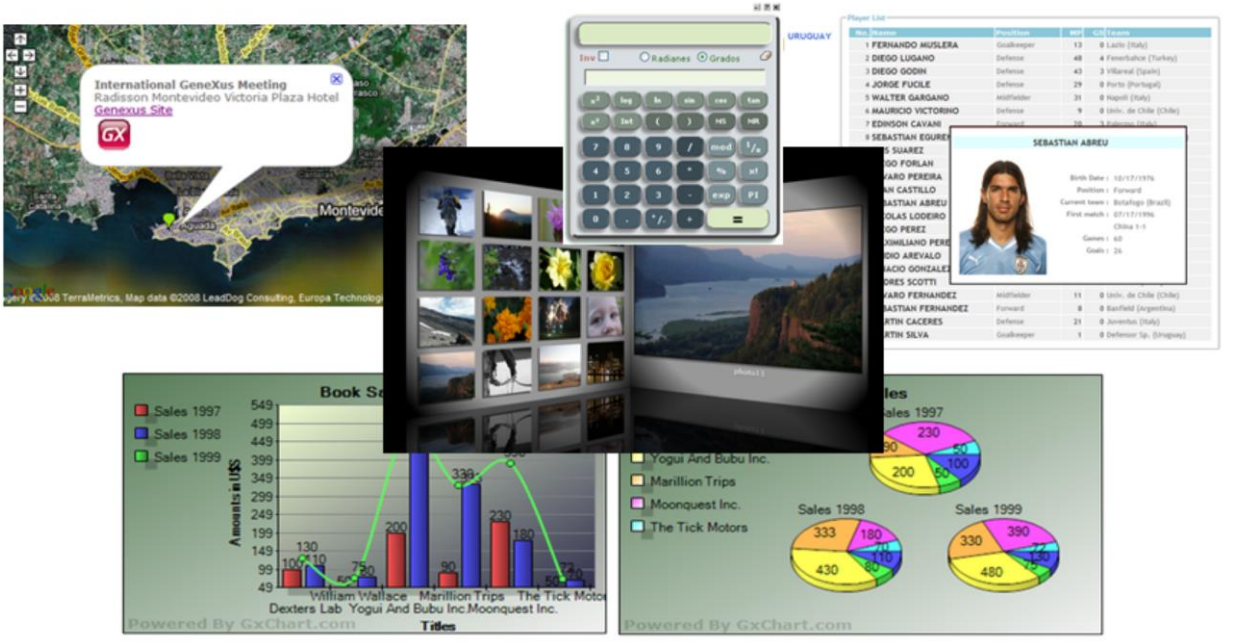
Screen components

Extended controls (User controls)

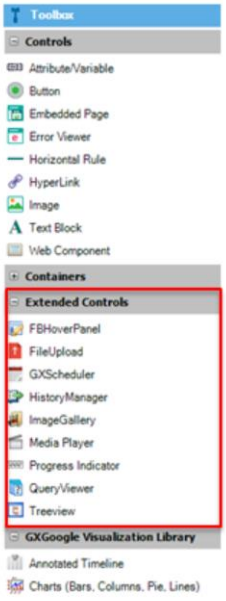
GeneXus 16

Extended controls (user controls)



