

Paging

For Each Command, Pattern, and Grid







GeneXus™

Pattern Work With for Web

GeneXus Application Name

Attractions

Insert Q Name

Id	Name	Country Name	Category Name	Photo	City Name	Description	UPDATE	DELETE
4	Christ the Redeemer	Brazil	Monument		Rio de Janeiro	An Art Deco statue of Jesus Christ in Rio de Janeiro	UPDATE	DELETE
12	Cinque Terre	Italy	Tourist site		Maranola	A string of centuries-old seaside villages on the rugged Italian Riviera coastline	UPDATE	DELETE
3	Eiffel Tower	France	Monument		Paris	Wrought-iron lattice tower on the Champ de Mars in Paris	UPDATE	DELETE
7	Forbidden city	China	Tourist site		Beijing	The former Chinese imperial palace and state residence of the Emperor of China from the Ming dynasty	UPDATE	DELETE
8	Glenfinnan Viaduct	Scotland	Tourist site		Glenfinnan	A railway viaduct on the West Highland Line in Glenfinnan. Located at the top of Loch Shiel in the West Highlands of Scotland, the viaduct overlooks the Glenfinnan Monument and the waters of Loch Shiel	UPDATE	DELETE
13	London Bridges	United States	Tourist site		San Francisco	Several bridges named London Bridge have spanned the River Thames between the City of	UPDATE	DELETE

10 Rows (Default)

As we have already seen in previous videos, when we use the grid of the Work With pattern of a transaction, it offers automatic paging. This means that N lines are loaded per page, and through the navigation controls, we can move through the different pages. By default 10 records per page will be loaded, but we will see in a moment that it is possible to modify this value.

Pattern Work With for Web

The screenshot shows the GeneXus IDE interface. The top pane displays the 'Patterns' configuration for the 'Attraction' transaction. The tree view shows the following structure:

- Work With for Web
- Work With for Smart Devices
- Apply this pattern on save
- Work With Pattern Instance
 - Transaction (Attraction)
 - Level (Attraction)
 - DescriptionAttribute (AttractionName)
 - Selection (Attractions)
 - modes (Insert, Update, Delete)
 - Attributes
 - AttractionId
 - AttractionName
 - CountryName
 - CategoryName
 - AttractionPhoto

The 'Properties' pane on the right shows the configuration for the selected 'selection: Selection (Attractions)' node:

Caption	
Description	Attractions
Is Main	False
Master Page	<default>
Paging	
Rows per page	<custom>
Custom Rows	5
Paging Mode	<default>
Show Current Page	<default>

Below the IDE, a preview of the 'Attractions' table is shown. The table has the following columns: Id, Name, Country Name, Category Name, Photo, City Name, Attraction, and Price. The table displays 5 rows of data, with a '5 Rows' indicator on the left side.

We select the Attraction transaction from our Travel Agency application. If within the Pattern section we choose the Selection node and see its properties, there is one called "Rows per page" with "default" value. As we've just said, that default value will be 10. If we want to show a different number, we simply change to "Custom," and the "Custom Rows" property will be enabled to indicate the number of records per page to be shown. Let's see the change at runtime.

Paging in WWAttractionsFromScratch

The diagram illustrates the configuration of a grid in GeneXus. It shows two scenarios for the 'PagingMode' property:

- Scenario 1 (Top):** The 'PagingMode' property is set to 'Infinite scrolling'. The grid displays 5 records: The Great Wall (China), Eiffel Tower (France), Christ the Redeemer (Brazil), and Smithsonian Institute (United States). The 'Loading...' indicator is visible at the bottom of the grid.
- Scenario 2 (Bottom):** The 'PagingMode' property is set to 'One page at a time'. The grid displays only the first record: Louvre Museum (France). The 'Loading...' indicator is visible at the bottom of the grid.

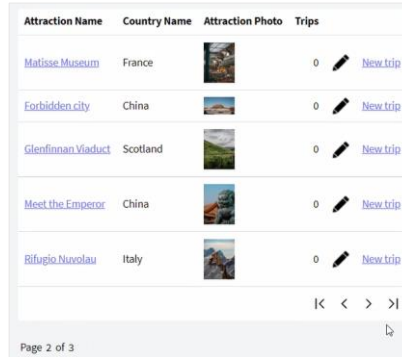
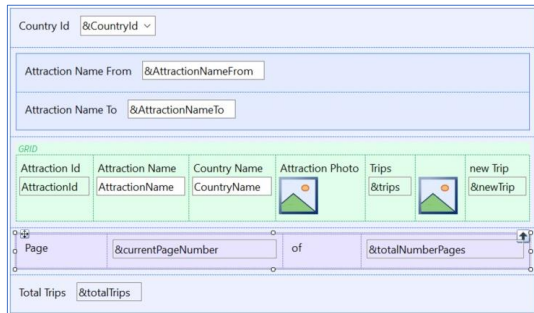
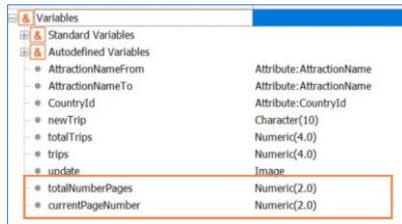
Now let's see how we can implement paging in a web panel created by us, such as WWAttractionsFromScratch that we created for our travel agency application.

