Web Screens

Base tables and navigation in web panel with multiple grids

GeneXus

GeneXus^{*}

Web Panel with SEVERAL Grids

But, what happens when a web panel has more than one grid? In that case, it's not possible to consider the base table of the web panel, but rather that of each grid.



The definition of navigations will depend on whether the grids are parallel or nested.

Consider the case of parallel grids in the first place: each grid will determine its own navigation independently from the other. So, it is possible that a base table exist for one grid and not for another grid.

In this example, both grids will have base table, because, clearly, you can see attributes in each of them, and that is enough to confirm that there will be an implicit navigation in each grid.

And the question is: how do we determine the base table for each of them?

Start by noting that this example differs from the previous one only in the fact that you have added the grid on the right and a variable to filter the data in that grid and another one to show a total.

It is a known fact that, in addition to the generic Refresh event of the **whole panel, when there's more than one grid, the generic Load event** disappears. Now you have specific Refresh event and Load event for each grid. Each Load event will be triggered only once or N times depending on

whether GeneXus finds a base table for that grid or not.

It is easy to infer that the base table of the first grid will be Attraction, and the base table for the second grid will be CountryCity, and in both cases the filtering will be by CountryId, an attribute received by parameter which, as usual, will not take part in the definition of base tables at all. It will be part of what follows such definition.

However, you could think that, since both tables are related in the database (note that, in fact, CountryCity is part of the extended table of Attraction, so, for each attraction loaded on this grid, there will be an **associated record in this table... or considering it the other way around,** for each city loaded in this other one, there will be N related attractions). As mentioned, you could think that this relation will have an effect on what is loaded into the grids, but it actually will not. GeneXus will not define any implied relation between them.

All the attractions in the country received by parameter will be loaded in the first grid, and all the cities in that country will be loaded in the second grid.

With this possible confusion now cleared, focus now on how GeneXus defines the base table for each grid.

GeneXus Attributes in the grid (visible or Web Form Rules Events Conditions Variables • hidden) 1 □ parm(in: CountryId); Grid1 Control Name • Grid Base Trn property Country Name CountryName Collection • Grid Order property Base Trn Attraction Order Countryld, AttractionName Grid Conditions property Attraction Name From &AttractionNameFrom • Conditions AttractionName >= &AttractionName... Grid Unique property • Attraction Name To &AttractionNameTo Unique Grid Data Selector property Save State False • Data Selector (none) Attributes in the grid's Load Attraction Id Attraction Name Attraction Photo Trips • Appearance AttractionName AttractionId &trips &update 0 event Layout (without context, ie: For each Behavior command and inline aggregate Total Trips &totalTrips Cell information formula) Row information Event Grid1.Refresh Event Grid2.Refresh &totalTrips = 0 &totalAttractions = 0 Endevent Endevent Event Grid2 Load Event Grid1.Load &attractions = Count(AttractionName) &trips = Count(TripDate)
&totalTrips = &totalTrips + &trips &totalAttractions = &totalAttractions + &attractions endevent Endevent

Take the first one in line.

Consider the grid's attributes (visible or hidden), the same grid properties as for the case of a single grid (the Base Transaction, obviously, and properties: Order, Conditions, Unique, and Data Selector). Unlike in previous cases, the "separate" attributes of all events will not be considered here. Only the Load event of the grid will be. So, in the case that a separate attribute existed in the Refresh event, it would not be participating.

GeneXus



And the same goes for any other event. Like these others.

