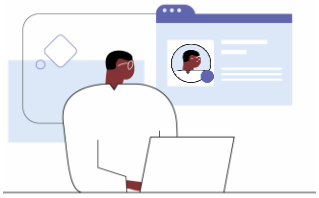
Pantallas web con foco en Customer-facing

Esquema de ejecución de eventos, gramática de eventos del cliente y determinación de tablas base

GeneXus™

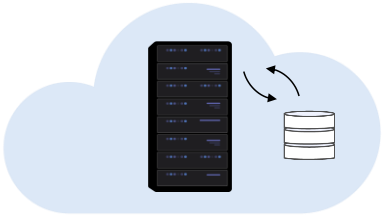
En este video comenzaremos a ver los eventos que podemos definir en las pantallas con foco en aplicaciones Customer-facing, en particular en las pantallas web y también nos detendremos en algunas características propias del objeto panel.

Eventos del cliente y del servidor



CLIENTE WEB con
foco en UX

- ClientStart
- Back
- User Events
- Control Events



SERVIDOR

- Start
- Refresh
- Load

Este tipo de aplicaciones incluyen dos tipos de eventos, los eventos que se disparan del lado del cliente y los eventos que se disparan en el servidor.

Los eventos del servidor son los mismos que ya vimos cuando estudiamos webpanels, los eventos Start, Refresh y Load, que nos permiten interactuar con la base de datos.
Los eventos del lado del cliente son el ClientStart, el Back, eventos definidos por el usuario y eventos asociados a los controles de la pantalla.

En el caso de que los objetos panel los usemos para generar una aplicación para dispositivos móviles, tendremos también otros eventos asociados a la plataforma móvil, por ejemplo relacionados al tipo de dispositivo y su orientación al ejecutarse la aplicación.

Eventos del lado del servidor

SERVIDOR

- Start
- Refresh
- Load

- Start Event execute only once.
- Refresh Event is executed after Start Event
- Refresh Event trigger Load Event
- Load Event is the last of system Events executed (only if a grid exist)
  - Grid with Base Table: Load is executed as many times as records in base Table.
  - Grids without Base Table: Load is executed only once.
  - SDT based Grids: Load is not triggered.

Veremos ahora los eventos del lado del servidor.

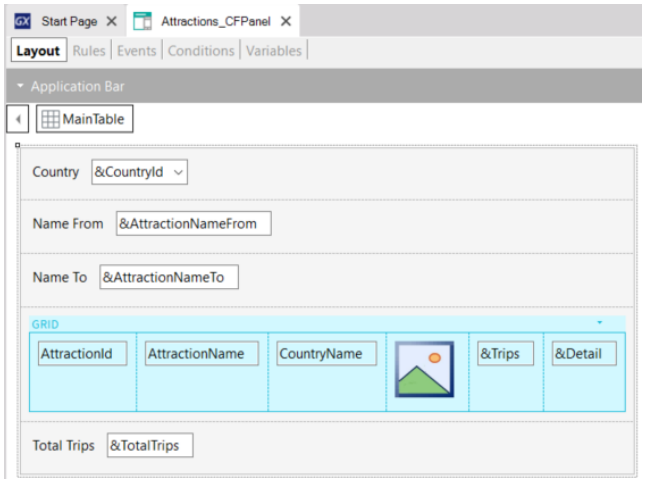
- El primero que se ejecuta es el evento Start. Se ejecuta solo una vez, cuando se abre el panel y no se ejecutará nuevamente a menos que salgamos del objeto y volvamos a ingresar en él.

- El evento Refresh se ejecuta luego del Evento Start, normalmente una sola vez, pero puede ser invocado nuevamente por medio del comando Refresh, en ese caso se ejecutara más de una vez y será el primer evento, ya que Start ya no se volverá a ejecutar.

- Cuando se invoque al evento Refresh, al finalizar se disparará el evento Load. Esto es solamente si hay al menos un grid en el panel y el grid no es una variable SDT colección.

