
GeneXus by Globant
GeneXus™
by Globant
training.genexus.com

GeneXus by Globant

Invocations Between Mobile Objects

Diego Marranghello

Veremos en este video, cómo invocar diferentes objetos y qué opciones de invocación tenemos. Para esto último usaremos las CallOptions, las cuales nos permitirán modificar en runtime ciertas propiedades que tienen que ver con la experiencia del usuario.

Pero antes repasemos la sintaxis de las invocaciones.

Por ejemplo el caso de las invocaciones a los objetos WorkWith:

Para invocar al List de un Work With, se hace indicando el nombre del Work With, punto, el nombre del nivel, y luego el método List sin parámetros. Y de esa manera accederemos al listado de registros.

Si queremos acceder al detalle de un registro en particular, en modo vista, se utiliza la sintaxis que vemos en pantalla, donde necesitamos pasar la primary key, para identificar de quién queremos mostrar ese detalle.

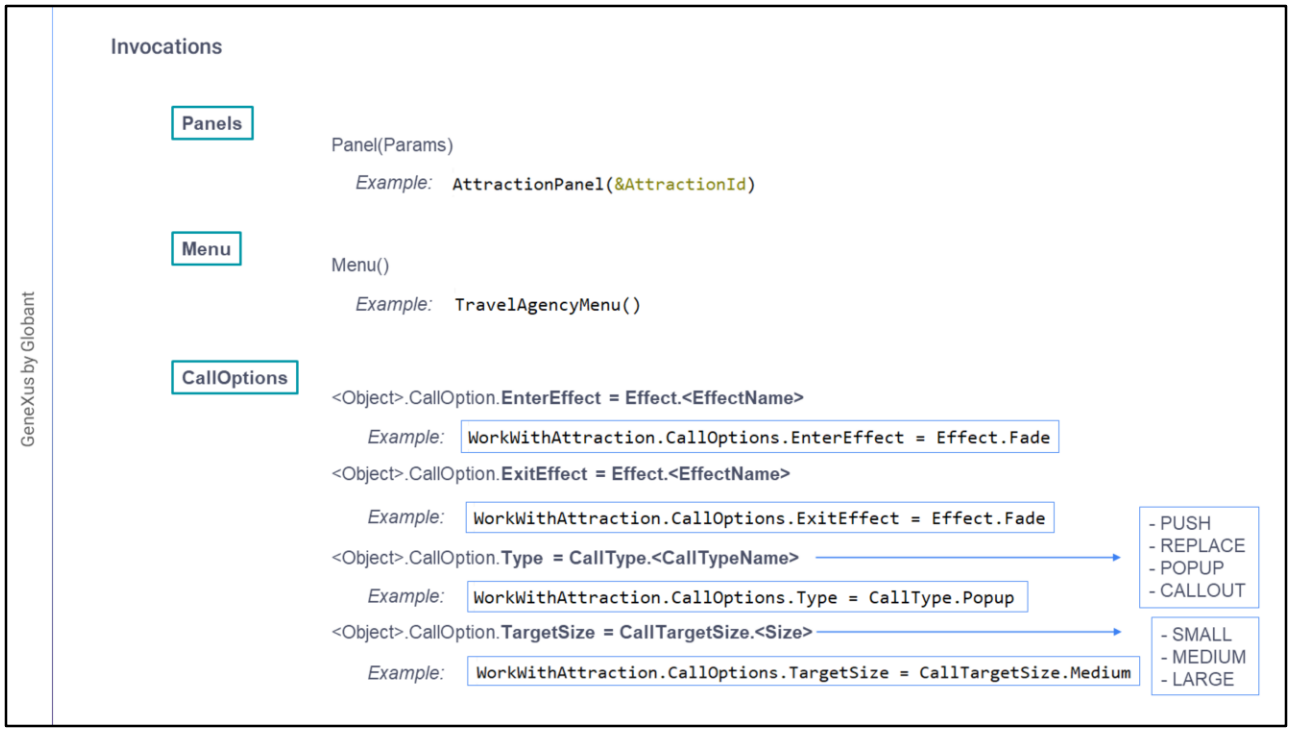
Para el caso del Detail en modo Edit, debemos indicarle el modo:

Si será Insert, Update, o Delete. O sea si será un nuevo registro, o una actualización o eliminación de uno ya existente.