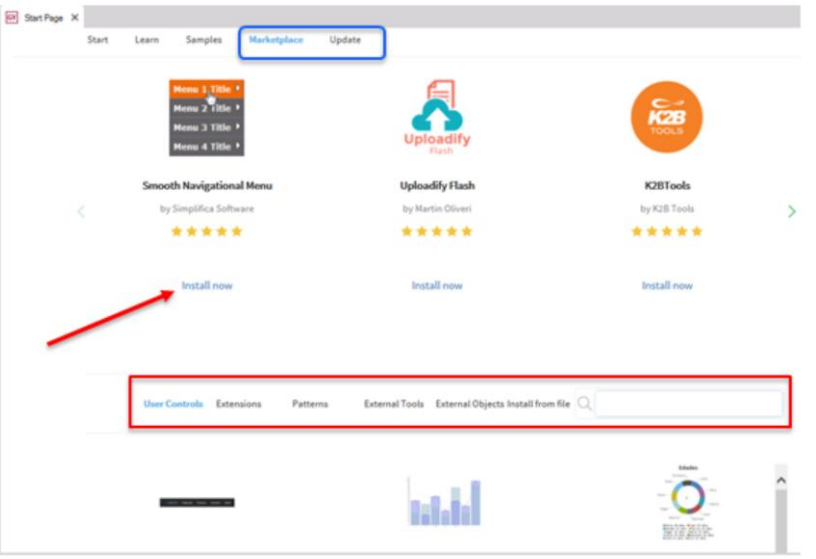
Extended Controls (also known as User Controls) allow us to create applications with rich, user-friendly interfaces, for example, by incorporating menus, maps, charts, calendars, diagrams and other resources.

Extended controls (user controls) in the Toolbar



In addition to the usual controls that are available in the toolbox, GeneXus allows us to create our own controls or use controls created by other users, which are called “user controls” or “extended controls”.

Start Page: Marketplace y Update



We can also obtain other User Controls to incorporate into our GeneXus IDE, by accessing the GeneXus Marketplace directly from the Start Page.

If we select the Start Page, we see a link to the Martketplace and another link called Update.

The Start Page contains new Tabs that replace the Add-in Manager: These tabs are named Marketplace and Update.

The tab Marketplace is useful to browse for User Controls, Extensions, Patterns, External Objects, etc. and install them , most of them for free. These resources were uploaded by members of the GeneXus Community. You may also create your own user controls, extensions, etc.

The tab Update checks for updates of the already installed ones and allows to update them in the current installation

Example...

The travel agency wants to show photos of the attractions it offers in an image gallery:

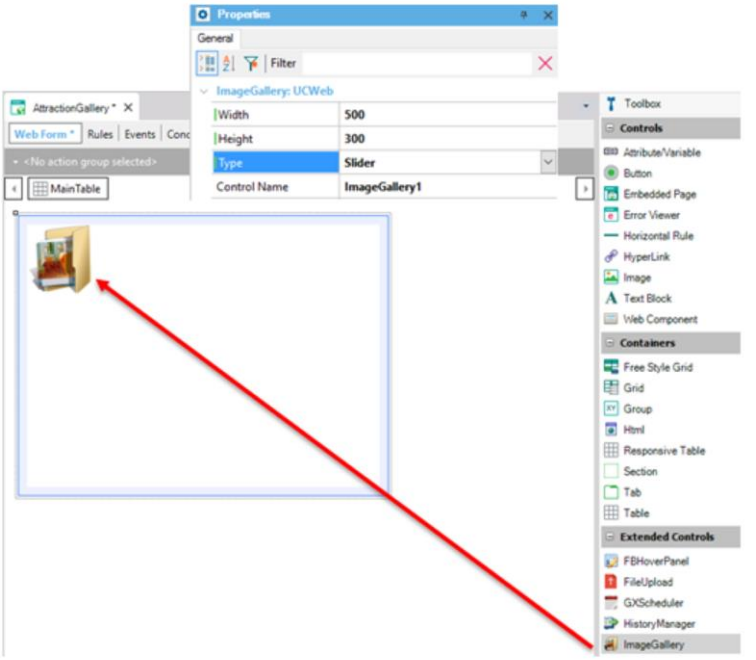
In a web panel we include a screen component (Extended Control) called ImageGallery

Suppose that the travel agency wants to show photos of the tourist attractions they offer in an image gallery.

In order to see the photos of tourist attractions as they are displayed on the screen, we need to use a special screen component, (extended control) called Image Gallery.

In general, extended controls use variables of structured data types; the data to be used must be loaded in these variables.

