

Smart Devices

GeneXus™ 15

ControlValueChanging new edit field event

Se incorpora un nuevo evento ControlValueChanging para campos variable o atributo, editables.

ControlValueChanging event

```

Event &Edit.ControlValueChanging(&NewEdit)

Composite
  &NewMessage
  ImageSendAudio.Visible = (0)
  ImageSendMessage.Visible = (0)
  Textfield.Line.Class = iif(&NewMessageLength = 0, ThemeClass:LineTextfieldFocus, ThemeClass:LineTextfield)
EndComposite

Event &NewMessage.ControlValueChanging(&NewMessageValue)
Composite
  &NewMessageLength = &NewMessageValue.Length()
  TextBlockHowLeft.Caption = Format("X/256", 256 - &NewMessageLength)
  ImageSendAudio.Visible = (&NewMessageLength = 0)
  ImageSendMessage.Visible = (&NewMessageLength > 0)
  Textfield.Line.Class = iif(&NewMessageLength = 0, ThemeClass:LineTextfieldFocus, ThemeClass:LineTextfield)
EndComposite

```

Se ejecuta mientras el usuario final está ingresando datos de entrada. Su propósito es dar feedback constante al usuario, por ejemplo con efectos de User Interface.

En el ejemplo del chat, mientras el usuario no ingresa ningún texto aparece el ícono para grabar audio. Pero cuando el usuario empieza a escribir texto en el campo, el ícono cambia al de enviar texto, y además por cada letra que el usuario va ingresando o borrando, se lleva un conteo debajo, pues no se permiten textos de más de 256 caracteres.

Si vemos el diseño del layout de esta parte del panel, vemos que tenemos la variable `&NewMessage`, que es donde el usuario va a ir ingresando el texto, y además tenemos las dos imágenes; solamente una va a estar visible en cada momento, dependiendo de si la variable tiene largo cero, es decir, si no se ingresó ningún carácter, o si tiene largo mayor que cero. Si tiene largo cero entonces se va a mostrar esta imagen, y si tiene largo mayor que cero, -lo que significa "se empezó a digitar algo"-, entonces se va a mostrar esta otra.

Pero por otro lado, cada vez que el usuario vaya agregando una nueva letra a su mensaje, esta cantidad de acá se va a tener que modificar. Por tanto necesitamos capturar cada momento en que el usuario cambia algo del campo editable. Y es para eso que tenemos este evento.

En general, la variable que va aquí... puede ser una variable o un atributo, que va a tener que corresponder a un control en el layout que sea editable. El valor que va a contener este control editable cuando se dispara el evento va a ser el valor anterior al cambio. Y esta variable de aquí, que va a tener que ser del mismo tipo de datos que esta de aquí, va a ser la que va a contener el valor nuevo.

Entonces vemos que dentro del evento lo que hacemos es calcular el largo de esta variable -la que tiene el cambio-; lo registramos en esta otra variable.

Y aquí, lo que vamos a mostrar -Caption- es 256 menos ese largo.

Si ese largo es cero, entonces vemos que la imagen -ImageSendAudio- queda con el valor visible. Y si no, si es mayor que cero, entonces el que queda visible es este otro.

Navigation