GeneXus by Globant

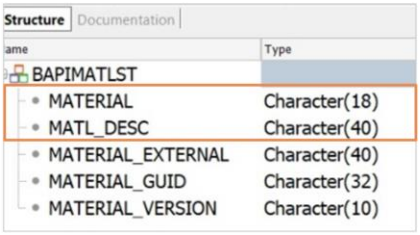
Integrando la KB a SAP ERP

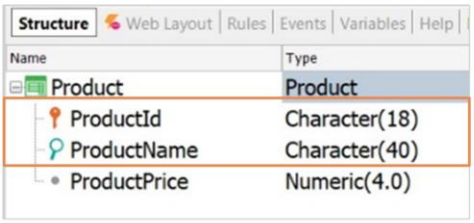
Ejecutando el método GetList

Anteriormente, hemos establecido la conexión con el ERP de SAP e invocado al método GetList de la Bapi de Materiales para recibir la lista de materiales.

Nuestro objetivo ahora es recorrer esa lista de materiales y volcarla en la tabla Producto de nuestra aplicación.

Ejecutando el método GetList





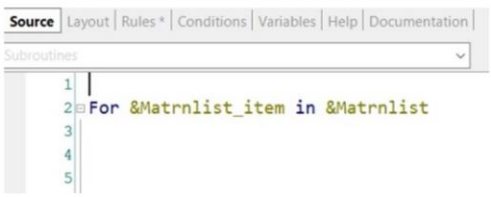
En primer lugar, observemos que en la definición de nuestra transacción Producto el Id es autonumerado, y el nombre es Character de largo 20.

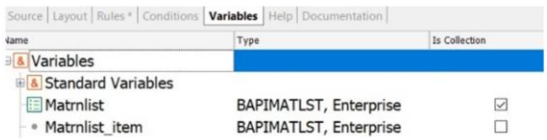
Pero si miramos ahora la estructura del SDT BAPIMATLIST vemos que el número asociado al material está definido como un Character de largo 18 y su descripción como Character de 40.

Así que modificamos la estructura de nuestra transacción para que sea compatible con estas definiciones.

Bien. Debemos ahora definir el proceso que se encargue de recorrer la lista de materiales recibida del ERP y volcarla en la tabla de producto

Ejecutando el método GetList





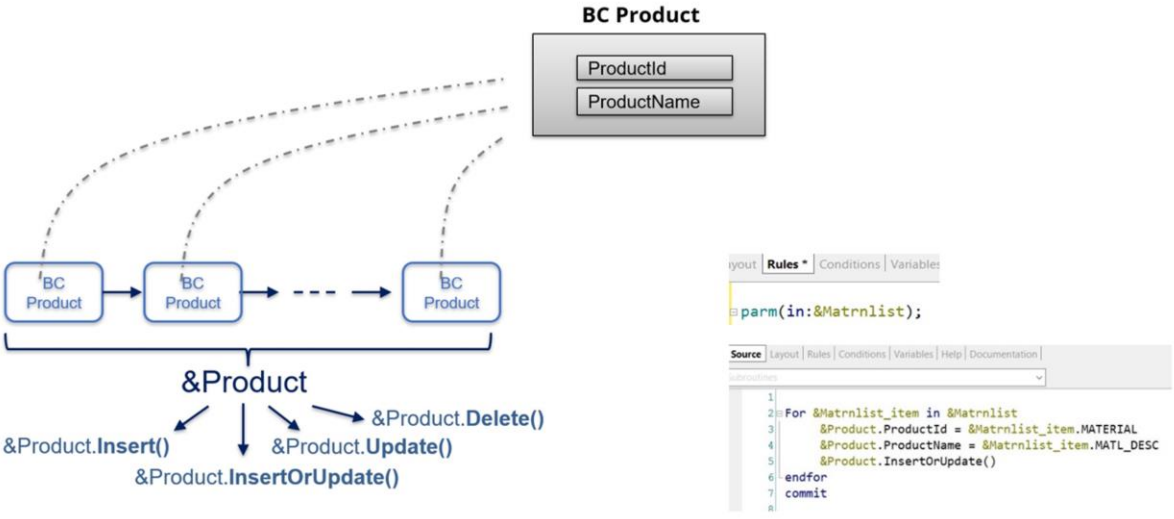
Creamos un procedimiento de nombre UserUpdateProducts.