Web Form Rules Events Conditions Variables		• Attributes in the grid (visible or hidden)
Attraction Name From &AttractionNameFrom Attraction Name To &AttractionNameTo GRID Attraction Id Attraction Name Attraction Id AttractionName Trips &AttractionId AttractionName &&meVTrip Total Trips &totalTrips	City Name &cityName CRID City Id City Name Attractions CityId CityName &attractions Total Attractions &totalAttractions We come Control Name Genet	 Grid Base Trn property Grid Order property Grid Conditions property Grid Unique property Grid Data Selector property Attributes in the grid's Load event
Event Grid1.Refresh &totalTrips = 0 Endevent Event Grid1.Load &trips = Count(TripDate) &totalTrips = &totalTrips + &trips Endevent	Collection Base Tin Attraction Outer Countryld, AttractionName - Condition: AttractionName -> &AttractionName Unique Fable Data Selector (none) > Appearance > Lapoot > Lapoot > Coll Information > Des informa	(without context_ie: For each c(+ fixed-part attributes_ggregate formula)

In addition, for the case of the first grid -and only for it- if attributes exist in the fixed part, as is the case, these attributes will also be considered for determining the base table. And only for it. In the case of all the other grids none of the attributes of the fixed part will participate.

So, in our case, in determining the base table of Grid1 all these grid attributes are considered, in addition to any others "separated" in this Load event. There isn't any. And, of course, the grid properties mentioned.

It is clear why there will be base table, which is Attraction. If any of these attributes were not in Attraction's extended table, then we would be warned in the navigation list.

Web Form Rules Events Conditions 1 = parm(in: CountryId Country Name CountryName Attraction Name From & AttractionNam Attraction Name To & AttractionNam GRD Attraction Id Attraction Name AttractionName Attraction Id Attraction Name AttractionName Attraction Id Attraction Name AttractionName	s Variables)); lamefrom teTo traction Photo Trips	City Name &cityName GRD City Id CityName A CityId CityName 8	ttractions attractions	 Attributes in the grid (visible or hidden) Grid Base Trn property Grid Order property Grid Conditions property Grid Unique property Grid Data Selector property Attributes in the grid's Load
Control Name Gr Total T Collection	rid2	Total Attractions &totalA	tractions	event (without context, ie: For each command and inline aggregate
Eve Order Citer Conditions Citer Cit	ityName like &cityName whe	Event Grid2.Refresh &totalAttractic Endevent	ins = 0	formula)
Eve Unique Fa	alse	Event Grid2.Load &attractions = &totalAttractic endevent	Count(Attraction ns = &totalAttra	onName) ractions + &attractions

Additionally, in order to determine the base table of Grid2, the ones **considered will be all these grid attributes and any others "separated" in** the Load.

In this case, there isn't any either. Also the Grid's properties will be considered.

We can clearly see why the base table is CountryCity. In this case, the CountryName attribute of the fixed part doesn't participate.



But what happens with the attributes that appear in these other events? They must just belong to the extended table of any of the grid base tables. Otherwise, we will informed in the navigation list.

In this case, we have CountryName and AttractionId.

GeneXus		Application Name
Country Name Attraction Name From Attraction Name To Attraction Name Attractio Eiffel Tower	France	City Name City Id City Name Attractions 1 Paris 2 2 Nice 1
Louvre Museum	0 New trip	Total Attractions 3

Like we said, for parallel grids, navigations are not related automatically.

If, for example, when the user clicks on a line of the grid that shows cities, we wanted the grid that shows tourist attractions to only show those in that city, as we can see here, how could we do this?

						GeneXus*
					Grid: Grid2	
					Control Name	Grid2
/eb Form Rules Events Conditions Variables					Collection	
1=parm(in: CountryId);					Base Trn	Country.City
					Order	
Country Name CountryName					Conditions	CityName like &cityName w
Attraction Name From & AttractionNameFrom	זו	City Name &cityNar	me	1	Unique	
					Save State	False
Attraction Name To &AttractionNameTo	-	GRID			Data Selector	(none)
GRID	_	City Id City Name	Attractions		> Appearance	
Attraction Id Attraction Name Attraction Photo Trips AttractionId AttractionName & & & & & & & & & & & & & & & & & & &		cityiu cityivame			> Layout	
	-				✓ Behavior	
	-	Total Attractions &t	otalAttractions		Sortable	True
					Allow Drop	False
Grid1's Conditions AttractionName >= &AttractionNameFrom					Allow Drag	False
when not &AttractionNameFrom.IsEmpty();		Event Grid2	2.OnLineActi	ivate	Notify Context Ch	False
<pre>AttractionName <= &AttractionNameTo when not &AttractionNameTo.IsEmpty();</pre>		&CityI Grid1.	[d = CityId .Refresh()		Allow Collapsing	False
<pre>CityId = &CityId when not &CityId.IsEmpty();</pre>		Endevent			Allow Selection	True
< >> ×					Allow Hovering	True
OK Cancel						1

There are several ways to go about it. We will show the one implemented here.