Solution

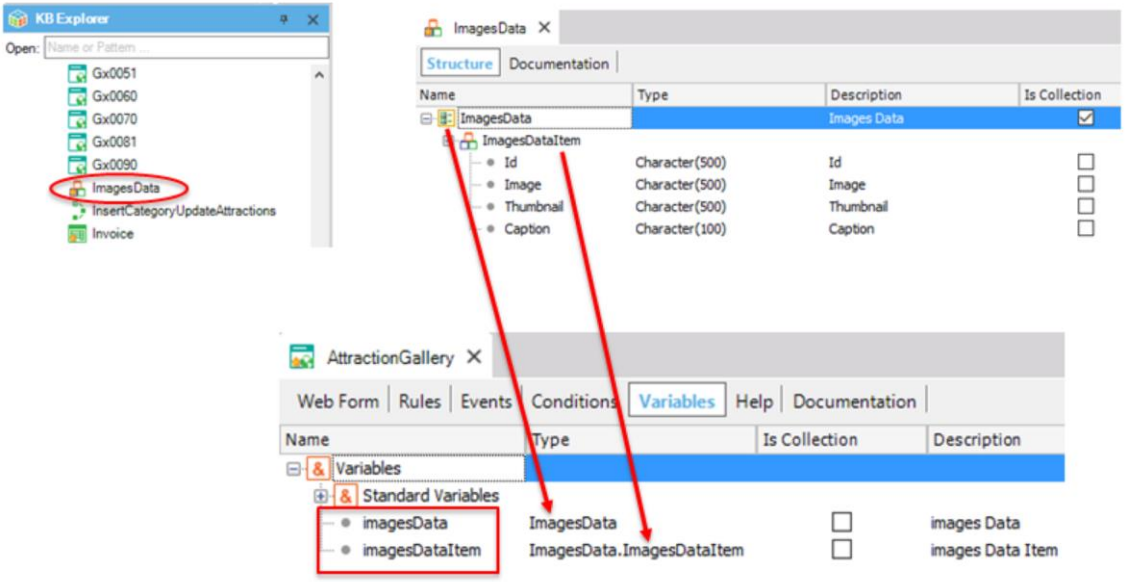


To create an image gallery, first we need to create a Web Panel where we will place this gallery, so we create a new object, of web panel type, and call it AttractionGallery.

Now we open the toolbox and drag an ImageGallery control to the form of this Web Panel.

