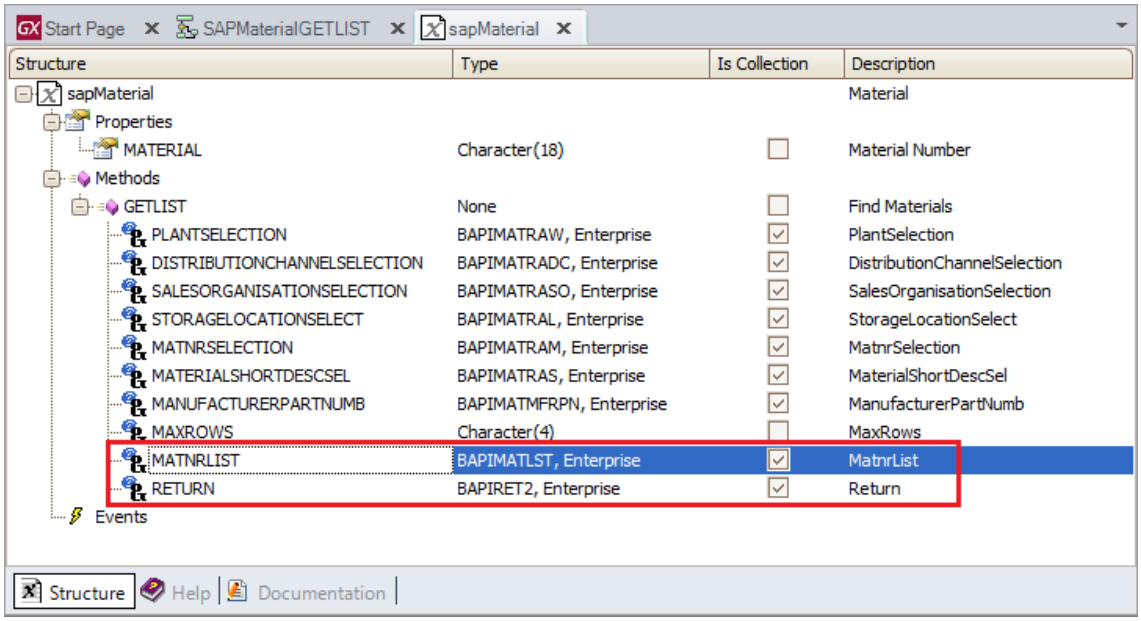
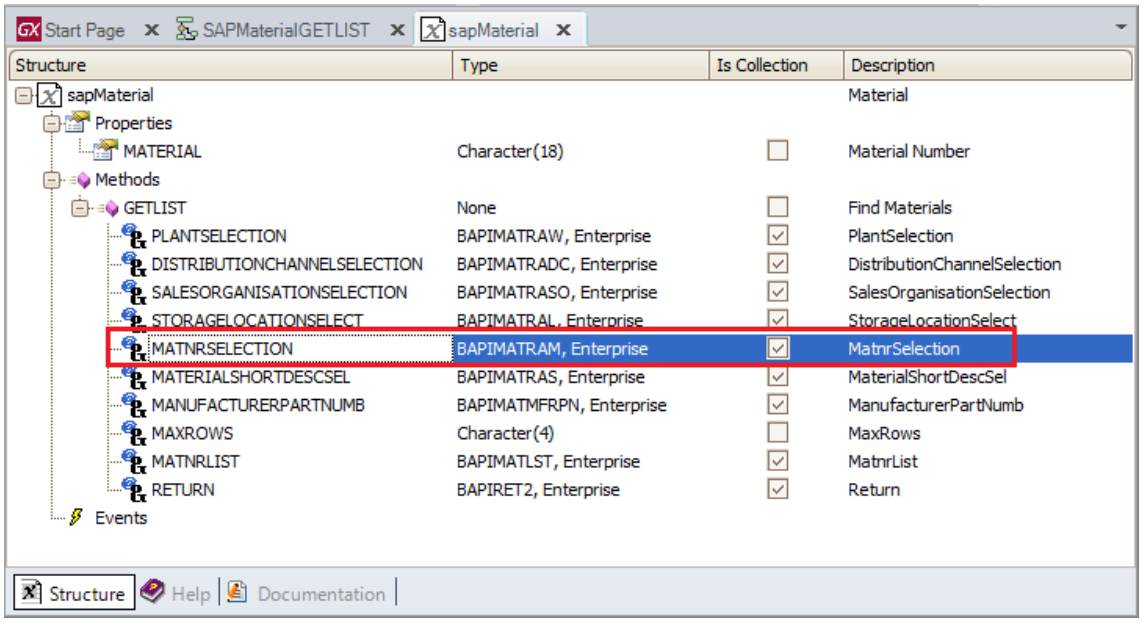
Creando una aplicación móvil con GeneXus para trabajar con los Materiales del SAP ERP – parte 4