Following a Save of our panel, we set the AllowSelection property for the cities grid to allow the selection of a line through a click on any part of it. We can make it appear with a different color, or not.

We also program the grid's OnlineActivate event, for a line to be triggered when the user selects it, with the possibility of assigning the city identifier of the line selected to a variable.

Then we refresh the attractions grid, because we added a new condition: that only the attractions whose city matches that of the &Cityld variable be loaded, to the extent that the latter is not empty.

This will achieve the behavior shown in runtime.

GeneXus	⁵⁴					Application Name
Count	try Name	France				
		City Name	Paris			
		Attraction Name	Attraction Photo	Trips		
		Eiffel Tower	A	0	💉 New trip	
		Louvre Museum		0	🖍 <u>New trip</u>	
		City Name	Nice			
		Attraction Name	Attraction Photo	Trips		
		Matisse Museum		0	New trip	

The other alternative is to directly show the attractions for each city in the country received by parameter. This means using nested grids, like we did here.

Web Form Rules Events Conditions Variables 1 parm(in: CountryId); Country Name CountryName	For each Country.City	Attraction (R) Country (R)
GRID GRID Attraction Id AttractionName Trips &trips &update2 &newTrip	print CityPB &trips = 0 For each Attraction &trips = Count() print AttractionPB endfor endfor	Attractionid AttractionName AttractionDescripti Country/d AttractionPhoto Chyld AttractionAddress Category/d Category/d Category/d Category/d Category/lame

We are well aware that having nested grids is similar to having nested For eachs, so the way to define their base tables and resulting navigations will be analogous.

If you program this object as listed, you would have the external For each navigating the CountryCity table, which will have an implied filter by CountryId, so it will go over all the cities in that country, and for each of them it will print its name; and prior to moving on the next one it will execute the internal For each, which will go over the Attraction table, filtering implicitly by country and city, and printing each attraction in that country and city.

This is the navigation that you will achieve in your web panel, but, like always, there are two options: implementing each grid with, or without base table.

		✓ Free Style Grid: Gr	id1
/eb Form Rules Events Conditions Variables		Control Name	Grid1
1 parm(in: CountryId);		Collection	Devenenting
		Save State	False
		Base Trn	Country City
Country Name CountryName	For each Country.City	Order	country.city
		Conditions	
RID	print CityPB	Unique	
GRID	For each Attraction &trips = Count() print AttractionPB	✓ Grid: Grid2	
Attraction Id Attraction Name Trips	endfor	Control Name	Grid2
AttractionId AttractionName & & & & & & & & & & & & & & & & & & &		Collection	Attraction
	endfor	l Order	AttractionName
		Conditions	Attaction tante
		Unique	
			4

When you implement both grids with base table, which is the way to work less, you will be establishing the Country.City base transaction for the first grid, and Attraction for the second one.

