

Pantallas web

Tabla base y navegación en Web panel sin grids

GeneXus[™]

Web Panel sin Grid

Empecemos por un web panel sin grid.

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

1 parm(in:&AttractionId);

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

1 AttractionId = &AttractionId;

- Attributes in the **form** (visible or hidden)
- Attributes in **events** (without context, ie: 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**

Alcanza con que haya algún atributo en el form para que GeneXus entienda que tiene que ir a buscar su valor a la base de datos. Esto, evidentemente, aplica tanto a web panels sin grids como con grids.

¿Pero qué pasaría si no hubiera ningún atributo en el form?

¿Esto ya provocaría que no se establezca tabla base?

La respuesta es no.

GeneXus va a buscar atributos tanto en el form (visibles u ocultos) como en todos los eventos. Pero en lo eventos solo tomará en cuenta los atributos que estén **“suelto”** por decirlo de algún modo, es decir, atributos que no tengan un contexto determinado de navegación que los explique, como sería el caso de atributos dentro de un for each, o dentro del cálculo de una fórmula inline de agregación, por ejemplo, porque las navegaciones para esos atributos estarán determinadas justamente por el comando for each o por la fórmula, respectivamente.

En este ejemplo tenemos tres eventos programados. El primero que vemos es el evento Load, que, como sabemos, se ejecutará una sola vez o n dependiendo de si GeneXus encuentra o no tabla base para el web panel. Si observamos su programación, tenemos un atributo, TripDate, pero está en un contexto de una fórmula inline de agregación, por lo que no

participa en la determinación de una tabla base para el panel, sino solo para la tabla a ser recorrida por la fórmula. De hecho, si el panel no llegara a tener tabla base, lo que sucedería es que el evento Load se ejecutaría una sola vez, y allí la fórmula, que contaría todos los registros de Trips (TripDate es un atributo secundario de la transacción Trip).

Sin embargo, la situación de los otros dos eventos que vemos es distinta, porque allí sí están apareciendo atributos sin otro contexto que los explique que no sea el de que ¿se está asumiendo que hay una tabla base y ese es su contexto! No están dentro de fórmulas inline, ni de for each, ni de ninguna otra estructura que establezca un contexto para ellos. Por lo que serán atributos de la tabla extendida de la tabla base del web panel.

Así es que, aunque no hubiéramos especificado ningún atributo en el form, por la sola presencia de estos atributos aquí GeneXus ya inferirá una tabla base (y claro, su extendida).

Antes de continuar, recordemos que ni los atributos que puedan aparecer en la regla parm ni los que puedan especificarse en las conditions del web panel van a participar. Así que no es la aparición de este AttractionId que vemos en las conditions el que va a hacer que el panel tenga tabla base. Lo veremos mejor en un instante.

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

```
1 parm(in:&AttractionId);
```

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

```
1 AttractionId = &AttractionId;
```

The screenshot shows a web form layout with the following elements:

- A top section with a label "Name" and an input field containing "&AttractionName".
- A middle section with a label "Country Id" and an input field containing "&CountryId", and a photo placeholder icon.
- A section with a label "Country Name" and an input field containing "&CountryName".
- A section with a label "City Name" and an input field containing "&CityName".
- A section with a label "Category Name" and an input field containing "&CategoryName".
- A bottom section with a label "Trips" and an input field containing "&trips".

- Attributes in the **form** (visible or hidden)
- Attributes in **events** (without context, ie: 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 &CountryName.Click

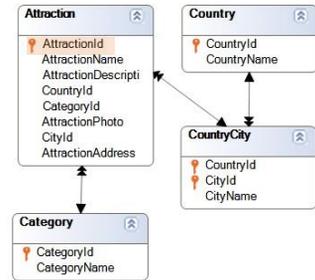
```
ViewCountryInfoSimple(CountryId)
```

Endevent

Event Start

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

Endevent



Imaginemos, por ejemplo, que hicimos un Save as del panel anterior, porque queríamos transformarlo en uno sin tabla base. Entonces cambiamos en el form todos los atributos por variables, que son las que ahora en el evento Start debemos colocar como invisibles o cambiarles el color o ponerlas en negrita. Además, como, justamente, estamos queriendo que no tenga tabla base, debemos programar un for each en el evento Load, ya que si todo va como lo pensamos, se ejecutará una única vez y allí es donde debemos cargar los valores de las variables. En este caso no necesitaremos escribir comando Load, puesto que no tenemos que agregar líneas a un grid. Las variables se cargarán en la pantalla con los valores que tengan en el momento de la carga.

Al for each no le hemos especificado transacción base, por lo que GeneXus extrae todos sus atributos y determina la mínima tabla extendida que los contiene a todos, que es la de la tabla Attraction. Y como tenemos una condition que incluye a un atributo de esa tabla, entonces la aplica. Todo funcionará perfecto.

En el evento Load el for each recuperará un único registro, el de la atracción AttractionId.

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

```
1 parm(in:&AttractionId);
```

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

```
1 AttractionId = &AttractionId;
```

The screenshot shows a web form layout with several input fields and a central image. The fields are arranged in a grid-like structure:

- Top row: Name &AttractionName
- Second row: Country Id &CountryId
- Third row: Country Name &CountryName
- Fourth row: City Name &CityName
- Fifth row: Category Name &CategoryName
- Sixth row: Trips &trips

In the center-right of the form, there is a landscape image showing a green mountain range under a blue sky with a yellow sun.

