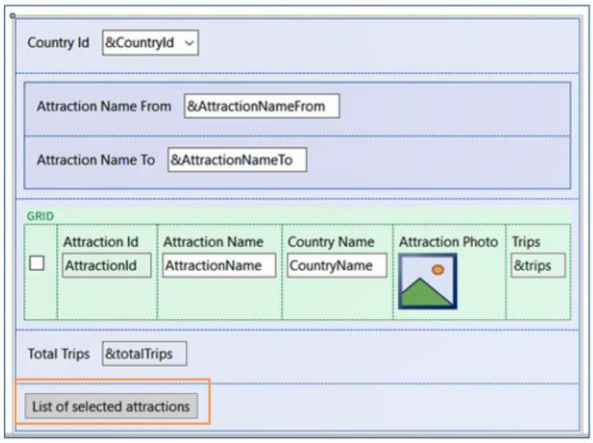
To display it on the screen, it is necessary to configure a color for each functionality. This is done by accessing the theme associated with the web panel that contains the grid. Then in the Grid class we have the Selected Row Class property, where we must assign a class that will contain the color for a selected row; we choose the GridRow class. And the Hover Row Class property, to which we will assign the GridRow1 class, created as a child of GridRow, which will contain the color that will be assigned to the row when we move over it.

Let's see it at runtime.

Now let's suppose that we need to select one or more attractions to print them in a PDF list, so we need to be able to make a multiple selection of attractions in this list.

Setting the AllowSelection property to True does not work because it does not support multiple selection.

WEB




```
Event 'List of selected attractions'
      For each line in Grid1
            if &selected
                  &SelectedAttractionsIds.Add(AttractionId)
            endif
      Endfor
Endevent
```

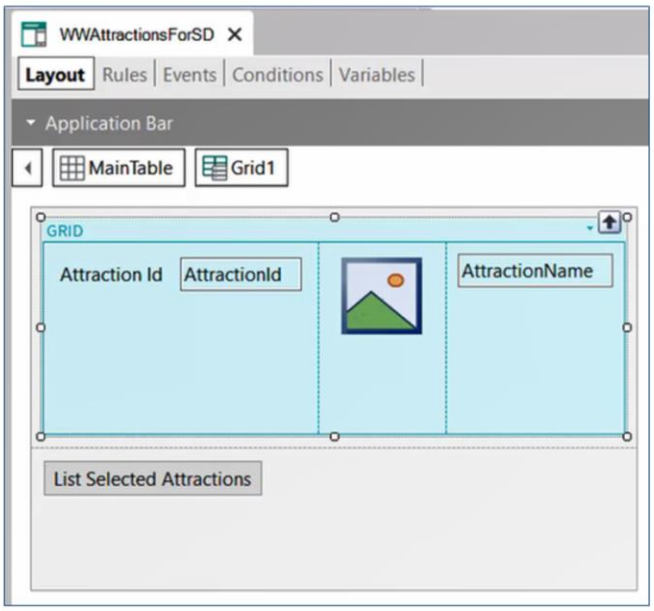
For a Web application, if our grid doesn't have paging, one of the options available to implement it is as follows:
First of all, we create a variable of Boolean type and add it to the grid in the first column. This will be used to select a record.

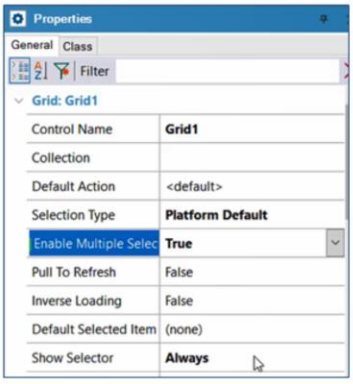
Then we add a button, which will invoke the procedure that will list the selected attractions.
To be able to send these attractions to the procedure, we must first save them in a collection.
Simply saving the ID of the selected attractions is enough, so we create a variable of ID type (domain of numeric type) and mark it as a collection.

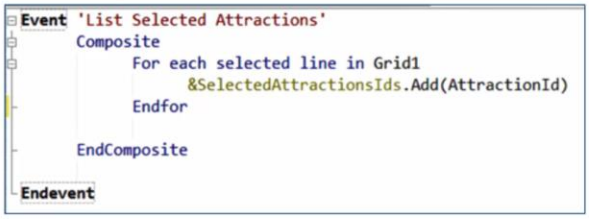
We access the event associated with the button, and here we need to run through each one of the grid rows; we do this using the instruction For Each Line, In, and the name of the grid. This will allow us to retrieve the values of each row of the grid.

Note that the For Each line command can be used for any situation in which the grid must be run through in a user event.
As we are only interested in the selected attractions, we condition the search with If &selected (which is the same as setting If &selected = true), and if this condition is met, we add the attraction ID to the collection variable created.