Web form Rules Events Conditions Variables 1 ≥ parm(in: CountryId); Country Name CountryName GRID GRID GRID GRID GRID GRID Grid2.Refresh() Grid2.Load() Eiffel Tower Grid2.Load() Louvre Museum Grid1.Load() → Nice Grid2.Refresh() Grid2.Refresh() Grid2.Load() Nice Grid2.Refresh() Grid2.Load() Matisse Museum		GeneXu	S *
I⊫parm(in: CountryId); Grid1.Refresh() Grid1.Load() → Paris Grid2.Refresh() Grid2.Load() Eiffel Tower Grid2.Load() Louvre Museum Grid1.Load() → Nice Grid1.Load() → Nice Grid2.Refresh() Grid1.Load() → Mice	Web Form Rules Events Conditions Variables		
Country Name CountryName GRID Grid2.Load() GRID Grid2.Load() Attraction Id Attraction Name Attraction Id AttractionName AttractionId AttractionName Grid2.Load() Louvre Museum Grid1.Load() Nice Grid2.Refresh() Grid2.Refresh() Grid2.Load() Matisse Museum	1 =parm(in: CountryId);	Grid1.Refresh()	
GRID GRID GRID GRID GRID Grid2.Load() Eiffel Tower Grid2.Load() Louvre Museum Grid2.Load() Louvre Museum Grid2.Load() Mice Grid2.Load() Mice Grid2.Load() Mice Grid2.Load() Mice	Country Name Country Name	Grid1.Load() Paris	
GRID GRID Attraction Id Attraction Name AttractionId AttractionName City Name CityName GRID AttractionId AttractionName CityName CityName CityName CityName CityName CityName CityName CityName Grid2.Load() Eiffel Tower Grid2.Load() Louvre Museum Grid1.Load() → Nice Grid2.Refresh() Grid2.Load() Matisse Museum		Grid2 Refresh()	
City Name CityName Grid2.Load() Louvre Museum Grid2.Load() Nice Grid2.Refresh() Grid2.Load() Matisse Museum	GRID	Grid2.Load() Eiffel Tower	
GRID Attraction Id Attraction Name AttractionId Trips &trips &update2 &trips &update3 &trips &update3 &trips &update3 &trips &update3 &trips <td>City Name CityName</td> <td>Grid2.Load() Louvre Muse</td> <td>eum</td>	City Name CityName	Grid2.Load() Louvre Muse	eum
Attraction/a Attraction/Name (&trps) (&update2) (Anewithp) (Grid2.Refresh()) (Grid2.Load()) Matisse Museum	GRID Attraction Id Attraction Name Trips	Grid1.Load() Nice	
Grid2.Load() Matisse Museum	AttractionId AttractionName & Rew Irip	Grid2.Refresh()	
		Grid2.Load() Matisse Mus	eum

For any case, the Refresh event of Grid1, the external one, will take place first, followed by that grid's Load event N, once or N times depending on whether the grid has its base table or not.

In this case, since two cities (Paris and Nice) have been entered for France, you know that the external grid's first Load will be Paris, and immediately after, and prior to executing the Load again to load Nice, the Refresh event of the nested grid will take place. And its Load event will come immediately after, once or N times, depending on whether it has base table or not. It does have base table in this case, so a Load will be triggered to load the Eiffel Tower and another one to load the Louvre Museum.

Once the load of the nested grid is concluded, it will move on to load the following city, Nice. And it will be the same, triggering the Refresh event of the nested grid once, to then trigger the Load event N times for loading the new attractions, in Nice, which in this example is just one.

	GeneXus
Web Form Rules Events Conditions Variables	Event Start
ip parm(in. country id),	&newTrip = "New trip" &update2 = "UPDATE"
	CountryName.ForeColor = RGB(147,4,55) //DarkBase CountryName.FontBold = True
Country Name CountryName	CityName.FontBold = True
	Event Gridt Refnerb
City Name CityName	&totalAttractions = 0
	endevent
Attraction Id Attraction Name Trips	Event Grid1.Load <pre>&attractions = Count(AttractionName)</pre>
AttractionId AttractionName & & & & & & & & & & & & & & & & & & &	<pre>&totalAttractions = &totalAttractions + &attractions endevent</pre>
	Event Grid2.Refresh
Total Trips &totalTrips	&totalTrips = 0
	Event Grid2 Load
Total Attractions &totalAttractions	&trips = Count(TripDate)
	Endevent

Here, the variables are added to the screen to complete the rest you had previously. This will make the example identical, with the need for programming all the events in the system.

