

Telas web

Tabela base e navegação em Web panel sem grids

GeneXus™

Web Panel with NO Grid

Comecemos com um web panel sem 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

Basta que haja algum atributo no form para que GeneXus entenda que tem que ir buscar seu valor na base de dados. Isto, evidentemente, aplica-se tanto a web panels sem grids como com grids.

Mas o que aconteceria se não houvesse nenhum atributo no form? Isto faria com que não se estabeleça tabela base?

A resposta é não.

GeneXus vai procurar atributos tanto no form (visíveis ou ocultos) como em todos os eventos. Mas nos eventos só levará em conta os atributos que estejam “soltos” por assim dizer, ou seja, atributos que não tenham um contexto determinado de navegação que os explique, como seria o caso de atributos dentro de um for each, ou dentro do cálculo de uma fórmula inline de agregação, por exemplo, porque as navegações para esses atributos serão determinadas justamente pelo comando for each ou pela fórmula, respectivamente.

Neste exemplo temos três eventos programados. O primeiro que vemos é o evento Load, que, como sabemos, se executará uma única vez ou n, dependendo se GeneXus encontra ou não tabela base para o web panel. Se observamos sua programação, temos um atributo, TripDate, mas está em um contexto de uma fórmula inline de agregação, por isso não participa na determinação de uma tabela base para o panel, somente para a tabela a ser percorrida pela fórmula. De fato, se o panel não chegar a ter tabela base, o que aconteceria é que o evento Load se executaria uma única vez, e ali a fórmula, que contaria todos os registros de Trips (TripDate é um atributo secundário da transação Trip).

No entanto, a situação dos outros dois eventos que vemos é diferente, porque ali sim estão aparecendo atributos sem outro contexto que os explique, que não seja o de que se está assumindo que **há uma tabela base** e **esse** é seu contexto! Não estão dentro de fórmulas inline, nem de for each, nem de nenhuma outra estrutura que estabeleça um contexto para eles. Então, serão atributos da tabela estendida da tabela base do web panel.

Então, mesmo que não tivéssemos especificado nenhum atributo no form, pela simples presença destes atributos aqui, GeneXus já irá inferir uma tabela base (e claro, sua estendida).

Antes de continuar, recordemos que nem os atributos que podem aparecer na regra Parm nem os que podem ser especificados nas conditions do web panel vão participar. Então, não é o aparecimento deste AttractionId que vemos nas conditions que vai fazer com que o panel tenha tabela base. Veremos melhor em um 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 fields and elements:

- Name: &AttractionName
- Country Id: &CountryId
- Country Name: &CountryName
- City Name: &CityName
- Category Name: &CategoryName
- Trips: &trips
- A photo placeholder icon on the right side.

- 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

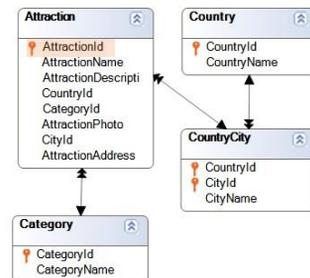
Event Start

&CountryId.Visible = False

&AttractionName.ForeColor = RGB(147,4,55) //DarkBase

&AttractionName.FontBold = True

Endevent



Imaginemos, por exemplo, que fizemos um Save as do panel anterior, porque queríamos transformá-lo em um sem tabela base. Então, mudamos no form todos os atributos por variáveis, que são as que agora no evento Start devemos colocar como invisíveis, mudar a cor ou colocá-las em negrito. Além disso, como estamos justamente querendo que não tenha tabela base, devemos programar um for each no evento Load, já que se tudo segue como pensamos, se executará uma única vez, e aí é onde devemos carregar os valores das variáveis. Neste caso, nós não precisaremos escrever o **comando** Load, uma vez que não temos que adicionar linhas a um grid. As variáveis serão carregadas na tela com os valores que tenham no momento da carga.

Ao for each não especificamos transação base, uma vez que GeneXus extrai todos os seus atributos e determina a **mínima tabela estendida que contém a todos**, que é a da tabela Attraction. E como temos uma condition que inclui um atributo dessa tabela, então a aplica. Tudo funcionará perfeitamente.

