

Stencils

User Interface & User Experience

GeneXus[®] 16



Let's start studying Stencil objects and what they are used for; next, we will see how to apply Stencils to our Knowledge Base.



At development time, usually there are elements repeated in several places; for example, in the event app we're developing, we've seen that a speakers list appears in several places. In the screen on the left we see the List of the Speakers Work With for Smart Devices component, and to the right we have the same list that is shown in the Speakers section of the Detail of the Countries Work With For Smart Devices component.

Also, we frequently want to show the same information using the same design. In these cases, we run the risk that they do not look the same, that they do not have the same design; also, every time we require changes in the way this information is presented we must modify all those places where the design is included.

One of the tasks that designers and developers must do together is to identify these patterns in the application, which we will call UI Control Composition. Once we have identified them, we are going to establish a unique design, so that this composition is displayed everywhere in the same way.

The new Stencil object will allow us to create these UI Control Compositions, defining their design without thinking about the code; these components will be reusable in different objects, and in this way we will maintain application consistency less effortlessly.

When we want to modify the design we will do it from the Stencil and the changes will be replicated on all the objects that use it.

Because it is an independent object, it will provide all the advantages of these objects; for example, it can be exported to be shared, we can maintain the history of changes that it is undergoing, and we will be able to use a Cross Reference to know where the Stencil is used. When we change it, we will know the impact this change will have.

The Stencil object will be the place where the developers' work will converge with that of the designers in the same object.



Let's see all this we've learned in GeneXus.

Demo: Stencil Implementa	ition				
Lard & Detail	aten i	Annu Cans Annu Cans Annu Cans Annu Cansulation Annu Cansulatio Annu Cansulation Annu Cansulation Annu	Age A Converse tail New Object Solid a Calegory Common	Select a Type	General Class If (1) V Filte × er Table rot Nam poerral okurns: ows Sp() Kdth
CompanyName CompanyName CompanyName CompanyName Com	Insert Attribute/Variable_ Insert Button_ Delete Del Convert to Canvas	Auto Grow Class Beckground Vaible Invisible Mo Enabled	Create a new Stenci	Sociaterini	leight uto Gron Jass ackgrou Isible
	Convert to Flee Wing as New Stancil Search for Compatible Stancils Go To Event insert Row Before Insert Row After Insert Column Before Column Sefore	 Scroll Beha Scroll Factor Zoom Facto Scroll Attad Layout Beh 	Description	Speaker Info	Cancel Com Fac
c 3 Any Platform, Default Orientations	insert Column After Delete Row Delete Column	Evenand Bros Output O Pro			

First, let's review the list of Speakers, as we had already seen in the presentation, and also in Countries. Let's see the Speakers of a Country, and note that they are different designs.

We open GeneXus.

First, we open WorkWithDevicesSpeakers and look at the List. Here is a table that we will turn into a stencil by right-clicking and using the option Wrap As New Stencil. As a result, the table will be turned into a Stencil, we will call it SpeakerInfo and click on Create.

Stencils		Gene
Demo: Stencil Implementation	Properties General Class Second	9 sakeef ull Vanne
Cooperation V	Control Name	&SpeakerFullName
	Attribute	&SpeakerFullName
I sugut Variables	Readonly	False
Layout valiables	~ Appearance	
A I MainTable III Table2 III SpeakerFullName	Label Position	None
	Class	AttributeTitlegx16
	Visible	True
	Invisible Mode	Keep Space
	Enabled	Irue
	Format	For the full Name
	invite message Speaker Full Name	
&SpeakerFullName &SpeakerId	Control Tune	LQ.
^o	Auto Grow	Edite
00	Input Type	Values
	Supper	No
&CompanyName	Auto correction	True
	Auto capitalization	First word
	 Cell information 	
	Row Span	1
	Col Span	1
	Horizontal Alignment	Default
	Vertical Alignment	Default

The first thing we are going to see here is that the attributes we had in the grid are now variables; on the other hand, we don't have all the properties available in the Stencil, only those that have to do with its design not with its behavior.

The classes are taken into account, as well as the sizes of the tables we had defined, etc. Note that we only have the Layout and Variables. We don't have Events because the Stencils don't implement any logic. We save the Stencil.