- El evento load es el último de los eventos del sistema a ser ejecutados y
  - Si tiene Tabla Base se ejecutará tantas veces como registros existan en la tabla base.
  - Si la grilla no tiene tabla base se ejecutará solo una vez
  - Y si la grilla esta basada en un SDT no se ejecutará el evento load.

Ejemplo de eventos del lado del servidor




```
Event Load
    &Trips = Count(TripDate)
Endevent

Event Refresh
    &TotalTrips = 0
    For each Trip.Attraction
        &TotalTrips += 1
    Endfor
Endevent

Event &CountryId.ControlValueChanged
    Grid1.Refresh()
Endevent

Event &AttractionNameFrom.ControlValueChanged
    Grid1.Refresh()
Endevent


Event &AttractionNameTo.ControlValueChanged
    Grid1.Refresh()
Endevent

Event Start
    &Details = "Details"
Endevent

Event &Details.Tap
    AttractionDetail_CFPanel.Call(AttractionId)
Endevent
```

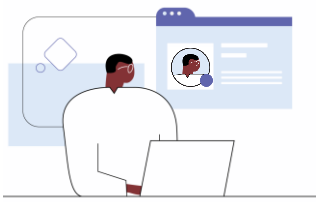
En el ejemplo del video anterior, cuando implementamos el objeto panel Attractions_CFPanel, programamos los eventos Start, Refresh y Load.

En el evento Load calculamos el total de viajes de una atracción mediante una fórmula que accedía a la tabla TripAttraction.

En el evento Refresh, programamos un For Each para que acceda a la tabla TripAttraction para contar el total global de viajes de las atracciones.

Y también programamos el evento Start para inicializar la variable &Details con el texto que queríamos que apareciera en pantalla.

Eventos del lado del cliente



CLIENTE WEB con foco en customer-facing

- ClientStart
- Back
- User Events
- Control Events

- Types:
  - System Events: ClientStart, Back
  - User Events: User Defined
  - Control Events: Predefined for each control
- Call to services to access server resources
- Do not trigger Server Side Events (unless Refresh command is used)
- Use different Grammar

Los eventos del lado del Cliente son la respuesta de la aplicación a la interacción del usuario.

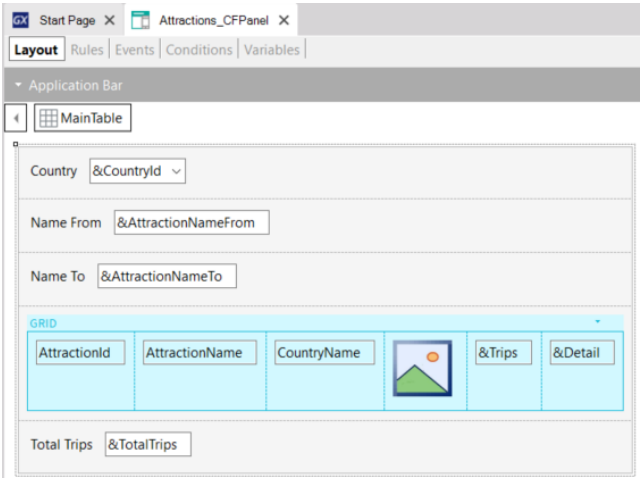
- Hay tres tipos de eventos de cliente: del sistema, de usuario y eventos de controles. Los eventos del sistema son el ClientStart y el Back, que no veremos en este curso. Los eventos de usuario son los creados por el usuario y los eventos de los controles son los asociados a los controles de la pantalla, como los eventos Tap (que corresponde al clic), Double Tap (que es un doble clic) o ControlValueChanged, entre otros

- El código asociado a estos eventos se ejecuta en el clliente, a menos que se requiera acceder a un servicio del servidor, por ejemplo si se quiere acceder a la base de datos

- Durante la ejecución de un evento del lado del cliente, los eventos del lado del servidor no se ejecutan, a menos que se requieran explícitamente a través del comando Refresh.