Ahora sí, si no hubo error, querremos obtener el listado de materiales, por lo que tendremos que invocar al método GETLIST de sapMaterial. Vamos a tener que pasar todos estos parámetros al método… Estos últimos dos son los de salida:

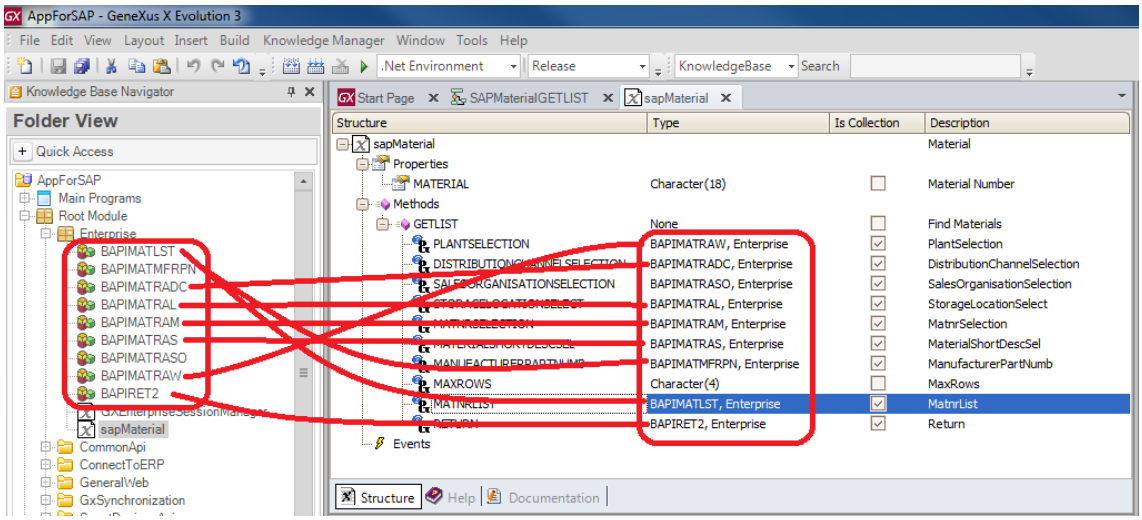


El penúltimo será la lista de materiales y el último la colección de mensajes de advertencia y error, entre otros, que se hayan producido.

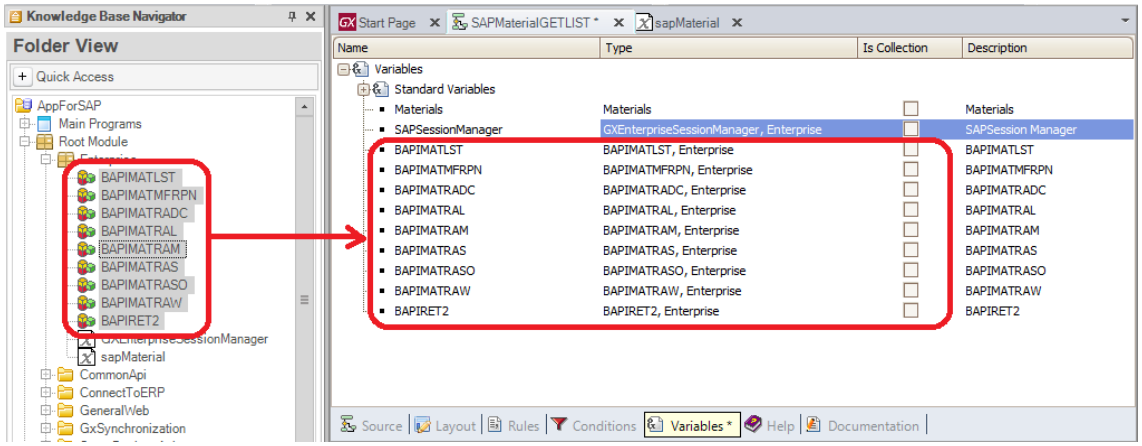
Los demás parámetros serán de entrada; en nuestro caso irán casi todos vacíos, pues no queremos filtrar por planta, canal de distribución, organización, y demás. Pero como queremos recuperar todos los materiales, debemos cargar el parámetro MATNRSELECTION como lo hicimos cuando testeamos el método a través del conector.