Este procedimiento debe recibir la lista de materiales devuelta por la ejecución del método GetList, así que difinimos la variable &MatRnList de tipo de dato BapiMatList, y la marcamos como colección.

Vamos a la solapa Rules y declaramos la correspondiente regla Parm. Necesitamos recorrer esta colección de materiales, así que debemos definir otra variable de tipo BapiMatList pero esta vez para representar a un elemento de la colección, así que la definimos simple y no como colección

Bien. Vamos ahora al source para recorrer la lista de materiales. Y para cada uno de estos elementos, debemos insertarlo como un producto nuevo en nuestra tabla, si es que no existe previamente, o actualizarlo si es que ya existe. Así que declaramos nuestra transacción Producto como Business Component, y utilizaremos el método InsertOrUpdate para procesar los registros.

Business Component: Product



Recordemos que Business Component es una propiedad de las transacciones que habilita a disparar su lógica fuera de la transacción

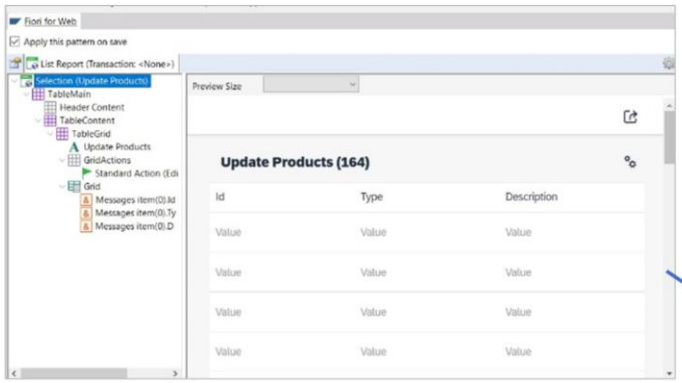
Los métodos que estamos viendo aplican tanto a variables simples como colección de Business Components. En particular el método InsertOrUpdate tiene un comportamiento particular, ya que primero intenta hacer el insert del registro, y si falla por clave duplicada, entonces lo recupera y lo actualiza.

Este es el comportamiento que nos sirve para resolver nuestro requerimiento ya que nuestro objetivo es mantener siempre actualizada nuestra tabla de productos, insertando los nuevos o actualizando los ya existentes.

Así que volvemos al procedimiento, y definimos la variable &Product Business Component.

Y en el source completamos el código. Una vez asignados los valores para el Id y nombre del producto, le aplicamos el método InsertOrUpdate. Cerramos el for, y declaramos el commit, necesario al trabajar con Business Componets.

Ejecutando el método GetList



Evento Start

```
SAPMaterialGetList(&Matrnlist, &Messages)

if &Messages.Count = 0
     UpdateMyProducts(&Matrnlist)
     msg("Products were updated")
else
     msg("Errors in the execution")
endif
```

Bien. Llegado a este punto, necesitamos ahora un objeto que dispare estos procesos. Así que creamos un web panel de nombre UpdateUserProducts y le aplicamos el pattern Fiori for Web. Elegimos el floorplan LisReport, y vamos a indicar que cargamos sus datos a partir de un SDT.

Recordemos que el proceso SAPMaterialGetList devuelve la lista de materiales y la lista de posibles errores ocurridos. La lista de materiales la manejamos internamente para actualizar nuestros productos, así que tomamos el sdt Messages como origen de datos.