In the grid properties, we see one called "Rows." By changing this value we can change the number of records to show on screen. At that moment the Paging property is enabled, which will give us the option to implement an infinite scroll. Let's see this functionality at runtime.

The grid loads the first records, and when you scroll on the grid, or simply move down with the sidebar, it will load the remaining records until loading all of them.

Now we leave the paging property with its default value again.

We see that only the number of attractions that we entered in the Rows property of the grid is shown on this page.



Let's suppose that, during the execution of the program, we want to know in which page of the grid we are at each moment; for example, to be able to show the user this information on the screen.

For this, we declare two variables of numeric type. In one we will save the current page, and in the other we will save the total number of pages of our grid.

In the Web Layout, we enter the two variables and a couple of text block objects to make clear what we want to show.

In the events section, in the Start event, we define the variable `totalNumberPages`, to which we assign the value of the `PageCount` property of the grid. This property gives the total number of pages contained in the grid. We did it in the Start event, since for this example we need to obtain this information only once, when the program is executed.

And in the Refresh event, we define the variable `currentPageNumber`, which will take the value of the `CurrentPage` property of the grid. This property returns the number of the page on which we are currently positioned. We program it in the Refresh event, since we must calculate

this value every time the page changes, and every time this happens the Refresh event is triggered, so here is a good time to do it.

Let's see it at runtime.

Paging in WWAttractionsFromScratch

Web Layout

The screenshot shows a web layout for a search application. At the top, there is a search form with three input fields: 'Country Id' (with a dropdown arrow), 'Attraction Name From', and 'Attraction Name To'. Below the form is a grid with six columns: 'Attraction Id', 'Attraction Name', 'Country Name', 'Attraction Photo', 'Trips', and '&newTrip'. The grid contains two rows of data. Below the grid is a yellow bar, and at the bottom, there are four navigation buttons: 'First Page', 'Previous Page', 'Next Page', and 'Last Page'. At the very bottom, there is a label 'Total Trips' followed by a text box containing '&totalTrips'.

Events

```

32 | Event 'FirstPage'
33 |     Grid1.FirstPage()
34 | - Endevent
35 |
36 | Event 'PreviousPage'
37 |     Grid1.PreviousPage()
38 | - Endevent
39 |
40 | Event 'NextPage'
41 |     Grid1.NextPage()
42 | - Endevent
43 |
44 | Event 'LastPage'
45 |     Grid1.LastPage()
46 | - Endevent
47 |

```

Now let's see how we can create our own links or navigation buttons, to customize them to our preferences in terms of design and location.

For this example, we enter 4 buttons in the web panel, outside the grid, which will be our navigation buttons. And then we configure the associated event for each one of them.

We will use the following four grid methods.

FirstPage for our first button, which will take us to the first page of records, PreviousPage for the second one, which will show us the page previous to the one currently selected. Then in the third button we will use the NextPage method, which will return the following set of records. And finally LastPage, which will show us the last page of records. Let's see it at runtime.

Paging in PDF Report

PDF Report

ID	Name	Country	Photo
1	Louvre Museum	France	
2	The Great Wall	China	
3	Eiffel Tower	France	
4	Christ the Redeemer	Brazil	
5	Smithsonian Institute	United States	
6	Musée d'Orsay	France	
7	Forbidden City	China	
8	Glenelg Harbour	Scotland	
9	Great Wall of China	China	
10	Big Ben London	Italy	

Source

```

AttractionsReport X
Source | Layout | Rules | Conditions | Variables
Subroutines
1 print Title
2 print ColumnTitles
3 For each Attraction
4   skip 2 count 5
5   print Attractions
6 -endfor
7

```

=

```

AttractionsReport X
Source * | Layout | Rules | Conditions | Variables
Subroutines
1 print Title
2 print ColumnTitles
3 @pageSkip = 2
4 @pageSize = 5
5 For each Attraction
6   skip @pageSkip count @pageSize
7   print Attractions
8 -endfor
9

```

Now, suppose we have a procedure that lists all the attractions in PDF format. For this case, how can we handle a potentially large number of records? We can do it within the For Each command with two clauses that we will see next, through which we will be able to control how many and what records to show in the list.

We access the “AttractionsReport” procedure, which lists all the attractions.

Let's suppose that, for some reason, we want the first 2 records not to be shown; that is, to list from the third record we have entered. To this end, we will use the Skip clause and then enter the number of records to be ignored, in this case 2.

From here we can also control the number of records we want to show, which will be through the Count clause. In this example, the first two attractions will be skipped, and the next 5 will be printed.

Instead of numbers it is certainly more convenient to use variables, whose value may be sent in a parameter or declared directly in the source.

Paging in PDF Report



```
1 print Title
2 print ColumnTitles
3 &pageNumber = 2
4 &pageSize = 5
5 For each Attraction
6     skip &pageSize*(&pageNumber-1) count &pageSize
7     print Attractions
8 endfor
9 |
```

For example, suppose we handle pages of 5 records each, and we want only the records of the second page. We could do this by declaring a new variable, where we will enter the page we want to show. And we implement it as follows.

For more information and details on this topic, visit our [Wiki](#).

GeneXus[™]

training.genexus.com
wiki.genexus.com