

# Web Screens

Base Table and Navigation of Web Panel with No Grid

*GeneXus™*

## ***Web Panel with No Grid***

Let's start with a web panel with no grid.

Web Form \* **Rules** \* Events | Conditions | Variables |

1 parm(in:&amp;AttractionId);

Web Form \* | Rules \* | Events | **Conditions** | Variables |

1 AttractionId = &amp;AttractionId;

- Attributes in the **form** (visible or hidden)
- Attributes in **events** (without context, i.e.: For each command; inline aggregate formula)

Event Load

`&trips = Count(TripDate)`

Endevent

Event CountryName.Click

`ViewCountryInfoSimple(CountryId)`

Endevent

Event Start

`CountryId.Visible = False``AttractionName.ForeColor = RGB(147,4,55) //DarkBase``AttractionName.FontBold = True`

Endevent

It is enough that there is some attribute in the form for GeneXus to understand that it has to search for its value in the database. This, of course, applies to both web panels with and without grids.

But what if there were no attributes in the form? Would this be enough for not establishing a base table?

The answer is no.

GeneXus will look for attributes in the form (visible or hidden) and in all events. But in the events it will only take into account the attributes that are "loose;" that is to say, attributes that do not have a specific navigation context that explains them, as would be the case of attributes within a For Each command, or within the calculation of an inline aggregation formula, for example, because the navigations for those attributes will be determined precisely by the For Each command or by the formula, respectively.

In this example we have three programmed events. The first one we see is the Load event, which, as we know, will be executed only once or n times depending on whether or not GeneXus finds a base table for the web panel. If we look at its programming, we have an attribute, TripDate, but it is in the context of an inline aggregation formula, so it does not participate in determining a base table for the panel, but only for the table to be run through by the formula. In fact, if the panel did not have a base table, what would happen is that the Load event would be executed only once, and there the formula, which would count all the Trips records (TripDate is a

secondary attribute of the Trip transaction).

However, the situation of the other two events we see is different, because there are attributes appearing without any other context to explain them except that it is assumed that **there is a base table** and that is its context! They are not within inline formulas, neither within a For Each command nor any other structure that establishes a context for them. They will be attributes of the extended table of the web panel's base table.

So, even if we had not specified any attribute in the form, by the mere presence of these attributes here, GeneXus will already infer a base table (and of course, its extended table).

Before continuing, let's remember that neither the attributes that may appear in the parm rule nor those that may be specified in the web panel conditions will participate. Therefore, it is not the appearance of this AttractionId that we see in the conditions that will make the panel have a base table. We will see this better in a moment.

Web Form \* **Rules** \* Events | Conditions | Variables

1 parm(in:&amp;AttractionId);

Web Form \* Rules \* Events **Conditions** Variables

1 AttractionId = &amp;AttractionId;

The screenshot shows a web form with the following fields and a photo placeholder:

- Name: &AttractionName
- Country Id: &CountryId
- Country Name: &CountryName
- City Name: &CityName
- Category Name: &CategoryName
- Trips: &trips

There is a photo placeholder on the right side of the form, showing a green mountain and a blue sky.

- Attributes in the **form** (visible or hidden)
- Attributes in **events** (without context, i.e.: For each command and inline aggregate formula)

## Event Load

For each

```
&AttractionName = AttractionName
&CountryId = CountryId
&CountryName = CountryName
&CityName = CityName
&CategoryName = CategoryName
&AttractionPhoto = AttractionPhoto
&trips = Count(TripDate)
```

endfor

Endevent

Event &amp;CountryName.Click

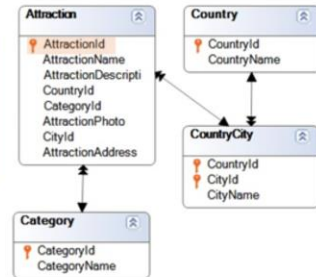
ViewCountryInfoSimple(CountryId)

Endevent

Event Start

```
&CountryId.Visible = False
&AttractionName.ForeColor = RGB(147,4,55) //DarkBase
&AttractionName.FontBold = True
```

Endevent



Imagine, for example, that we made a Save as of the previous panel, because we wanted to transform it into one without a base table. Then we change in the form all the attributes to variables, which are the ones that now in the Start event we must set as invisible or change their color or put them in bold. Besides, since we precisely want it not to have a base table, we should program a For Each command in the Load event, because if everything goes as we think, it will be executed only once and that is where we should load the variable values. In this case, we won't need to write a Load **command**, because we don't have to add lines to a grid. The variables will be loaded on the screen with the values they have at the time of loading.