SD

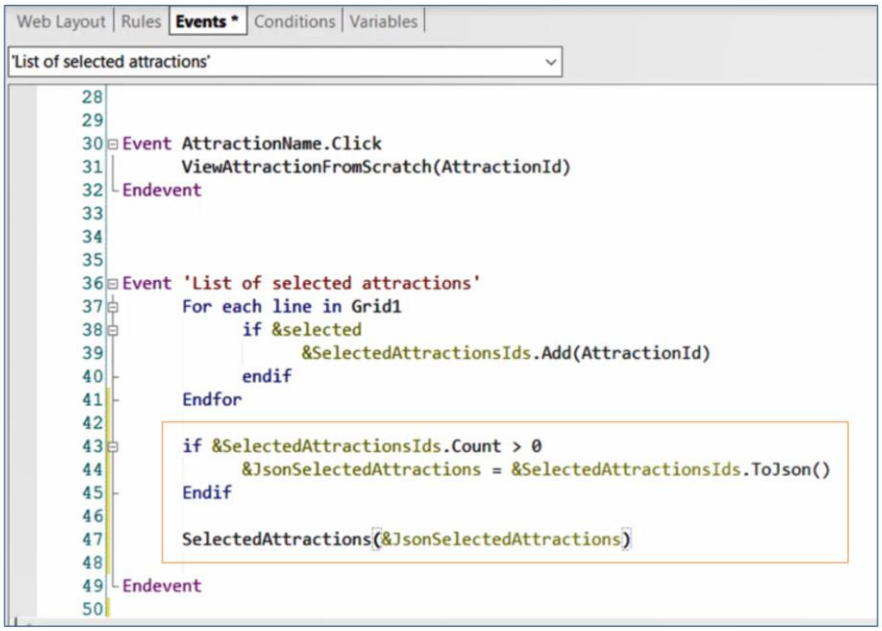






In an application for Smart Devices, it is possible to configure the grid to allow multiple selection by means of a grid property named Enable Multiple Selection; we must set it to true. And to display the checkbox that allows us to select the row, we leave the Show Selector property set to Always. These properties save us from having to declare a Boolean variable, as we just did for the web.

Then, to run through the rows, the command For Each selected line must be used. It must be grouped in a Composite block. What this command will do is that if one of the instructions fails, the rest will not be executed.

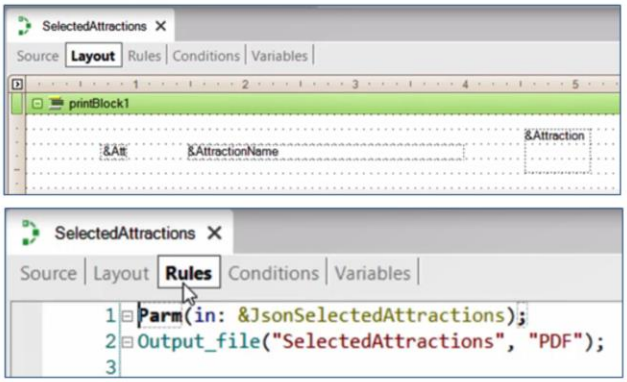


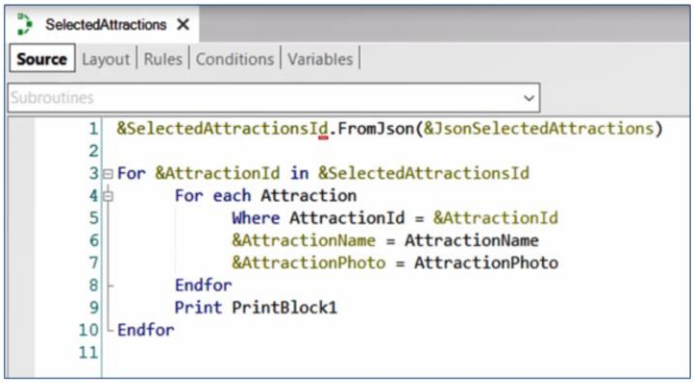
Going back to our Web Panel, once all the grid rows have been run through, we must send the collection of attractions to the procedure. We cannot pass by parameter directly; instead, we will have to serialize its content, and generate a structured format text file, such as a JSON or an XML. Both formats are used for data exchange. In this case, we will use JSON.

To this end, we create a variable with this name, of LongVarchar type. The value it will take will be the value of the collection after the ToJson method is applied.

Finally, we invoke the procedure that will list the selected attractions, which we call SelectedAttractions, and pass by parameter the variable containing the JSON we've just obtained.

Procedure Object





Now, let's see how the called procedure receives and processes all this information.

