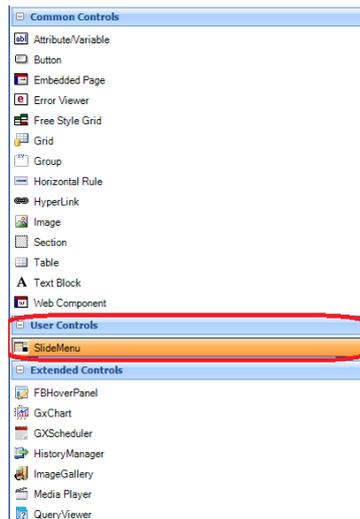


## Data Provider que carga colección con valores fijos (no extraídos de la base de datos)

Hasta el momento hemos visto ejemplos de uso de Data Providers, que han accedido siempre a la base de datos para recuperar información.

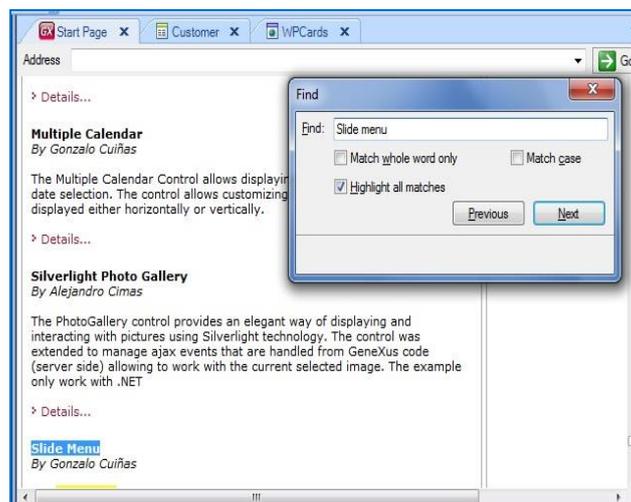
Sin embargo también es posible definir un Data Provider que cargue una estructura de datos con valores fijos.

Como ejemplo de aplicación, usaremos el user control: Slide Menu, para ofrecer un menú siempre visible en las páginas de la aplicación:



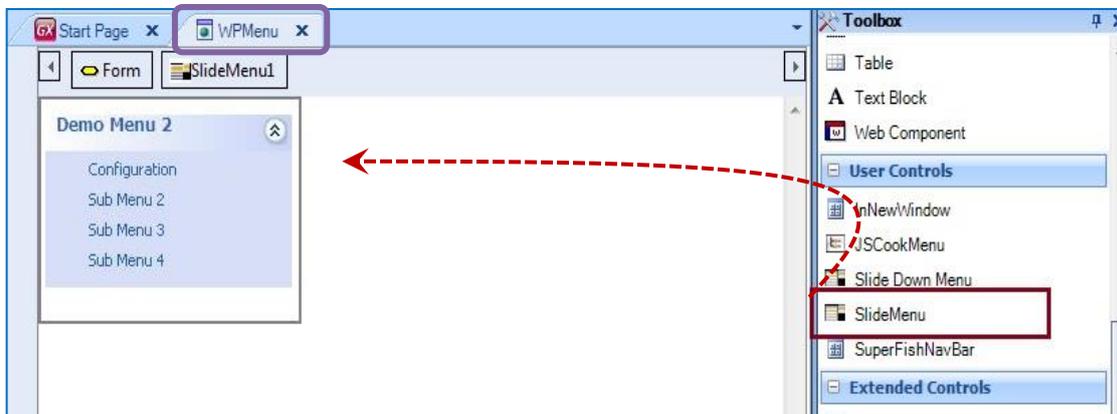
Como ya sabemos, cuando se utiliza un user control, se suele crear automáticamente algún SDT en la base de conocimiento y ahí surge la necesidad de hacer la carga de este SDT utilizando un Data Provider (que en este caso, cargará valores fijos).

El user control Slide Menu no está inicialmente integrado a la barra de herramientas de GeneXus, así que hay que instalarlo. Para ello, en el catálogo de user controls disponible en la Start Page, presionamos Ctrl + F y buscamos: Slide Menu



Una vez encontrado, seleccionando Details, es muy sencillo proceder a su instalación.

Bien. Una vez completada la instalación del user control, creamos un web panel de nombre WPMenu y desde la Toolbox arrastramos el user control Slide Menu sobre el form.