We have not specified a base transaction for the For Each command, so GeneXus extracts all its attributes and determines the **minimum extended table that contains them all**, which is the Attraction table. And since we have a condition that includes an attribute from that table, then it applies it. Everything will work perfectly.


In the Load event, the For Each command will retrieve a single record, that of the AttractionId attraction.

Web Form \* **Rules** \* Events | Conditions | Variables

1 parm(in:&amp;AttractionId);

Web Form \* Rules \* Events **Conditions** Variables

1 AttractionId = &amp;AttractionId;

Name <input type="text" value=" &amp;AttractionName"/>	
Country Id <input type="text" value=" &amp;CountryId"/>	
Country Name <input type="text" value=" &amp;CountryName"/>	
City Name <input type="text" value=" &amp;CityName"/>	
Category Name <input type="text" value=" &amp;CategoryName"/>	
Trips <input type="text" value=" &amp;trips"/>	

## Web Panel ViewAttractionFromScratch2 Navigation Report

Name: ViewAttractionFromScratch2

Description: View Attraction From Scratch2

Environment: C# Default (C#)

Spec. Version: 17\_0\_0-144971

Form Class: HTML

Program Name: ViewAttractionFromScratch2

Parameters: in: &amp;AttractionId

## Event Load

## For First Attraction (Line: 2)

Order: AttractionId

Index: IATTRACTION

Navigation Start from: AttractionId = &amp;AttractionId

filters: Loop while: AttractionId = &amp;AttractionId

Join location: Server

```

-Attraction ( AttractionId )
  -Country ( CountryId )
  -CountryCity ( CountryId, CityId )
  -Category ( CategoryId )
  -count( TripDate ).navigation
    -TripAttraction ( AttractionId )
      -Trip ( TripId )

```

The navigation list that we would expect for our web panel without a base table would be this one, where we see that inside the Load event the For Each command is being specified for the Attraction table (and note that it is being correctly filtered by AttractionId, equal to the value of the variable received by parameter).

Because it is the primary key of the table, the For Each command will retrieve a single record, and because it is ordered by that attribute, it will be the first (and only one). That's why instead of saying For Each, here it says For First.

Web Form \* **Rules \*** Events Conditions Variables

1 parm(in:&amp;AttractionId);

Web Form \* Rules \* Events **Conditions** Variables

1 AttractionId = &amp;AttractionId;

## Web Panel ViewAttractionFromScratch2 Navigation Report

Name: ViewAttractionFromScratch2

Description: View Attraction From Scratch2

Environment: Default (C#)

Spec. Version: 17\_0\_0-144971

Form Class: HTML

Program Name: ViewAttractionFromScratch2

Parameters: in: &amp;AttractionId

## Event Load

Order: CountryId  
 Index: ICOUNTRY  
 Navigation Start from: FirstRecord  
 filters: Loop while: NotEndOfTable



Country ( CountryId )

## For First Attraction (Line: 2)

Order: AttractionId  
 Index: IATTRACTION  
 Navigation Start from: AttractionId = &AttractionId  
 filters: Loop while: AttractionId = &AttractionId  
 Constraints: CountryId = @CountryId  
 Join location: Server

Attraction ( AttractionId )  
 CountryCity ( CountryId, CityId )  
 Category ( CategoryId )  
 count( TripDate ) navigation  
 TripAttraction ( AttractionId )  
 Trip ( TripId )

However, if we check the navigation in GeneXus, it will show this other report, in which it obviously found a base table. We see it clearly in the Load event. It is the Country table.

Note that it will run through the entire table, and will run one Load event per country. And for each one, it will run the For Each command.

What will we see on the screen then? We will not notice the problem at runtime, because the assignments to the screen variables will only be executed when the Load for the **country of the attraction** is being executed, as it is the only case in which the body of the For Each will be executed.

We've been lucky. However, we have incorrectly programmed our web panel. Why did it find a base table?