En el caso de Insert tenemos 2 opciones, no pasar ningún parámetro, para que el usuario pueda ingresar todos los datos de cero. O si queremos inicializar ya algunos valores en

ese Insert, debemos pasarlos en un Business Component. Para este caso, debemos previamente inicializar en el Business Component los valores que nos interese pasar y luego esos valores van a aparecer inicializados en la pantalla.

Y para los casos de update y delete, debemos pasarle por parámetro la clave primara del registro que queremos eliminar o actualizar.



Por otro lado tenemos la invocación a los Panels, que es exactamente igual a la invocación a cualquier otro objeto GeneXus, indicando el nombre del panel, y pasando los parámetros que sean necesarios según los parámetros de entrada que tenga ese objeto Panel.

También podemos llamar a un objeto Menu.

Luego tenemos las CallOptions, las cuales como dijos nos permitirán modificar en runtime ciertas propiedades que tienen que ver con la experiencia del usuario, estas configuraciones deberán ser realizadas un instante antes de invocar al objeto.

Las propiedades que podemos configurar son:

• Los efectos de transición, que pueden ser el de Entrada o el de Salida. Los valores posibles son los del dominio Effect.

Este es un dominio enumerado, el cual ofrece todas estas opciones.

• El tipo de llamado, utilizando el dominio enumerado CallType, con los tipos Push, Replace, Popup y Callout (este ultimo solo disponible en iOS)

Esta configuración nos permite modificar el comportamiento de la llamada, respecto al tipo de call, que tendrá que ver con el stack de invocaciones y con el funcionamiento del objeto llamado.

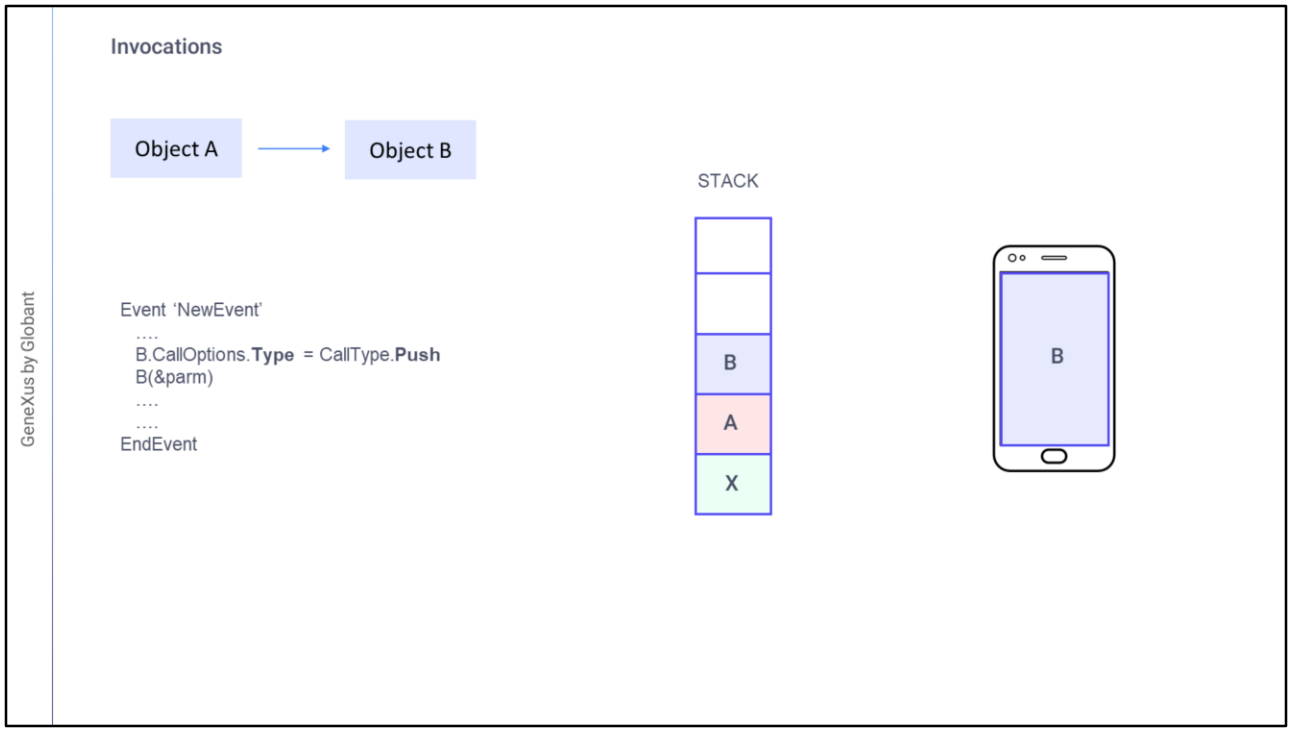
Los tipos Push y Replace definen qué va a suceder con el stack de invocaciones cuando se haga la llamada, que tendrá que ver con a qué objeto se vuelve al finalizar la ejecución del llamado o al hacer back.