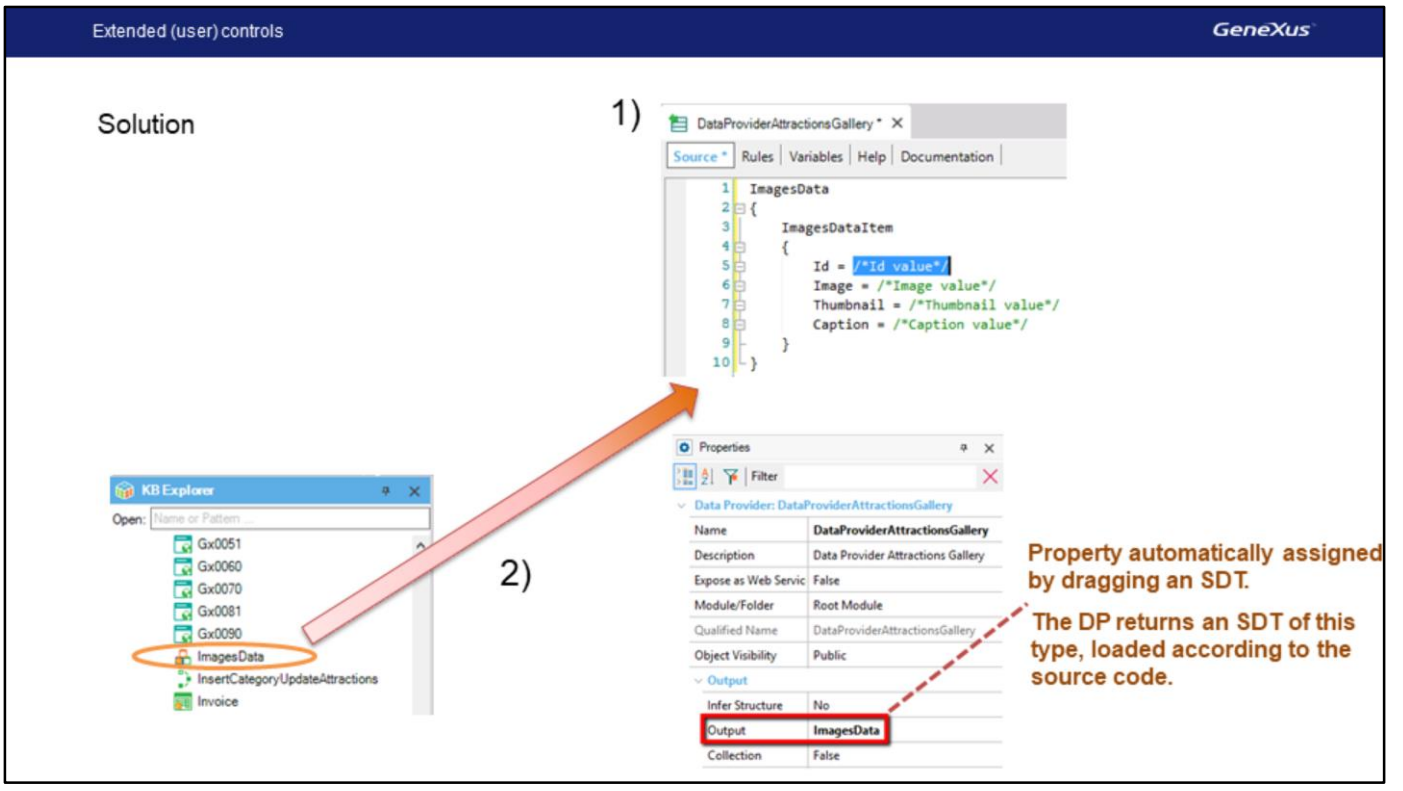
We select the ImageGallery control in its properties, and assign it a width of 500, a height of 300 and Slider type.

Solution



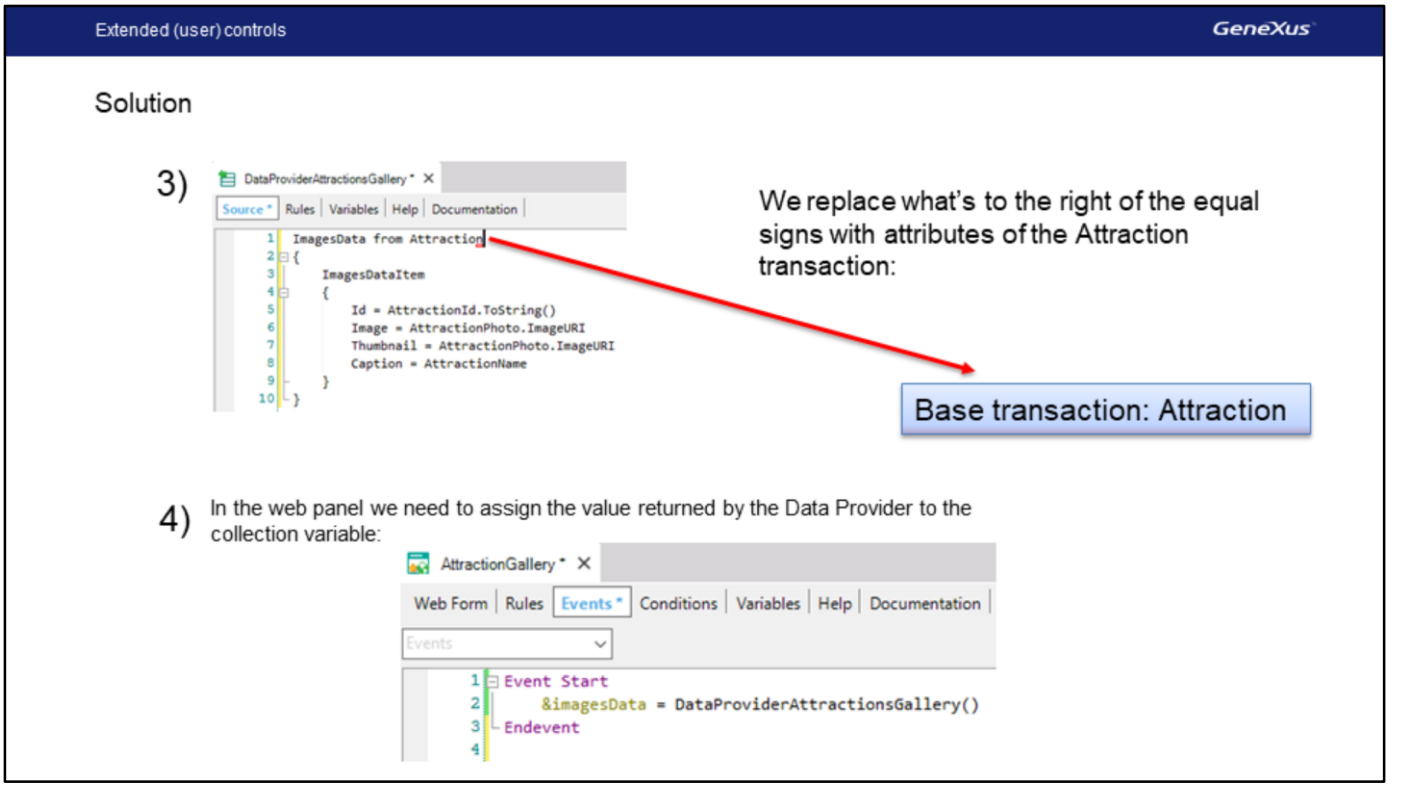
Note that this action has also created a structured data type called ImagesData; in the web panel we can also see that two variables have been created to save the image collection (of ImagesData type) and the item selected (of ImagesData.ImagesDataItem type).