Web Form \* **Rules \*** Events Conditions Variables

1 parm(in:&amp;AttractionId);

Web Form \* Rules \* Events **Conditions** Variables

1 AttractionId = &amp;AttractionId;

The screenshot shows a web form design tool interface. The form is titled "Name" and contains several input fields: "&AttractionName", "Country Id &CountryId", "Country Name &CountryName", "City Name &CityName", "Category Name &CategoryName", and "Trips &trips". To the right of the form is a placeholder for a photo, represented by a small image icon of a landscape with a green hill and a blue sky with a sun.

- Attributes in the **form** (visible or hidden)
- Attributes in **events** (without context, i.e.: For each command and inline aggregate formula)

Event Load

For each

```
&AttractionName = AttractionName
&CountryId = CountryId
&CountryName = CountryName
&CityName = CityName
&CategoryName = CategoryName
&AttractionPhoto = AttractionPhoto
&trips = Count(TripDate)
```

endfor

Endevent

Event &amp;CountryName.Click

ViewCountryInfoSimple(CountryId)

Endevent

Event Start

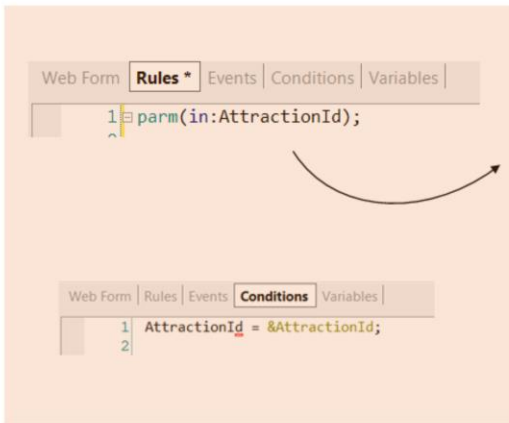
```
&CountryId.Visible = False
&AttractionName.ForeColor = RGB(147,4,55) //DarkBase
&AttractionName.FontBold = True
```

Endevent

Because we missed this attribute! What is it doing there, when the &CountryId variable should be in its place?

Precisely, when GeneXus is going to determine the navigation of the web panel, by finding this attribute without context within an event, it understands that it is then implicitly being told that there is a context: that of the web panel's base table! And since it is the only attribute in that situation, it is the only one that will be considered in this case to determine what that base table will be. That's why it selects the Country table.





Only **after** the base table is determined!

### Web Panel ViewAttractionFromScratch2 Navigation Report

<b>Name:</b> ViewAttractionFromScratch2	<b>Environment:</b> Default (C#)
<b>Description:</b> View Attraction From Scratch2	<b>Spec. Version:</b> 17_0_0-144971
	<b>Form Class:</b> HTML
	<b>Program Name:</b> ViewAttractionFromScratch2
	<b>Parameters:</b> in: AttractionId

Event Load

For First Attraction (Line: 2)

Order: AttractionId  
Index: IATTRACTION

Navigation filters: Start from: AttractionId = @AttractionId  
Loop while: AttractionId = @AttractionId

Join location: Server

-Attraction ( AttractionId ) INTO CityId CategoryId AttractionPhoto.Uri  
 AttractionName CountryId AttractionPhoto  
 -Country ( CountryId ) INTO CountryName  
 -CountryCity ( CountryId CityId ) INTO CityName  
 -Category ( CategoryId ) INTO CategoryName  
 -count( TripDate )navigation ( AttractionId )

Formulas

Navigation to evaluate: count( TripDate )

Given: AttractionId  
Index: IATTRACTION  
Group by: AttractionId

-TripAttraction

Here we see the navigation list according to that programming, and we are going to correct the error. We will replace the attribute with the variable, &CountryId.

Now we save. And let's see the new navigation list, which now is indicating what we expected; i.e.: the Load event has no base table and within it the Attraction table is run through with the For Each command.

In summary, we managed to ensure that our web panel has no base table. We can see how it is true, then, that the attributes that appear in the conditions are not going to participate in the determination of a base table for the web panel, because otherwise this attribute should have determined a base table and it did not.

Likewise, if instead of receiving in the variable we receive in the attribute, note what happens with the navigation. It is still a navigation for a panel without a base table.

The conditions –either implicit because they are received in attribute in the parm rule, or explicit because they are specified in the panel conditions– are taken into account to filter the data **ONLY AFTER** the table to be navigated and the tables of the extended table to be accessed are determined. It is only there, if necessary, that these filters are applied.

# GeneXus™

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