	GeneXus
Web Form Rules Events Conditions Variables 1 parm(in: CountryId);	Event Grid1.Refresh &totalAttractions = 0 endevent For each Country.Clty where CountryId = @CountryId
Country Name CountryName	Event Grid1.Load &attractions = Count(AttractionName) &totalAttractions = &totalAttractions + &attractions endevent
City Name CityName	Event Grid2.Refresh &totalTrips = 0 Endevent
Attraction Id Attraction Name AttractionId AttractionName AttractionRame Extrips & Barbar	For each Attraction order AttractionName where CountryId = @CountryId where CityId = @CityId
Total Trips &totalTrips Total Attractions &totalAttractions	Event Grid2.Load &trips = Count(TripDate) &totalTrips = &totalTrips + &trips Endevent
	Load endfor
	endfor

Translating this into a GeneXus pseudo-code would result in something like this.

First, the external grid's Refresh is executed. There, we set to zero the variable that will count the total number of attractions that will be loaded. Then, and because this is the case of a grid with the CountryCity base table, GeneXus will establish the implicit For each that will navigate this table, filtering by the value of Countryld received by parameter. For each record found, that grid's Load event will be executed. It will have that city's attractions, which it will add to the variable that will be totalized. Then the city is loaded on the grid, from the Load command that GeneXus includes.

Immediately after, the Refresh of the nested grid is executed, leaving in zero the value of the total sum of trips that include that attractions that are loaded afterwards. And because it has base table, GeneXus writes another implicit For each to navigate that base table –Attraction– to which it adds all the corresponding clauses, according to what the developer made **explicit in the grid's properties. In this case, you had only established base** transaction and order clause.

Additionally, it adds the implicit conditions that precisely relate to the fact that this grid is nested with another one, and a relation between the table exists. For this reason, it will only go through the records of the attractions

table that match the country and city of the record loaded in the external For each. And for each of them, it will execute the Load of this nested grid. And then it will load the line on the grid.

Web Form Rules Events Conditions Variables	Event Grid1.Refresh <u> &totalAttractions</u> = 0 endevent
Country Name &CountryName	Event Grid1.Load &attractions = Count(AttractionName) &totalAttractions = &totalAttractions + &attractions endevent
GRID City Name &cityName GRID GRID	Event Grid2.Refresh &totalTrips = 0 Endevent
Attraction Id AttractionName AttractionName AttractionName AttractionName AttractionName AttractionName	For each Attraction order AttractionName where CountryId = @CountryId where CityId = @CityId
Total Trips &totalTrips Total Attractions &totalAttractions	Event Grid2.Load &trips = Count(TripDate) &totalTrips = &totalTrips + &trips Endevent
	Load endfor

Of course, if the first grid had no base table, then the implicit For each disappears, as well as the Load command.

Web Form Rules Events Conditions Variables Events 1 parm(in: CountryId); end Country Name & CountryName end GRID city Name & CityName GRID attraction Id Attraction Id AttractionName Total Trips & cotalTrips	t Grid1.Refresh &totalAttractions = 0
Country Name &CountryName GRID City Name &CityName GRID Attraction Id Attraction Name Trips &update2 &newTrip Total Trips &totalTrips	vent
GRID City Name &cityName GRID Attraction Id Attraction Name AttractionName & Trips &trips &update2 &unewTrip Total Trips &totalTrips	t Grid1.Load For each Country.City &CountryName = CountryName &cityName = (ityName &attractions = Count(AttractionName) &totalAttractions = &totalAttractions + &attractions Load
City Name &cityName GRID Attraction Id Attraction Name AttractionId AttractionName Trips &trips &trips &trips &trips	endfor vent
Attraction Id Attraction Name AttractionId AttractionName Total Trips &totalTrips	Event Grid2.Refresh &totalTrips = 0 Endevent
Total Trips &totalTrips	For each Attraction order AttractionName
Total Trips &totalTrips	where CountryId = @CountryId
Total Trips &totalTrips	where CityId = @CityId
Total Attractions RitotalAttractions	Event Grid2.Load &trips = Count(TripDate) &totalTrips = &totalTrips + &trips Endevent
	Load
	endfor

And we will have to explicitly write them in the grid's Load event.