Este web panel entonces mostrará la colección de errores, en caso de registrarse alguno, y procesará la lista de materiales.

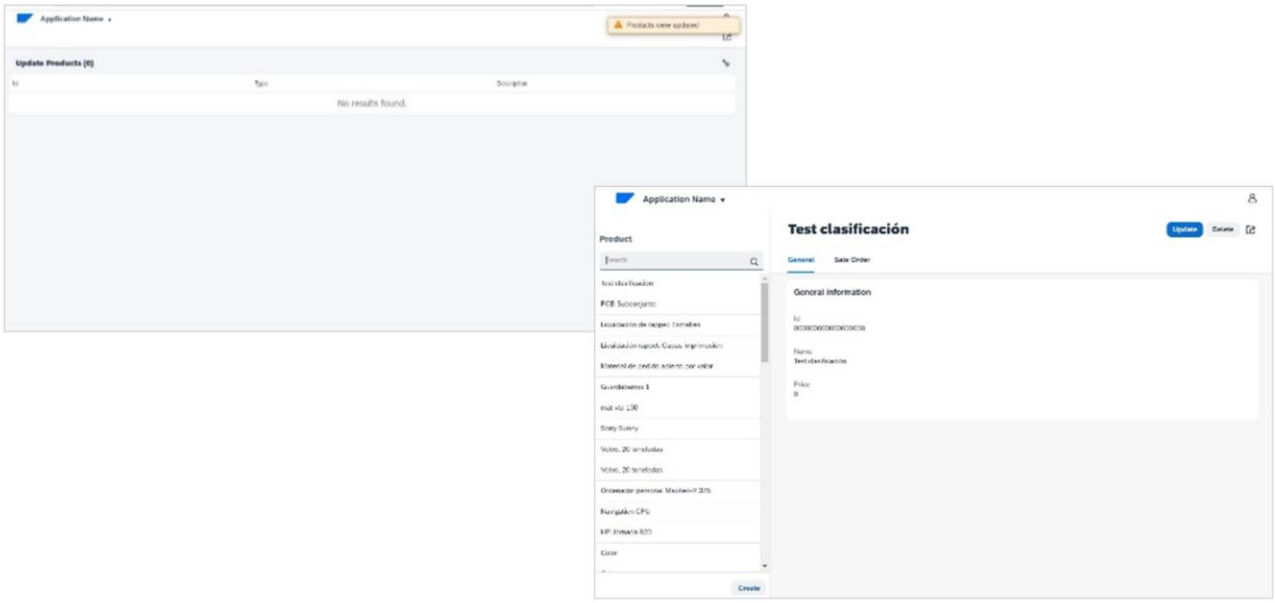
Quitamos las acciones sobre el grid, y verificamos las propiedades correspondientes para incluirlo en el Fiori lanunchpad y en el menú de la master page. Grabamos. Y vamos ahora a sus eventos.

Queremos que al abrir este objeto se llame al proceso que invoca al método GetList, se muestren los errores si los hubo, y se procese la lista de materiales.

Así que en el evento Start definimos la llamada al procedimiento SAPMaterialGetList, definiendo las variables necesarias.

Luego consultamos si hubo errores o no, y procesamos los materiales mostrando mensajes que indiquen si hubo fallas o se realizó correctamente.

Ejecutando el método GetList



Antes de ejecutar, nos aseguramos de eliminar los datos de prueba para cargar los productos a partir de los materiales devueltos por el ERP.

Presionamos F5 Primero verificamos que no tenemos ningún producto cargado. Y ahora vamos al web panel para cargar los materiales devueltos por el ERP. Vemos que se ejecutó correctamente y no hay devolución de errores ni advertencias.

Y finalmente volvemos a los productos para verificar que se cargaron correctamente. A continuación, trabajaremos con las órdenes de venta.

GeneXus by Globant

GeneXus™
by Globant

training.genexus.com