Instantáneamente, se crea en la base de conocimiento este SDT de nombre SlideMenuData:

Name	Type	Description	Is Collection
SlideMenuData		Slide Menu Data	<input checked="" type="checkbox"/>
SlideMenuDataItem		SlideMenuDataItem	
Id	Character(500)	Id	<input type="checkbox"/>
Name	Character(500)	Name	<input type="checkbox"/>
Link	Character(500)	Link	<input type="checkbox"/>

Si lo observamos que se trata de una colección de elementos que componen un menú. Cada uno de los elementos tendrá un identificador, un nombre o un título en el menú y un objeto al cual invocar.

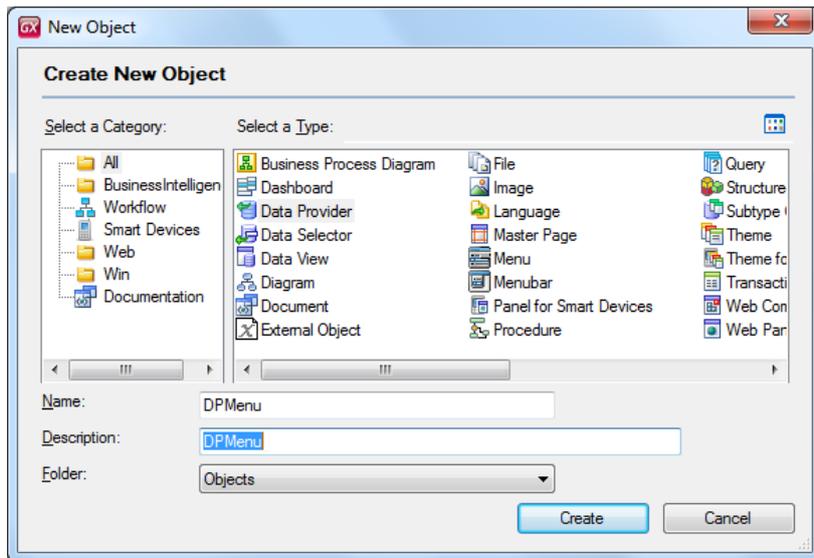
Pero además, en el web panel se han definido algunas variables. Vamos a verlas:

Name	Type	Is Collection	Description
Variables			
Standard Variables			
SlideMenuData	SlideMenuData	<input type="checkbox"/>	Slide Menu Data
selectedItem	SlideMenuData.SlideMenuDataItem	<input type="checkbox"/>	selected Item
SlideMenuDataItem	SlideMenuData.SlideMenuDataItem	<input type="checkbox"/>	Slide Menu Data Item

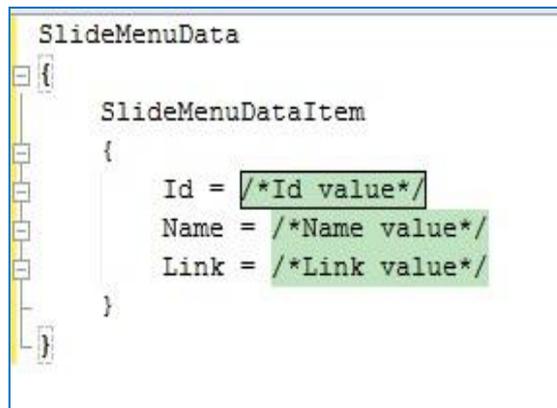
Esta variable es del tipo SlideMenuData, o sea, del tipo colección que define al menú.

Mientras que las otras dos variables son del tipo SlideMenuData.SlideMenuDataItem. O sea, que son elementos de la colección del menú.

Para cargar la variable colección que contiene la información del menú en el web panel, vamos a utilizar un Data Provider. Así que creamos un objeto de tipo Data Provider de nombre DPMenu.



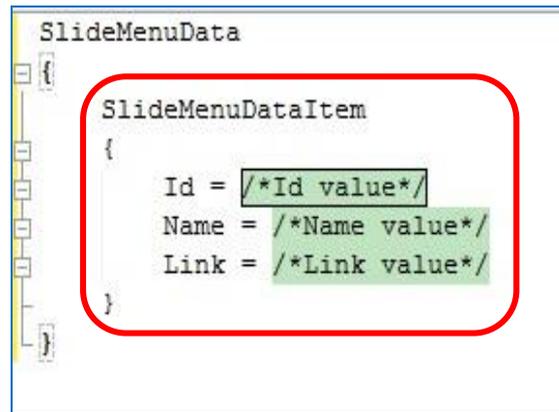
Y arrastramos el SDT SlideMenuData.



¿Con qué datos cargaremos la colección? ¿Necesitamos acceder a la base de datos?

No. Porque las opciones que mostrará el menú, no están almacenadas en la base de datos, así que tendremos que cargar esta estructura con valores fijos.

