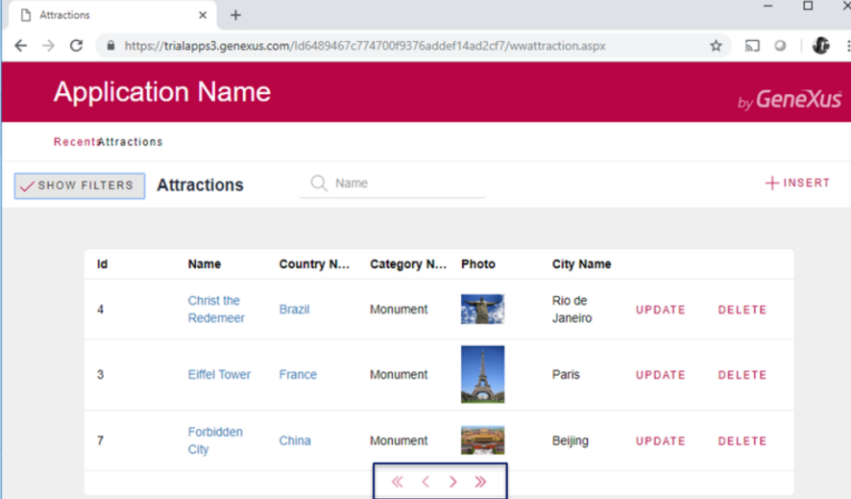


Paging Clauses in For Each Command




GeneXus 16 Course

GeneXus 16

Where was paging already applied?



The screenshot shows a web browser window with the URL `https://trialapps3.genexus.com/Id6489467c774700f9376addef14ad2cf7/jwattraction.aspx`. The application header is red with the text "Application Name" and "by GeneXus". Below the header, there is a search bar with the text "RecentAttractions" and "Attractions". A "SHOW FILTERS" button is visible on the left, and a "+ INSERT" button is on the right. The main content is a table with the following columns: "Id", "Name", "Country N...", "Category N...", "Photo", and "City Name". The table contains three rows of data:

Id	Name	Country N...	Category N...	Photo	City Name
4	Christ the Redeemer	Brazil	Monument		Rio de Janeiro
3	Eiffel Tower	France	Monument		Paris
7	Forbidden City	China	Monument		Beijing

At the bottom of the table, there is a pagination control with four arrows: a double left arrow, a single left arrow, a single right arrow, and a double right arrow. An arrow points from the text "To move between pages" to the pagination control.

Automatic paging for the grid was already included in the Work With grid. What does this mean? N lines per page are loaded. To move from page to page and, for example, to move to the next N lines of information, you use the options that you see in the slide.

The size of the page is configurable. By default it is 10 rows. In the image we are seeing that we have changed it to 3, as indicated below.

Let's imagine that for some reason we want to get the list of attractions, but instead of showing all the attractions, we want to filter by paging.

What does "paging" mean? That all the information is divided into blocks (called pages) that contain a fixed amount of information (in our example, 3 tourist attractions). The Work With component contains an example that we've already seen, where the grid is paged so that it shows 10 records per page.

In the example, each page is made up of three attractions.

Where was paging already applied?

Default value: Rows per page <default>

The screenshot shows the GeneXus IDE interface. On the left, the 'Instance Data' tree is expanded to show a 'Selection (Attractions)' node. On the right, the 'Properties' window is open, displaying the configuration for this selection. The 'Paging' section is expanded, showing the 'Rows per page' property set to '<custom>' and 'Custom Rows' set to '3'. A blue box highlights these two properties. An arrow points from the 'Default value: Rows per page <default>' text above to the '<default>' value in the 'Rows per page' dropdown menu.

Property	Value
Caption	
Description	Attractions
Is Main	False
Master Page	<default>
Paging	
Rows per page	<custom>
Custom Rows	3
Paging Mode	<default>
Show Current Page	<default>

Here we changed the default value of the property “Rows per page” in the Selection node, which was <default> to <custom> in order to customize it. This opens a new property, “Custom Rows”. We enter 3.




Note: we won't see it here, but the <default> value is configured in a centralized location of the KB, which is not enabled in the trial version.

Paging clauses in For Each command

- How is it implemented?



Attraction List

Id	Name	Country	Photo
4	Christ the Redeemer	Brazil	
5	Smithsonian Institute	United States	
6	Matisse Museum	France	

Page 2: 3 attractions of 7

↑
Page size 3

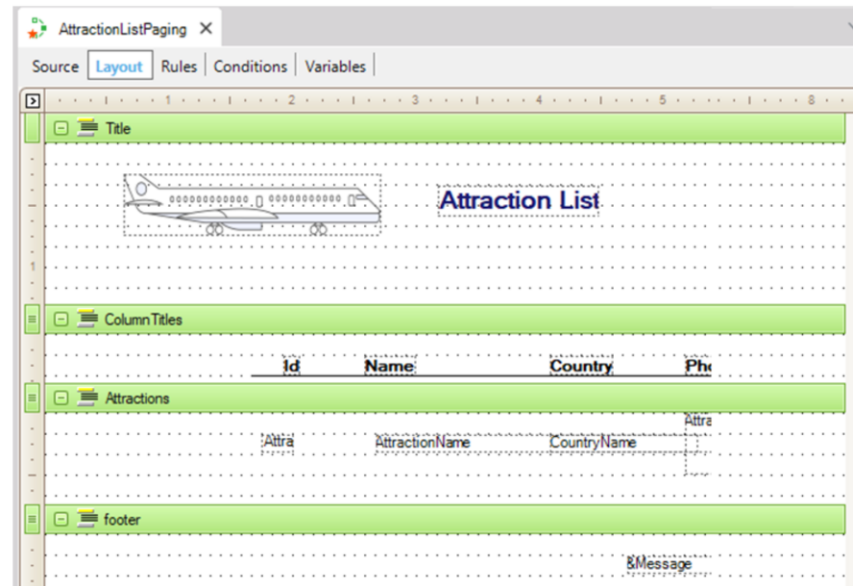
To implement paging for the For Each command, there are a couple of clauses to do it.