- Estos eventos del lado del cliente, tendrán una gramática particular diferente a los eventos del lado del servidor.

Ejemplos de eventos del lado del cliente




```
Event Load
    &Trips = Count(TripDate)
Endevent

Event Refresh
    &TotalTrips = 0
    For each Trip.Attraction
        &TotalTrips += 1
    Endfor
Endevent

Event &CountryId.ControlValueChanged
    Grid1.Refresh()
Endevent

Event &AttractionNameFrom.ControlValueChanged
    Grid1.Refresh()
Endevent


Event &AttractionNameTo.ControlValueChanged
    Grid1.Refresh()
Endevent

Event Start
    &Details = "Details"
Endevent

Event &Details.Tap
    AttractionDetail_CFPanel.Call(AttractionId)
Endevent
```

En el mismo objeto que vimos antes, programamos solamente eventos asociados a controles, no programamos ningún evento de usuario.
A cada variable de los filtros en pantalla, le programamos su evento ControlValueChanged, para disparar al método Refresh del Grid, lo que provocará que se disparen los eventos Refresh y Load del servidor para actualizar el contenido de la grilla con los filtros ingresados.

También programamos el evento Tap de la variable &Details, para invocar al panel AttactionDetail_CFPanel, para ver el detalle de la atracción seleccionada, cuyo identificador AttractionId pasamos por parámetro.

Gramática de eventos del lado del cliente

COMMANDS

Composite

<Control>.<Property> = *<value>*

If **<Bool_expr>**
Do case... endcase
Do while <Bool_expr>
Do-sub (except Menu for Smart Devices)

For each selected line

Simple variable assignation: &var = ***<expr>***

SDT or BC elements assignation:
&SDT.A = <value>
&BC.A = <value>

Return
Refresh

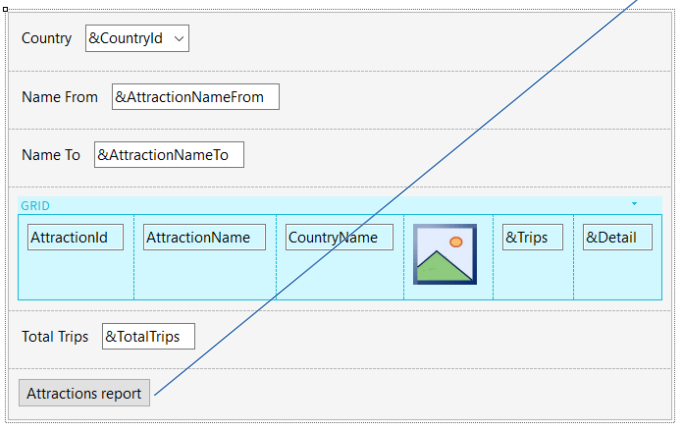
Inside an **expression**:
- Variables
- Attributes
- Constants
- Methods
- Functions
- Control properties
- Operators (+, -,, /, ^)

Veamos ahora la gramática de eventos del lado del cliente.

En los eventos del lado del cliente, hay restricciones en cuanto a los comandos que se pueden utilizar, no así para los eventos de lado del servidor.

Los comandos aceptados por el momento son los que se muestran. No entraremos en más detalle en este curso,

Comando Composite




```
Event "Attractions report"
    Composite
        AttractionsByName("C", "M")
        Return
    EndComposite
Endevent
```