No evento Load o for each recuperará um único registro, o da atração AttractionId.

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

1 parm(in:&AttractionId);

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

1 AttractionId = &AttractionId;

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

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

```

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

A lista de navegação que esperaríamos para nosso web panel sem tabela base seria esta, onde vemos que dentro do Load está se especificando o For each para a tabela Attraction (e observemos que está sendo filtrado corretamente por AttractionId igual ao valor da variável recebida por parâmetro).

Por ser a chave primária da tabela, é que o for each recuperará um único registro, e por estar se ordenando por esse atributo, será o primeiro (e único). Por isso, em vez de dizer **For each**, aqui diz **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

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

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)

Entretanto, se pedimos a GeneXus para ver a navegação, nos mostrará este outro relatório, no qual, evidentemente, encontrou uma tabela base. A vemos claramente no evento Load. É a tabela Country.

Observemos que vai percorrer toda a tabela e vai executar um evento Load para cada país. E para cada um, vai executar o For each.

O que veremos na tela então? Não perceberemos em execução o problema, porque somente serão executadas as atribuições às variáveis de tela quando estiver executando o Load para **o país da atração**, porque é o único caso em que se executará o corpo do For each.

Tivemos sorte. Mas programamos errado o nosso web panel. Por que encontrou tabela base?

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 fields and controls:

- Name: &AttractionName
- Country Id: &CountryId
- Country Name: &CountryName
- City Name: &CityName
- Category Name: &CategoryName
- Trips: &trips
- A photo placeholder icon on the right side.

- 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

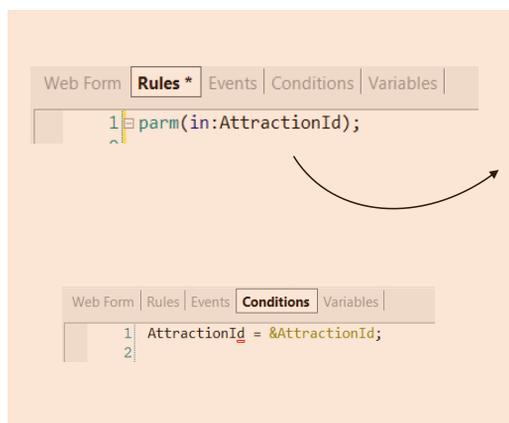
Event Start

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

Endevent

Porque ignoramos este atributo! O que ele faz ali, quando teria que ser a variável &CountryId que estivesse em seu lugar?

Justamente, quando GeneXus vai determinar a navegação do web panel, ao encontrar este atributo sem contexto dentro de um evento, entende que então está lhe dizendo implicitamente que sim, há um contexto: o da tabela base do web panel! E como é o único atributo nessa situação, é o único que neste caso vai se considerar para determinar qual será essa tabela base. Por isso está escolhendo a tabela 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](#)

[TripAttraction](#)

Aqui vemos a lista de navegação de acordo com essa programação, e vamos corrigir o erro. Vamos substituir o atributo pela variável, &CountryId.

Agora gravamos. E vejamos a nova lista de navegação que, agora sim, está indicando o que esperávamos, ou seja: o evento Load não tem tabela base e dentro do mesmo é que se percorre a tabela Attraction com o For each.

Em suma, então, conseguimos que nosso web panel não tivesse tabela base. Podemos ver como é certo, então, que os atributos que aparecem nas conditions não vão participar na determinação de uma tabela base para o web panel, porque, caso contrário, este atributo teria que ter determinado tabela base e não o fez.

O mesmo se, em vez de receber na variável, recebêssemos no atributo, observemos o que acontece com a navegação. Ainda é uma navegação para um panel sem tabela base.

As condições (sejam elas implícitas por serem recebidas em atributo na regra parm) ou explícitas por estarem especificadas nas conditions do panel são levadas em conta para filtrar os dados SOMENTE DEPOIS de que a tabela a ser navegada e as tabelas da estendida a serem acessadas, são determinadas. Apenas ali, se corresponde, aplicam-se esses filtros.

GeneXus™

training.genexus.com

wiki.genexus.com

training.genexus.com/certifications