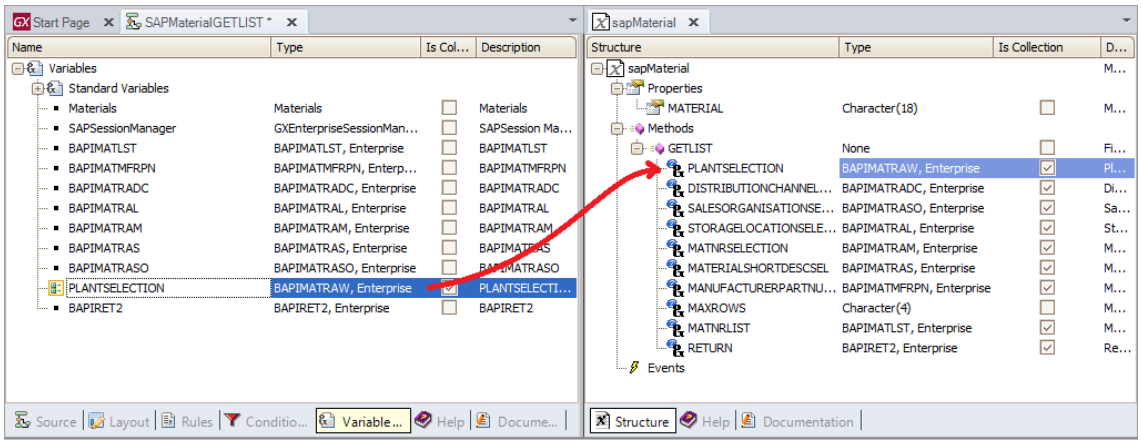
Lo que haremos entonces será definir en nuestro procedimiento variables para cada parámetro del método, que, como vemos, son de los tipos de datos importados con la bapi, más un carácter de 4.



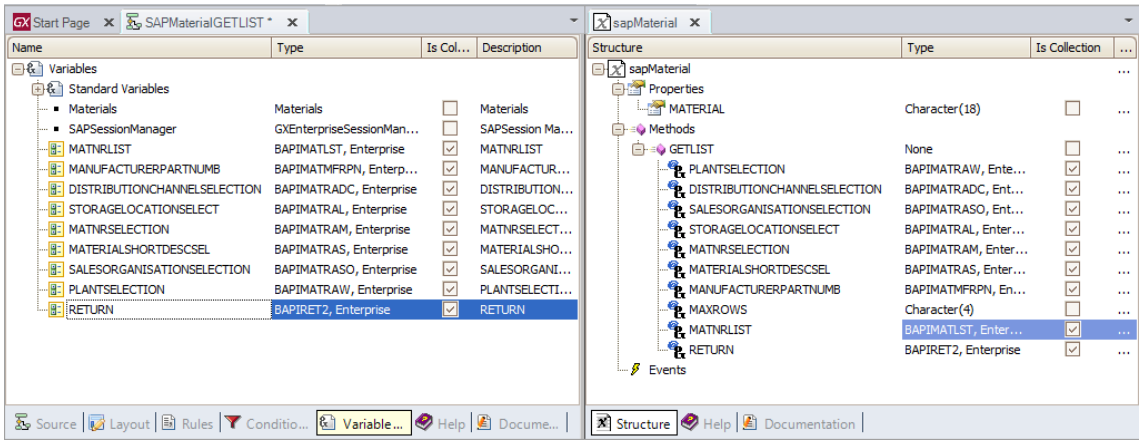
Para ello, en nuestro procedimiento arrastramos a la sección de variables los tipos de datos estructurados y vemos que crea variables con igual nombre. Podemos cambiárselos, para que nuestro código quede más claro, asumiendo los mismos nombres que los parámetros del método.