In this case, upon finding the Load command inside the grid's Load event because the grids are nested, GeneXus will immediately trigger the Refresh and Load events of the nested grid. And depending, again, on whether the nested grid has base table or not, GeneXus will either establish an implicit For each and its Load or not. Except that in this case, you will have to the explicit the Where of cities, which was implicit before, as you will see next.

Web Form Rules Events Conditions Variables 1 = parm(in: CountryId);	Event Grid1.Refresh &totalAttractions = 0 endevent Event Grid1.Load For each Country.City
Country Name &CountryName GRID City Name &CityName	&CountryName = CountryName &cityName = CityName &attractions = Count(AttractionName) &totalAttractions = &totalAttractions + &attractions Load endfor endevent Event Grid2.Refresh &totalTrips = 0
GRID Attraction Name Trips AttractionId AttractionName Trips &AttractionId &AttractionName Etrips Total Trips &totalAttractions	Endevent Event Grid2.Load For each Attraction order AttractionName where CityName = &cityName &AttractionId = AttractionId &AttractionName = AttractionName &AttractionPhoto = AttractionPhoto & trips = Count(TripDate) & dtotalTrips = &totalTrips + &trips Load endfor Endevent

If you wanted that the nested grid not have base table, it is clear that you would substitute attributes by variables in the grid, and you will have to program the For each explicitly in the Load event, as well as in the Load command.

And precisely because there are not base tables, leaving the logic for loading grids fully up to the developer, GeneXus may not establish the automatic join between the For eachs, and for this reason you need to explicit the filters.

You will be establish only the filter by CityName, because the filter by CountryId will be carried out due to the parameter.

		GeneXus
	Event Grid1.Load	*
Event Grid1.Refresh &totalAttractions = 0 endevent Event Grid1.Load For each Country.City &CountryName = CountryName &cityName = CityName &attractions = Count(AttractionName) &totalAttractions = &totalAttractions + &attractions Load endfor	For Each CountryCity (Line: 14) Order: CountryId Index: ICOUNTRYCITY Navigation filters: Start from: CountryId = @CountryId Join location: Server EcountryId = @CountryId Image: CountryCity (CountryId, CityId) Ecountry(City (CountryId, CityId)) Image: Country(AttractionName).navigation Education (CountryId, CityId)	*
Event Grid2.Refresh	Event Grid2.Load	*
Endevent	For Each Attraction (Line: 28)	*
Event Grid2.Load For each Attraction order AttractionName where CityName = &cityName &&AttractionId = AttractionId &AttractionName = AttractionPhoto &AttractionPhoto = AttractionPhoto &&trips = Count(TripDate) &&totalTrips = &totalTrips + &trips Load endfor Endevent	Order: <u>AttractionName</u> Navigation filters: Start from: FirstRecord Loop while: NotEndOfTable Constraints: <u>Countryld = @Countryld = @Countryld = @CiryName</u> Join location: Server	

Here, it is implemented in GeneXus.

If you take a look at its navigation list, you will clearly see that it did not select base table for any of the two grids.

And if you execute... you will not see any difference with the web panel that had base tables for both grids.

	Base Table	S
Country Name CountryName	?	Country Name &CountryName
City Name CityName		GRID City Name &city/Name
Attraction Id Attraction Name AttractionId AttractionName Image: Constraint of the structure Image: Constraint of the structure Image: Constraint of the structure Image: Constraint of the structure	?	Attraction Id Attraction Name & Trips & & & & & & & & & & & & & & & & & & &
Total Trips &totalTrips		Total Trips & totalTrips
Total Attractions &totalAttractions	. L	Total Attractions & & total Attractions

But to reach this navigations, you had to determine the base tables in the first place, just as it occurs with the For eachs.