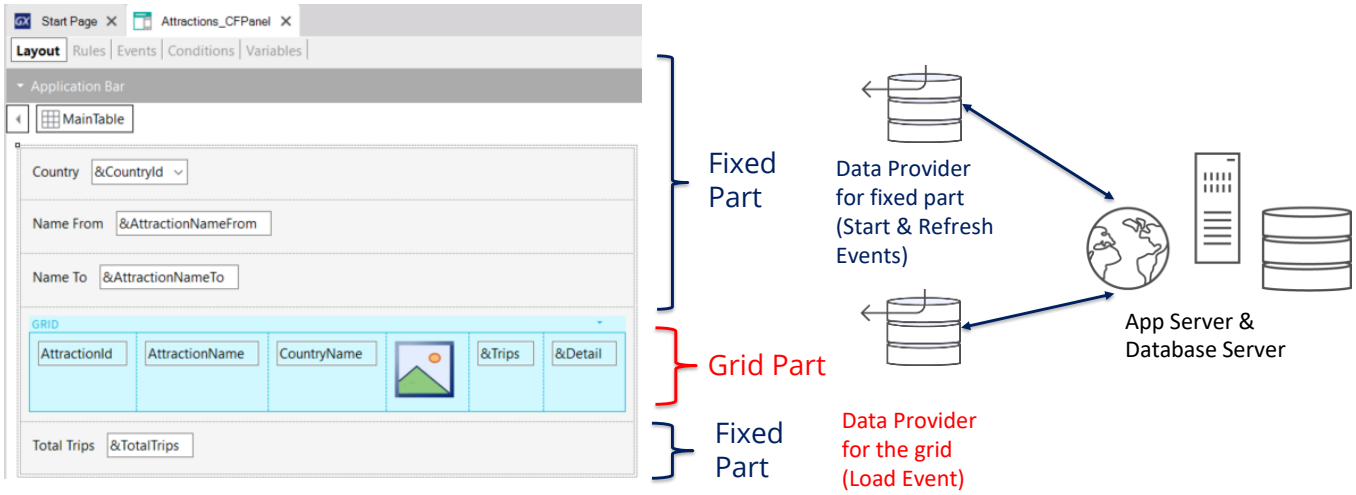
- Only for Client Side Events with more than 1 line of Code.
- Stop execution on Error.
- Automatic Error Handling.

El comando Composite se utiliza en eventos del lado del cliente en objetos panel, cuando deba escribirse dos o más líneas en un evento, y en este caso es obligatorio agrupar el código completo del evento dentro de este comando.

La importancia de este comando es que, cuando ocurre un error en la secuencia de llamadas, la ejecución se detiene y se manejan los errores automáticamente, desplegándolos en la pantalla sin tener que implementar ninguna programación.

Esta es una gran diferencia con los webpanels, dado que en éstos cuando dentro de un evento un objeto llamado produce un error, no se interrumpe la ejecución, sigue en la sentencia siguiente y es el desarrollador quien debe encargarse de manejar los errores y programar las acciones a tomar.

Partes de un objeto panel



Veamos ahora algunas características del objeto panel.

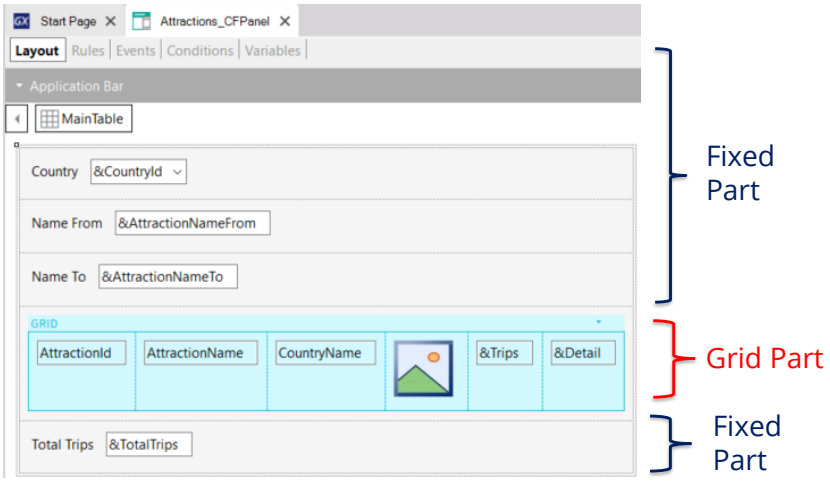
Como mencionamos antes, en los objetos panel podemos identificar dos partes distintas, la primera parte a la que denominamos la parte fija o plana que contiene todo aquello que esté en el formulario y que no esté incluido en ningún grid. En el ejemplo que vemos en pantalla, la parte fija esta compuesta por las variables de los filtros: &CountryId, &AttractionNameFrom, &AttractionNameTo y el total de viajes &TotalTrips.