Los tipos Popup y Callout, serán para que la pantalla a invocar sea del estilo popup o callout.

Para Popup, la pantalla podrá ser modal o no, dependiendo de si hay parámetros devueltos o no los hay. Las ventanas modal bloquean todas las funciones y concentran el foco en una acción particular. El usuario solamente puede hacer esa acción o cerrar la ventana, si no es modal, haciendo tap fuera del área se volverá al llamador.

Para el caso de Callout, no será modal.

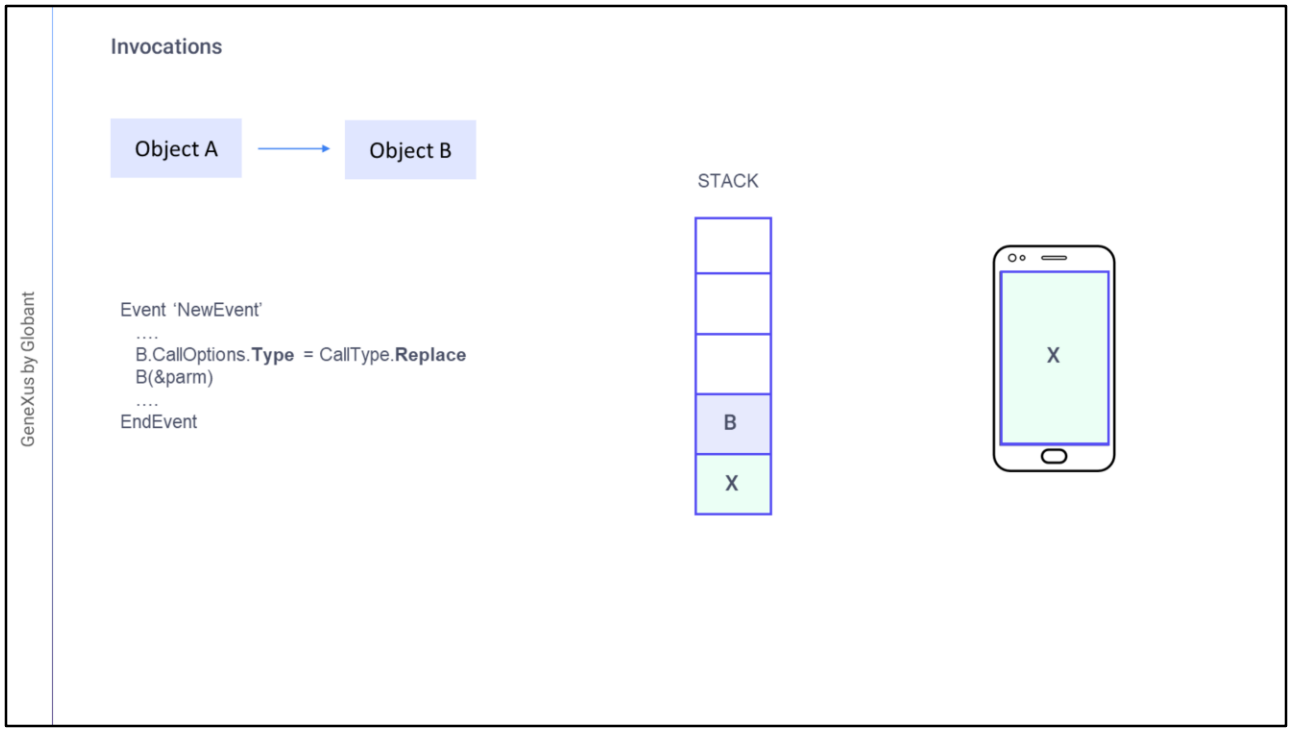
Cuando elegimos alguna de estas dos opciones, aparece la otra CallOption: CallTargetSize, para indicar el tamaño de la pantalla Popup o Callout, con las opciones Small, Medium y Large