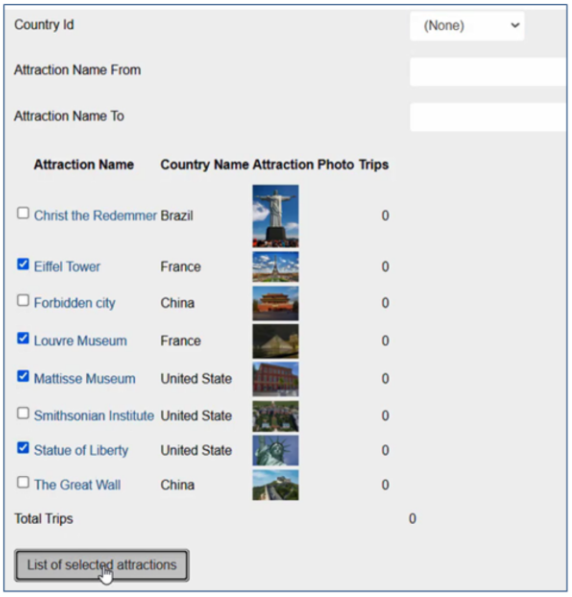
In the Layout, we enter only these three variables to show information of the selected attractions.
In the rules section of the procedure, in the Parm rule, we declare an input variable, of type LongVarChar, in charge of receiving the list of attractions that we are passing to it.

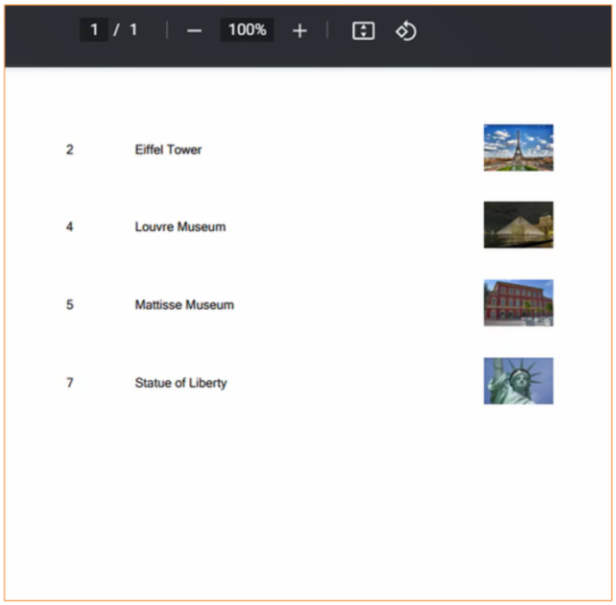
In the Source, we declare a variable of collection type, to which we will load the content of the variable that contains the value received by parameter, applying the FromJson method.

In this way, the IDs of the selected attractions are stored in this collection variable. Now we will have to run through it to access them. We will do it by means of the following instruction.

Next, we run through the Attraction table, and for each AttractionId we get the name of the attraction and its photo, loading them into these two variables created for this purpose.
After we close this For Each, we print the printblock containing the variables that will show the ID, name, and photo of our attractions.

At runtime





Now let's try this functionality we've just programmed.

Remember that, for this example, we want to select some attractions and have a PDF list of them printed.
We access the web panel that shows the list of all the attractions entered, and select some of them. Next, we press the button to execute this action, and the list with the attractions that were selected is displayed.

Summary

- Selecting a Grid row
  - **AllowSelection Property** set to True
- Selecting multiple Grid rows
  - Web
    - **Adding a Boolean variable to the Grid**
    - Rows are run through with For each line
  - SD
    - **Enable Multiple Selection Property** set to True
    - Rows are run through with **For each selected line**
- Passing data by parameter
  - Content is serialized to Json format using the ToJson() method.
  - When the data is received, it must be loaded into a collection variable using the FromJson() method.

To sum up what we've seen in this video:

To select only one row of our grid, changing the AllowSelection property of the grid to True and assigning it a color from the theme is enough.

If, as we've just seen in the example above, we need to select more than one row, this property will not be helpful, so we must use a Boolean variable in the grid to store the row selection.

Then, to run through the grid rows, the For each line command must be used. This command iterates over the rows loaded in the grid, and for each iteration we will obtain the values of the grid columns for each row run through.
To know if the row was selected or not, we will check the value of the Boolean variable.

In applications for Smart Devices, the grid does have a property for this functionality: the Enable Multiple Selection property.

Then the show selector property is used together with the For each selected line command to run through these rows.

To pass the collection of attractions to the procedure, the content must be serialized. And once received by parameter from the other object, to be able to handle the records, we create a collection variable and load it with this data by means of the FromJson method.

For more information, you can visit our Wiki.

GeneXus™

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