La segunda parte la denominaremos parte del grid o variable, y estará compuesta en este caso por el grid que contiene los datos de las atracciones.

Siempre tendremos una parte fija, y podemos tener una parte variable por cada una de las grillas que tenga el panel.

Para cada parte (fija y grid) GeneXus va a generar automáticamente Data Providers que serán publicados como servicios en el servidor y resolverán el acceso a los datos. Estos data providers no los veremos en la Base de Conocimientos ya que GeneXus se encargara de generarlos y mantenerlos, pero si podremos ver a que datos acceden cada uno, si vemos su listado de navegación, como hicimos antes.

Orden de ejecución de los eventos



- ClientStart
- *Call to Start & Refresh Data Provider*
- Start
- Refresh

- ***Fixed Part is Drawn***

- *Call to Grid Data Provider*
- Load

- ***Grid Part is Drawn***

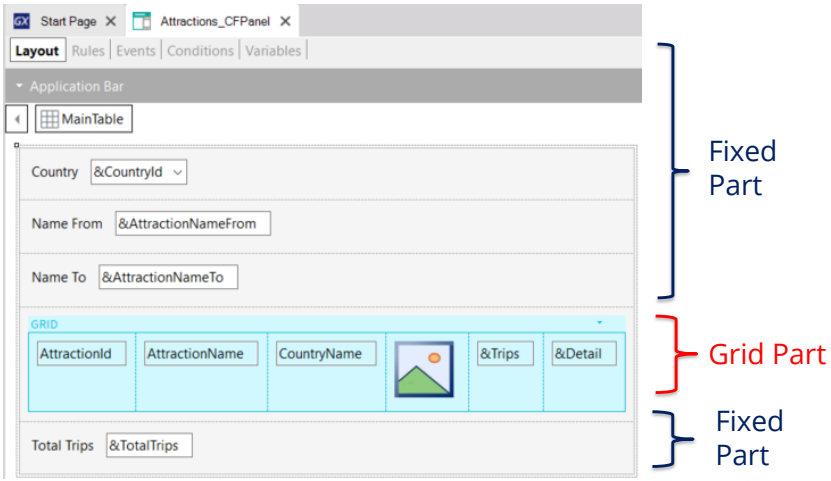
Ahora veamos que es lo que ocurre cuando ejecutamos un objeto panel.

Primero se ejecuta el evento ClientStart, se ejecuta solo una vez y corre como ya vimos en el cliente, .Luego se ejecuta un data provider que devolverá los datos necesarios para cargar la parte fija del panel. Este data provider integra la ejecución del código de los eventos Start y Refresh que se ejecutarán en el servidor y devuelve en un único resultado, la información para cargar la parte fija. Luego se dibuja la parte fija del panel.

A continuación, se ejecuta un segundo Data Provider que resolverá la recuperación de los datos requeridos por la grilla. Dentro de la ejecución de este data provider se ejecuta el código del evento Load en el servidor. Este evento Load se ejecutara N veces cuando la grilla tenga tabla base, una vez por cada registro y una sola vez si la grilla no tiene tabla base.
Al finalizar la ejecución del data provider el mismo devolverá la información generada por todas estas ejecuciones del evento Load en un único resultado, con el que se cargará el grid y luego, se terminará de dibujar el grid.

La particularidad de que la pantalla sea dibujada en dos momentos distintos tiene sus consecuencias como veremos a continuación.