Let's imagine that for some reason we want to get the list of attractions, but instead of showing all the attractions, we want to filter by paging three records at a time.

In the example, the pages consist of three records (three attractions), and we want to list the content of the second page only, according to the order of the attractions by ID.

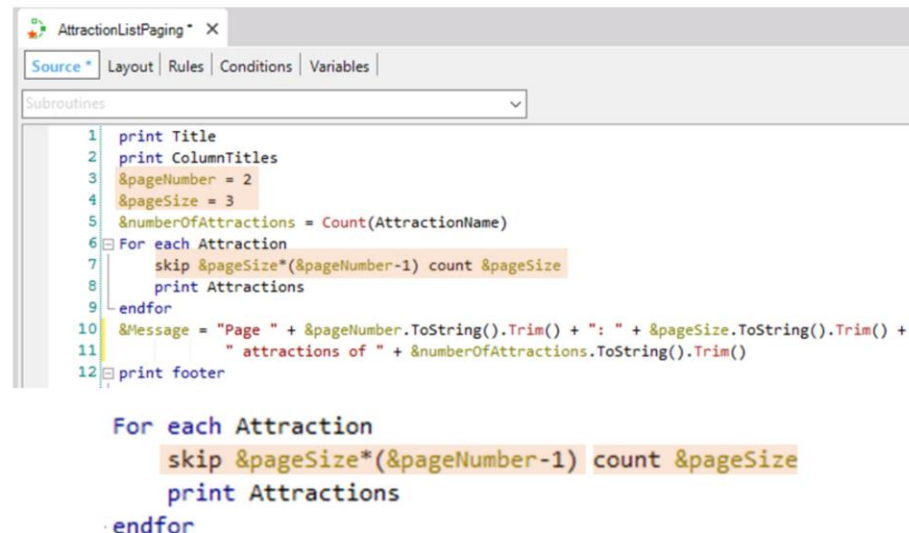
How do we implement this?

Paging clauses in For Each command



In the layout we add the footer printblock with a &Message variable of Character(20) type.

Paging clauses in For Each command



```
1 print Title
2 print ColumnTitles
3 &pageNumber = 2
4 &pageSize = 3
5 &numberOfAttractions = Count(AttractionName)
6 For each Attraction
7     skip &pageSize*(&pageNumber-1) count &pageSize
8     print Attractions
9 endfor
10 &message = "Page " + &pageNumber.ToString().Trim() + ": " + &pageSize.ToString().Trim() +
11     " attractions of " + &numberOfAttractions.ToString().Trim()
12 print footer
```

For each Attraction
 skip &pageSize*(&pageNumber-1) count &pageSize
 print Attractions
endifor

Look that the Skip and Count clauses of the For Each command. In this way, we indicate that we want to skip the first N records that would be retrieved by the For Each command if the clause wasn't included. N is the result of evaluating the $\&pageSize * (\&pageNumber - 1)$ expression. Note that this calculation allows us to skip all the records of previous pages. Then, with the Count clause we're saying that we want to keep the next M records. M is the value of $\&pageSize$, which represents the number of records in each page.

Here, since we haven't explicitly included an Order clause, it is ordered by AttractionId.

Paging clauses in For Each command: Paging in the DB server

For Each Attraction (Line: 14) ⌵

Order: [AttractionId](#)
Index: IATTRACTION

Navigation Start from: FirstRecord

filters: Loop while: NotEndOfTable

Join location: Server

Optimizations: Server Paging

Attraction ([AttractionId](#))

Country ([CountryId](#))

As we can see, this query is made in the database in an optimized way. The DBMS will select the records to return.

Paging clauses in For Each command

```
For each BaseTransaction
  skip expression, count expression,
  order att1, att2, ... , attn [when condition]
  order att1, att2, ... , attn [when condition]
  unique att1, att2, ... , attn
  using DataSelector(parm1, parm2, ... , parmn)
  where condition [when condition]
  where condition [when condition]
  where att IN DataSelector(parm1, parm2, ... , parmn)

  main code

When none
  ...
Endfor
```

Here are the Skip and Count clauses that allow paging the For Each command, with the other clauses that we've seen so far.

As we've said before, with SKIP we indicate that we want to skip the first X records (X is the result of evaluating the expression that follows SKIP) and keep the next Y records. Y is the value of the expression indicated after COUNT.

Genexus™

The power of doing.

Videos

training.genexus.com

Documentation

wiki.genexus.com

Certifications

training.genexus.com/certifications