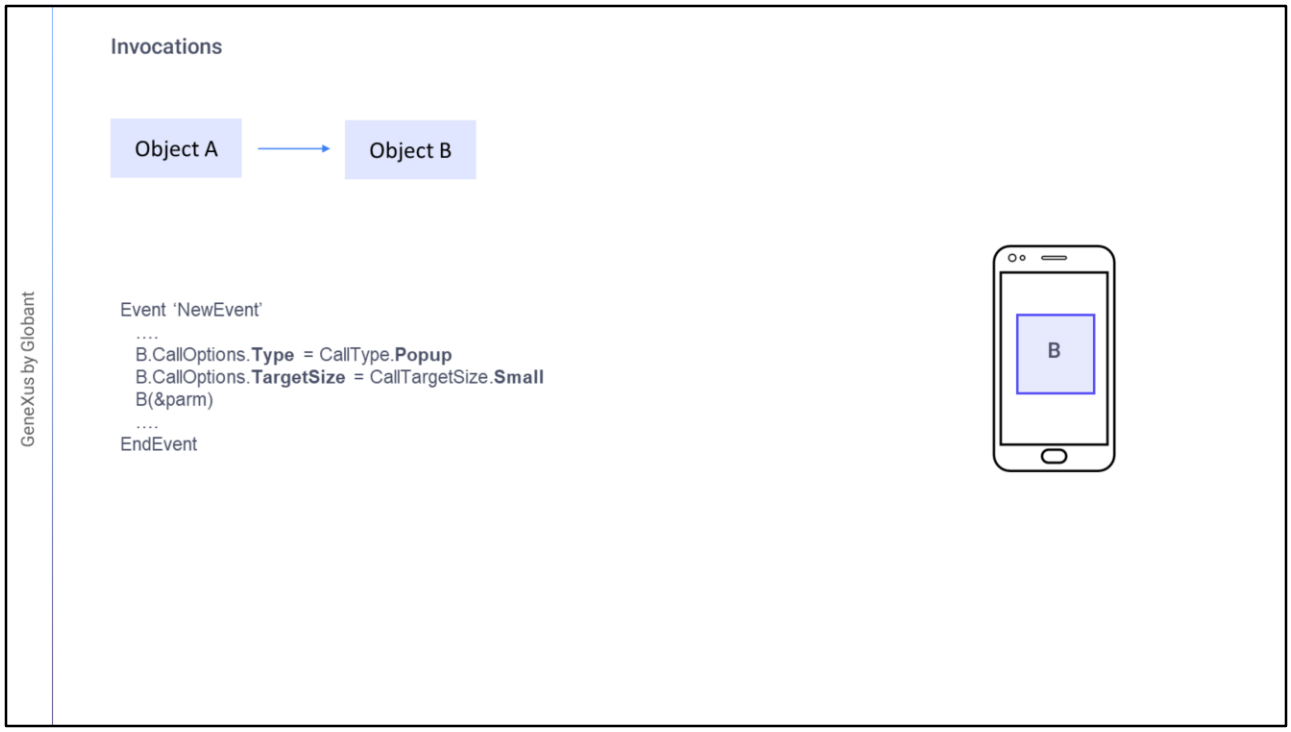
Respecto a los tipos Push y Replace, para entender el funcionamiento de ambos, veamos un pequeño ejemplo:

Supongamos que un objeto X, llamó a un objeto A.
Si ahora desde A en un evento, llamamos a un objeto B, con el tipo de call Push, el objeto llamado es colocado arriba en el stack.

Su pantalla se abre sobre la pantalla del llamador, ocupando exactamente el mismo lugar. Y el llamador espera para continuar su ejecución, a que termine la ejecución del objeto llamado: B que es así eliminado del stack.