Determinación de tablas base de la parte fija y del grid



Attributes involved in determining of **Fixed Part base table**:

- Attbs in fixed part of panel (form)
- Attbs outside For Each in events: Refresh, Buttons or controls in fixed part and Application Bar
- Attbs in Conditions Tab

Attributes involved in determining of **Grid Part base table:**

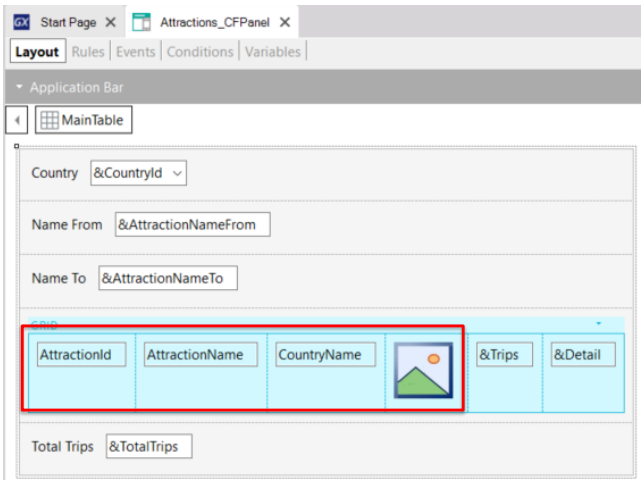
- Attbs in grid columns
- Attbs in Order, Search, Advanced Search and Conditions
- Attbs outside For Each in events: Load, Buttons or controls inside the grid

En un objeto panel la parte fija y el grid determinan navegaciones independientes y cada parte tendrá su tabla base, como si hubiera dos for eachs paralelos. Esto establece una diferencia importante con los webpanels.

Para determinar la tabla base de la parte fija, se tendrán en cuenta los atributos que pertenezcan a la parte fija del form, los atributos que pertenezcan a los eventos asociados a la parte fija siempre y cuando estén fuera de un comando For Each (en el evento Refresh y eventos asociados a botones o controles de la parte fija, incluidos los del Application Bar), y atributos del Tab Conditions del objeto Panel.

Para determinar la tabla base del grid, se tendrán en cuenta los atributos incluidos en las columnas del grid, tanto visibles como no visibles, los atributos referenciados en el Order, Search, Advanced Search y Conditions del grid y los atributos fuera de cláusulas For Each que estén en el evento Load y en los eventos de botones o controles dentro del grid.

Determinación de tablas base de la parte fija y del grid



Fixed Part: No base table

Fixed Part

Grid Part

Fixed Part

Grid Part base table: Attraction

```
Event Load
    &Trips = Count(TripDate)
Endevent

Event Refresh
    &TotalTrips = 0
    For each Trip.Attraction
        &TotalTrips += 1
    Endfor
Endevent

Event &CountryId.ControlValueChanged
    Grid1.Refresh()
Endevent

Event &AttractionNameFrom.ControlValueChanged
    Grid1.Refresh()
Endevent


Event &AttractionNameTo.ControlValueChanged
    Grid1.Refresh()
Endevent

Event Start
    &Details = "Details"
Endevent

Event &Details.Tap
    AttractionDetail_CFPanel.Call(AttractionId)
Endevent
```

Por lo tanto, en el ejemplo que vimos, como en el formulario no hay atributos en la parte fija (sólo variables), tampoco hay botones, no hay atributos en el Tab Conditions del panel, tampoco hay atributos en los eventos asociados a las variables y en el evento Refresh no hay atributos fuera del For Each, la parte fija del panel no tiene tabla base.

En el grid tenemos presentes los atributos AttractionId, AttractionName, CountryName y AttractionPhoto, por lo que la tabla base será Attraction, ya que en el evento Load el único atributo presente está en la fórmula Count.

Más información….

https://wiki.genexus.com/commwiki/servlet/wiki?24829

Si quiere saber más sobre los temas tratados en este video, puede consultar la documentación relacionada a la programación para dispositivos móviles nativos (Smart Devices), ya que como dijimos antes, el objeto panel también se utiliza en dichas aplicaciones.

Puede encontrar más información en la página del wiki que aparece en pantalla.

training.genexus.com
wiki.genexus.com
training.genexus.com/certifications