¿Y cómo lo hacemos? Copiaremos y pegaremos este grupo N veces, tantas veces como opciones tenga el menú.



De esta forma cada grupo representa una instancia de la colección a la cual le cargaremos valores fijos.

Vamos a definir nuestro menú solamente con dos opciones, para acceder a trabajar con Países y Atracciones.

Le asignaremos al primer grupo el identificador: 1, la descripción “Work with countries” y la invocación al objeto correspondiente utilizando el método Link.

```

SlideMenuData
{
  SlideMenuDataItem
  {
    Id = "1"
    Name = "Work With Countries"
    Link = WWCountry.Link()
  }
}

```

Para definir la segunda opción del menú, asignaremos el identificador: 2, la descripción: “Work with attractions” y la invocación también al objeto correspondiente:

```

SlideMenuData
{
  SlideMenuDataItem
  {
    Id = "1"
    Name = "Work With Countries"
    Link = WWCountry.Link()
  }
  SlideMenuDataItem
  {
    Id = "2"
    Name = "Work With Attractions"
    Link = WWAttraction.Link()
  }
}

```

Y así podríamos seguir definiendo todas las opciones que deseemos mostrar en el menú.

Volvamos ahora al web panel para invocar a este Data Provider. En el evento Start, realizamos la invocación al Data Provider de forma tal que el mismo retorna su output en la variable &SlideMenuData:

**Event Start**

```
&SlideMenuData = DPMenu()
```

**EndEvent**

Vamos a probar lo que implementamos Presionamos F5.

Ejecutamos el web panel, y vemos el menú con las opciones definidas.

Vamos ahora a seleccionar una de las opciones, por ejemplo Work With Country.

Vemos que se invoca el objeto correspondiente, pero el menú desapareció.



¿Y si queremos que el menú esté siempre visible en todas las páginas de la aplicación?

Entonces dicho menú debería formar parte de la master page que se crea automáticamente con la KB.

Vamos entonces a hacer que nuestro menú esté dentro del objeto AppMasterPage.

Para lograrlo tenemos dos opciones. Veamos la primera.

Vamos a declarar al web panel WPMenu como Component. Si observamos la propiedad Type del web panel, vemos que puede tomar el valor Master Page, Web Page o Component.

Por defecto todo web panel es una Web Page y ya sabemos que una Master Page es una plantilla para que toda la aplicación se vea con el mismo diseño. Pero, ¿qué significa definir que un Web Panel es un Component?

Cuando declaramos que un web panel es un Component, estamos diciendo que dicho web panel podrá ser incluido dentro de otro objeto.

Sigamos con nuestro ejemplo.

Observemos que en la master page creada por GeneXus, hay un control llamado ContentPlaceHolder. Se trata del contenedor donde se presentarán las transacciones y demás objetos o páginas. O sea, que siempre vemos el diseño de la Master Page en ejecución, y en el sector donde está ubicado el control ContentPlaceHolder, vemos nuestros objetos interactivos.

Bien. Ahora queremos que siempre esté disponible el menú a la izquierda del control ContentPlaceHolder. Así que desde la Toolbox vamos a arrastrar un control de tipo WebComponent que será el encargado de “contener” y mostrar un objeto de la base de conocimiento que indiquemos (en nuestro caso, el web panel que contiene el menú).

Para que los controles queden alineados utilizaremos una Tabla.

Definimos una tabla de una fila y dos columnas, ubicamos el ContentPlaceHolder a la derecha, y ahora sí arrastramos desde la Toolbox el control de tipo WebComponent a la izquierda.

Observemos las propiedades de este control WebComponent.

En la propiedad Object debemos indicar el objeto de tipo Component que será cargado dentro de este control. Así que completamos la propiedad con el nombre de nuestro web panel WPMenu.

Presionemos F5 para ver el resultado en ejecución.

Seleccionamos por ejemplo la opción Work With Country y vemos el menú contenido dentro de la master page.

¿Qué pasa si seleccionamos ahora Work With Attraction?

El menú también aparece visible.

La otra alternativa sería eliminar el web panel WPMenu de nuestra base de conocimiento y arrastrar directamente el user control Slide Menu sobre la master page, ubicándolo siempre en una celda a la izquierda del control ContentPlaceHolder. De esta forma el menú estaría incluido directamente en la AppMasterPage y no en un objeto declarado como Component.

Solamente restaría invocar al Data Provider desde el evento Start de esta master page para cargar el menú.

Cualquiera de estas dos propuestas, ofrece que el menú forme parte del diseño de la master page, y por lo tanto aparezca visible en todos los objetos.