Demo: Stencil Implementation		Properties	<i>a</i> >	
		General Class		
WorkWithDevicesSpeaker	x	🔚 🔁 🏹 Filter	>	
List & Detail		Attribute/Variable: S	peakerFullName	
Level (Speaker)	Layout Rules Events Conditions Variables	Attribute	SpeakerFullName	
Detail	Application Bar Insert	Readonly	True	
Section (General) Section (Speaker)	MainTable Grid1 Table1 SpeakerInfo	V Link		
		Autolink	True	
		V Virtual Keyboard		
		Enter Event	<platform default=""></platform>	
	SpeakerFullName SpeakerId	~ Behavior		
	o CompanyName	Input History	True	
		Is Password	False	
	o o			

Note that now in WorkWithDevicesSpeaker, in the List, the table has become a Stencil, SpeakerInfo. Also, note that in attribute properties we have fewer properties to configure, because all the properties related to design are on the Stencil side, and here we only have properties related to behavior. The attributes are left as they were originally. For example, we wouldn't be able to move an attribute to another place because everything highlighted in gray within the table or the SpeakerInfo Stencil is ReadOnly. We save this object as well.

Now we will apply the Stencil to WorkWithDevicesCountry, inside the Speakers Section. Note that here we don't have a table, but directly the attributes inside the grid. We will create a table to turn it into a Stencil, take the attributes inside the table and have the option to turn the table into a Stencil. We will use the option Search for Compatible Stencils, which will open the Stencils tab. Here we only have one for now, but if we had many we would be able to search them by name.

Here are three options, Open, Insert and Swap. Let's look at each one of them.

The Open option will open the Stencil directly; the Insert option will insert the contents of the Stencil in the place where we're positioned in the Layout. Let's give it a try. I'll make the Insert at the end. Note that the entire Stencil table is added with the variables where we were positioned. We delete.

The Swap option is the one we're interested in right now. It will have some intelligent features and will try to search, for each variable in the Layout, a match by data type in the Layout of the Panel where this Stencil is applied.

In this case, the variable &SpeakerImage of the Stencil will find a match with the SpeakerImage attribute, so it will leave the attribute instead of using the variable. Of course, it will apply the style set for the Stencil. For &SpeakerFullName it will look for an attribute or variable of Char type, find SpeakerName, use it and leave SpeakerName; it will not turn it into SpeakerFullName.

For &SpeakerId, it doesn't find a numeric attribute or variable in the Layout, and the same

will happen with &CompanyName. We don't have another attribute, because it has already been mapped (SpeakerName); we don't have a second attribute of Char type, so it will add &CompanyName to the Layout. Let's see it.

We click on Swap and, as expected, we find SpakerImage (Attribute), SpeakerName (Attribute) instead of &SpeakerFullName, &SpeakerId (as a variable) and &CompanyName (as a variable).

We may leave the variables and load them in the Load event. In this case, we're not interested, so we change it so as not to use variables and use the attribute directly. We save.

I forgot to change SpeakerName for SpeakerFullName, so now we save again and run the application.

Let's check how the system looks like; we see the list of Speakers with the design it previously had, only that now it uses and makes reference to the Stencil.

For example, let's see the one we had already seen, Uruguay, in Speakers, and that now it has exactly the same designs in both lists.

Line & Devial Line & Genelari Line & Genelari Line & Genelari Applicant II: Internet Applicant II: Internet Manibalis (Effecting II: Internet) Manibalis (Effecting II:	na Casi agi san Page A agi mokrimounosopiater A	
La Lett • Application Tax Insert • Table • Table La Section (General) € ()		Class
G Section (General) (HainTable Grid1 Table1 Hable2)	able: Table2	le: Tabl
	Select a Category Select a Type	trol Nav
CompanyName CompanyName CompanyName CompanyName Delete Delete Convert to Flow Convert to Flow	Columns Spie Rens Spie Width Height And Grav Class Reckground Vable Unskille Mo Create a new Stence Institute Sorel Reha Sorel Reha Description Speaker Info	olumns 3 ows Style Vidth leight uto Grov Jass ackgrour isible vvisible N nabled croll Bel
Insert Row Before	Connect Cancel	croll Fac
Inset Column Before Any Patform, Default Orientations (M. Default After Row Default Orientations (M. Default Column Default Column	Layout Bah Bana Bana Bana Bana Bana Bana Bana	icroll Att

Let's go over what we've done. First, we turned a Table into a Stencil. We did this for simplification purposes, but we could have created it from scratch like any other GeneXus object.



We also saw that we can only modify the design inside the Stencil. In the Objects that implement it, the controls are ReadOnly.

We saw the panel where the KB Stencils are displayed, and here we only saw the SWAP option; the INSERT option will not look for the variable mapping in attributes. The Stencil where we're positioned will be inserted directly, but it will not replace any content; it will add it and the OPEN option will simply allow opening the Stencil.

You can read more about the use of Stencils in the GeneXus Wiki. <u>https://wiki.genexus.com/commwiki/servlet/wiki?38418</u>

In the next video, we will begin to study in more detail the use of different controls to improve UI and UX design in our system.

GeneXus



Videos

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