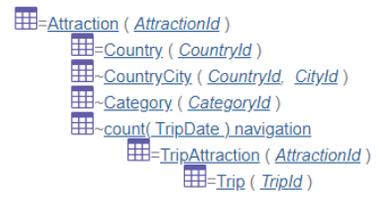
Web Panel ViewAttractionFromScratch2 Navigation Report

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

Event Load

For First Attraction (Line: 2)

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



El listado de navegación que esperaríamos para nuestro web panel sin tabla base sería este, donde vemos que dentro del Load se está especificando el For each a la tabla Attraction (y observemos que se está filtrando correctamente por AttractionId igual al valor de la variable recibida por parámetro).

Por ser la clave primaria de la tabla es que el for each recuperará un solo registro, y por estarse ordenando por ese atributo, será el primero (y único). Por eso en lugar de decir For each, aquí dice For First.

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

```
1 parm(in:&AttractionId);
```

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

```
1 AttractionId = &AttractionId;
```

Web Panel ViewAttractionFromScratch2 Navigation Report

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

Sin embargo, si pedimos a GeneXus para ver la navegaci3n, nos mostrar4 este otro reporte, en el que, evidentemente, encontr3 una tabla base. La vemos claramente en el evento Load. Es la tabla Country.

Observemos que va a recorrer toda la tabla, y va a ejecutar un evento Load por cada pa3s, y para cada uno, va a ejecutar el For each.

¿Qu3 veremos en la pantalla entonces? No nos daremos cuenta en ejecuci3n del problema, porque solamente van a ejecutarse las asignaciones a las variables de pantalla cuando se est3 ejecutando el Load para el pa3s de la atracci3n, porque es el 3nico caso en que se ejecutar4 el cuerpo del For each.

Hemos tenido suerte. Pero hemos programado mal nuestro web panel. ¿Por qu3 encontr3 tabla base?

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

1 parm(in:&AttractionId);

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

1 AttractionId = &AttractionId;

- Attributes in the **form** (visible or hidden)
- Attributes in **events** (without context, ie: 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 &CountryName.Click
  ViewCountryInfoSimple(CountryId)
```

Endevent

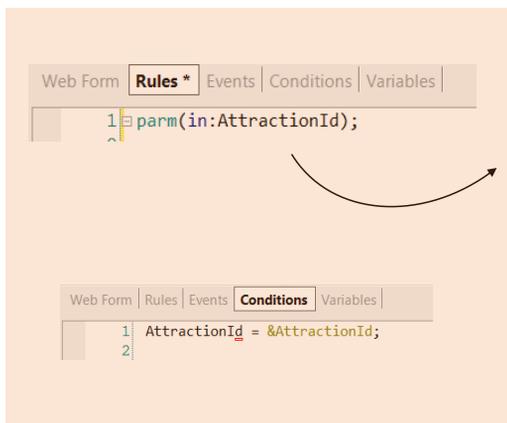
Event Start

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

Endevent

¡Porque se nos pasó por alto este atributo! ¿Qué hace allí, cuando tendría que ser la variable &CountryId la que estuviera en su lugar?

Justamente, cuando GeneXus va a determinar la navegación del web panel, al encontrar este atributo sin contexto dentro de un evento, entiende que entonces se le está diciendo implícitamente que sí hay un contexto: ¡el de la tabla base del web panel! Y como es el único atributo en esa situación, es el único que en este caso se va a considerar para determinar cuál será esa tabla base. Por eso está eligiendo la tabla Country.



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

Web Panel ViewAttractionFromScratch2 Navigation Report

Name: ViewAttractionFromScratch2	Environment: CH# Default (C#)
Description: View Attraction From Scratch2	Spec. Version: 17_0_0-144971
	Form Class: HTML
	Program Name: ViewAttractionFromScratch2
	Parameters: 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](#)

Attraction

Aquí vemos el listado de navegación de acuerdo a esa programación, y vamos a subsanar el error. Vamos a sustituir el atributo por la variable, &CountryId.

Ahora grabamos. Y veamos el nuevo listado de navegación, que ahora sí, está indicando lo que esperábamos, es decir: el evento Load no tiene tabla base y dentro del mismo es que se recorre la tabla Attraction con el For each.

En definitiva, entonces, conseguimos que nuestro web panel no tuviera tabla base. Podemos ver cómo es cierto, entonces, que los atributos que aparezcan en las conditions no van a participar en la determinación de una tabla base para el web panel, porque de lo contrario este atributo tendría que haber determinado tabla base y no lo hizo.

Lo mismo, si en lugar de recibir en la variable recibiéramos en el atributo, observemos qué pasa con la navegación. Sigue siendo una navegación para un panel sin tabla base.

Las condiciones (ya sean implícitas por recibirse en atributo en la regla parm) o explícitas por estar especificadas en las conditions del panel son tomadas en cuenta para filtrar los datos SOLO LUEGO de que la tabla a ser navegada y las tablas de la extendida a ser accedidas son determinadas. Recién allí, si corresponde, se aplican esos filtros.

*GeneXus*TM

training.genexus.com
wiki.genexus.com