Si en cambio desde el objeto A, llamamos a B con el tipo de call Replace, el objeto llamado también se abrirá ocupando exactamente la misma área de pantalla que el llamador, pero va a sustituir en el stack al objeto llamador. Por lo que, cuando termine su ejecución no volverá a continuar la ejecución del evento de A, sino que volverá al objeto que estuviera antes en el stack; en este caso, el objeto: X

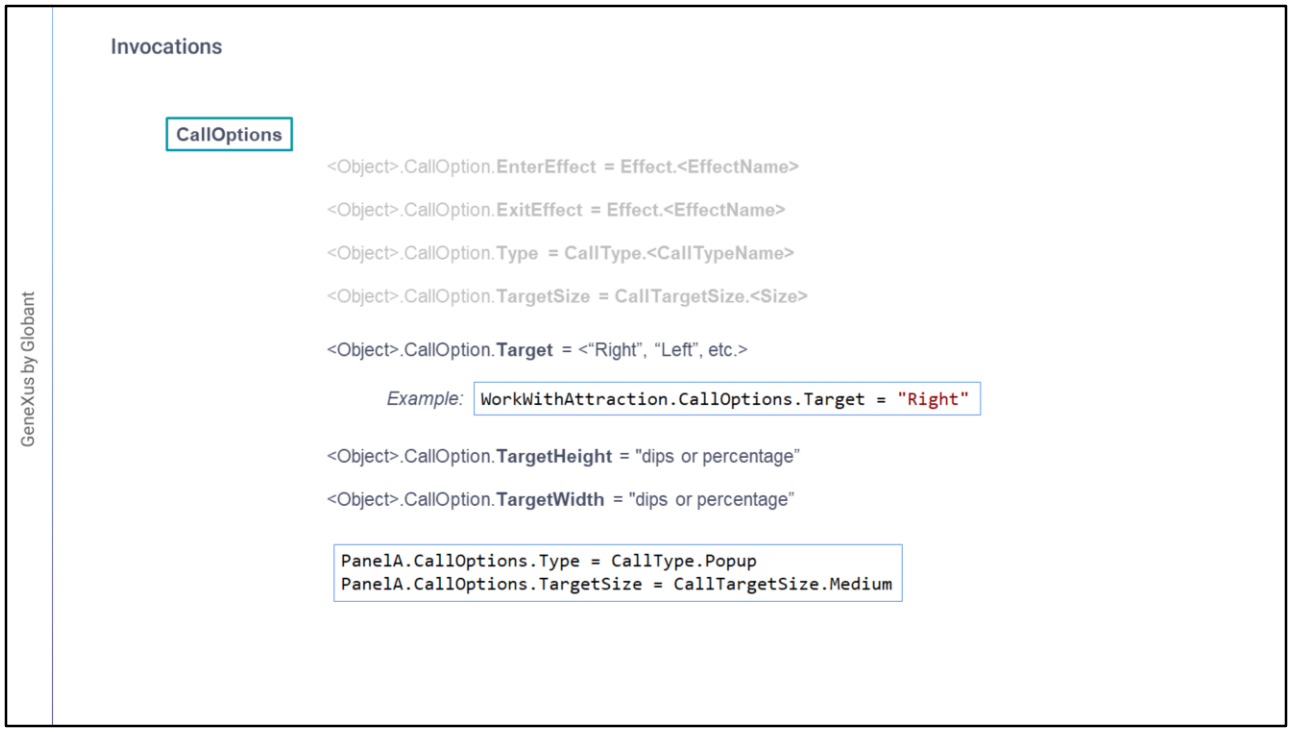


Ahora respecto a Popup y Callout, veamos el tipo Popup.

Si no se modifica el TargetSize, ocupará la misma área de pantalla que el llamador.
En caso contrario ocupará el área que hayamos especificado.

Si entre los parámetros de invocación alguno es output, entonces el diálogo será modal,
es decir, el llamador va a esperar el retorno de la ejecución de B para continuar

Si ninguno de los parámetros es de output, entonces el diálogo no será modal.



Volviendo a las calloptions, además tenemos:

- El target, donde se desplegara el objeto llamado, para indicar en cuál de los targets posibles se desea cargar la pantalla llamada

Esto tiene sentido cuando se trata de estilos de navegación donde existen múltiples targets para una misma pantalla, o sea pantalla partida, más utilizado en tabletas que en teléfonos. No puede utilizarse con Callout o Popup.

- Y de manera personalizada podemos definir el tamaño de la ventana llamada, especificando el Ancho y Alto, en dips o porcentajes, usando las propiedades TargetWidth y TargetHeight

Por ejemplo si quisiéramos llamar un panel como Popup con un tamaño Medium podríamos configurar las propiedades Type y TargetSize como vemos en pantalla y luego hacer el llamado al objeto normalmente.


GeneXus by Globant
GeneXus™
by Globant
training.genexus.com