Web Form Rules Events Conditions Variables 1 parm(in: CountryId); Country Name CountryName GRID City Name CityName	Ist GRID Attributes in the grid (visible or hidden) Grid Base Trn property Grid Order property Grid Conditions property Grid Unique property
GRID Attraction Id Attraction Name AttractionId AttractionName Returns & & & & & & & & & & & & & & & & & & &	 Grid Data Selector property Attributes in the grid's Load event (without context, ie: For each command and inline aggregate formula)
Total Attractions &totalAttractions	+
	• Fixed-part attributes

In order to determine the base table of the first grid on screen, the exact same thing that you saw for parallel grids will apply. That is to say that, GeneXus takes into account the attributes of the grid itself, plus those corresponding to the fixed part of the screen. In addition, of course, to any that may appear in the grid's properties (Base transaction, Order, etc.) and those of the Load event for the grid. But neither those of the Refresh of the grid, nor those of any other event.

	GeneXus
	Constral Class
	Filter
Web Form Bules Fuents Conditions Variables	Free Style Grid: Grid1
1 approx (in: Country Id.):	Control Name Grid1
Tip par int the counce year,	Collection
9	Rendering Mode Responsive
Country Name CountryName	Save State False
	Base Trn Country.City
GRID	Order
City Name CityName	Conditions
GRID	Unique
Attraction Id Attraction Name Trips AttractionId AttractionName ExtractionS	Attraction (a) Attractionid AttractionName CountryId CountryName
Total Trips &totalTrips Total Attractions &totalAttractions	AttractionDescripti Country/id AttractionPhoto City/d AttractionAddress
	Category CategoryName

When a base transaction has been specified, it will have that base table, and all the attributes mentioned will have to belong to its extended table.

		GeneXus*
	Properties	
	General Class	
	🔚 👌 🖌 Filter 🗙 🗙	
Web Form Pulse Events Conditions Variables	Free Style Grid: Grid1	
1 page (in: CountryId):	Control Name Grid1	
in parme in councilita);	Collection	
3	Rendering Mode Responsive	
Country Name CountryName	Save State False	
	Base Trn	
GRID	Order	
City Name CityName	Conditions	
GRID	Unique	
Attraction Id Attraction Name Trips		
AttractionId AttractionName & & & & & & & & & & & & & & & & & & &	Attraction (Country (S))	
	AttractionId CountryId	
	AttractionName AttractionDescripti	
Total Trips &totalTrips	CountryId	
	AttractionPhoto	
Total Attractions &totalAttractions	AttractionAddress CountryId CountryId CountryId CountryId CountryId	
	Cityld	Base table
	Category	
	CategoryId	
	CategoryName	

If no base transaction has been specified, then GeneXus will find the minimum extended table containing all the attributes mentioned and will then select its base table as the grid's base table.

Web Form Rules Events Conditions Variables 1 parm(in: CountryId);	Genexus Nested GRID
Country Name GRID City Name City City Name City City Name City City City City City City City City	 Attributes in the grid (visible or hidden) Grid Base Trn property Grid Order property Grid Conditions property Grid Unique property Grid Data Selector property Attributes in the grid's Load event (without context, ie: For each command and inline aggregate formula)

The base table of the nested grid is determined as if the grid were parallel and not nested, except that it will be the same as for determining the base table of a nested For each.

When the nested grid HAS NO BASE TRANSACTION SPECIFIED, then GeneXus will determine it on its own, and here is where the fact that this grid is nested with another one may determine a table different from the one that would be determined if the grid were parallel.

Grid Base Trn property	dden)
 City Name CityName Grid Order property Grid Conditions property Grid Unique property Grid Unique property Grid Data Selector property Attractionid Irrips & totalTrips 	mand

These cases are not frequent, but it's good to be aware of them.



For instance, if the grids were inverted and the external grid navigated the Attraction table, with no base transaction specified in the internal grid, with only the CityName attribute implied, and the grid being parallel, then it is clear that GeneXus would determine the cities table as its base table. However, in this case, and because it is nested with a grid that has an extended table that includes the attributes of the second grid, it will select for this second grid the same base table as that of the first grid, thus implementing a control break.

GeneXus

Base tables: ready!

And its navigations!

And this concludes the study of determining base tables and navigations for all cases of web panels.



training.genexus.com wiki.genexus.com