To load the collection using attraction data, we will create an object of Data Provider type and call it: DataProviderAttractionGallery



1. We create the object Data Provider DataProviderAttractionGallery.

2. We drag the ImagesData structured data type from the Knowledge Base Navigator window to the Data Provider Source.
   - After doing this, if we open the Data Provider properties we can see that GeneXus has assigned the name of the ImagesData collection to the **Output property**. This means that the Data Provider will return a collection of ImagesData structured data type loaded with data.
   - Since ImagesData is already a collection, it isn't necessary to configure the **Collection property** to True. We would do so if we wanted the Data Provider to return a collection from a simple structured data type.



3. Assuming that we want to load this collection with the contents of the ATTRACTION table, we need to replace, in each line of the Source that assigns a value to a member, the corresponding attribute to the right of the assignment sign.

GeneXus determines the database table that has to be navigated to perform the loading according to the base transaction to the right of the Form clause. In this case, it runs through the ATTRACTION base transaction, and for each attraction found, it copies the data stored in each record's attributes to the members of a new item of the collection.

Note that in some cases, the data needs to be converted in order to be stored because the members are all of Character(500) type.

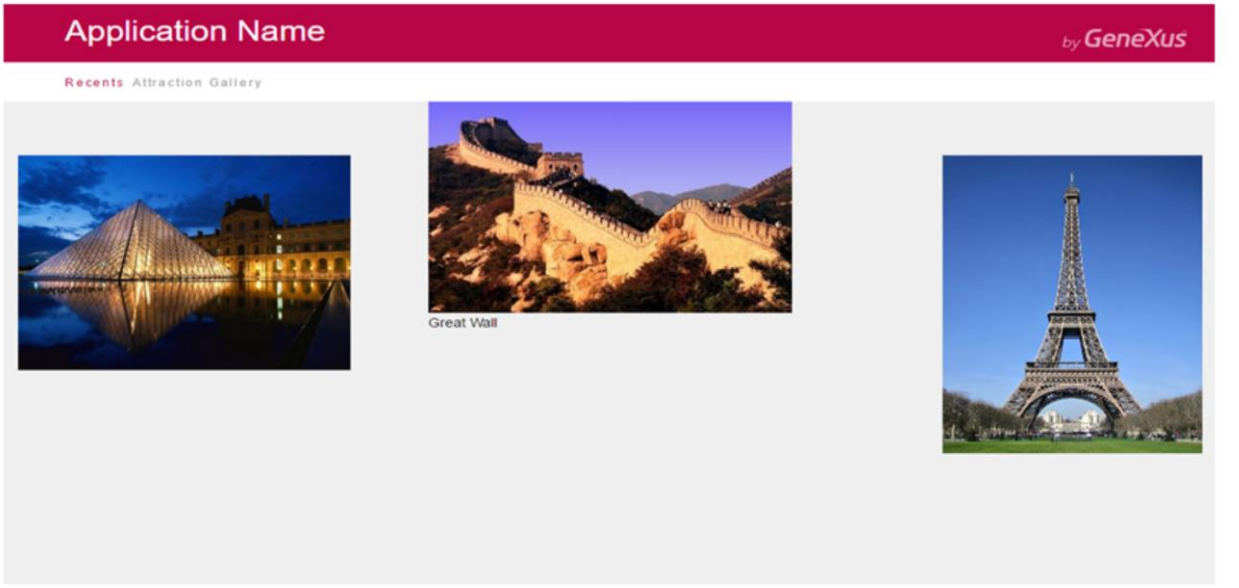
In this way, the attraction identifier (of ID type), is converted to String to assign it to the Id member. Also, the members Image and Thumbnail are loaded with the photo URL, through the ImageURI property of the AttractionPhoto attribute.

The final result is that the data of all the clients in the database will have been stored in the collection.

Note that we have simply dragged the definition of the SDT to the Data Provider source and **stated** which attribute values we want to be loaded in the collection of attraction photos.

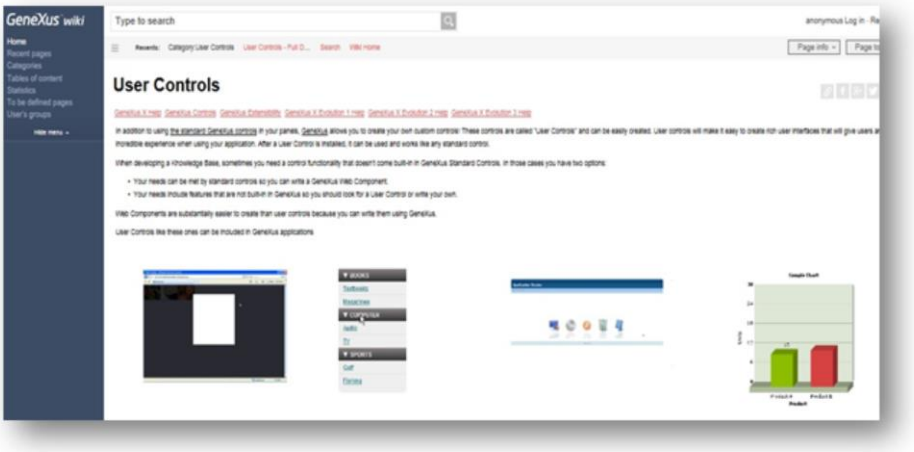
4. Lastly, we open the AttractionsGallery web panel events to invoke the Data Provider so that it loads this collection. We delete the sample code and in the Start event we assign the result returned by our Data Provider to the &ImageData variable; that is to say, the collection of attractions stored in the database.

Application at runtime



Here we see the application at runtime.

Learn more about Extended Controls (or User Controls):



http://wiki.genexus.com/commwiki/servlet/wiki?5273,Category%3AUser+Controls.

For more information about User Controls, read the Community Wiki page about this topic on the link shown on the screen.

Find other resources: marketplace.genexus.com

To access the User Controls published by the community and download them, visit: marketplace.genexus.com

GeneXus™

The power of doing.

Videos	training.genexus.com
Documentation	wiki.genexus.com
Certifications	training.genexus.com/certifications