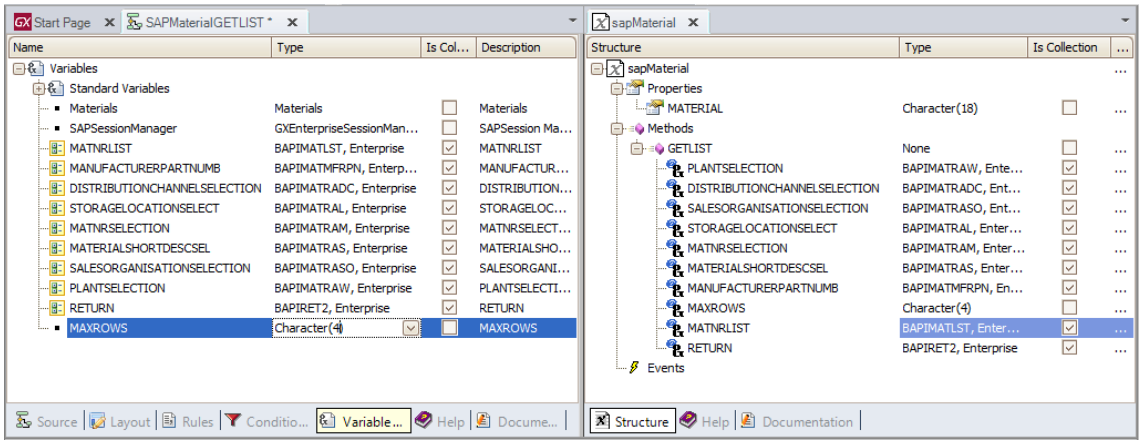
Así, vemos que PLANTSELECTION es del tipo de datos BAPIMATRAW pero no simple, sino colección, es decir, una colección de ese tipo de datos. Así que marcamos la variable de igual manera, y le cambiamos el nombre.



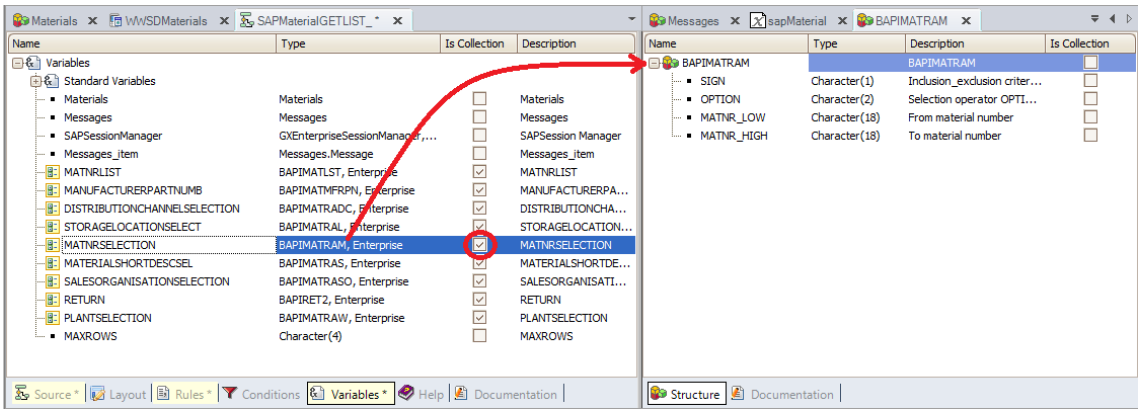
Y hacemos lo mismo con las demás…



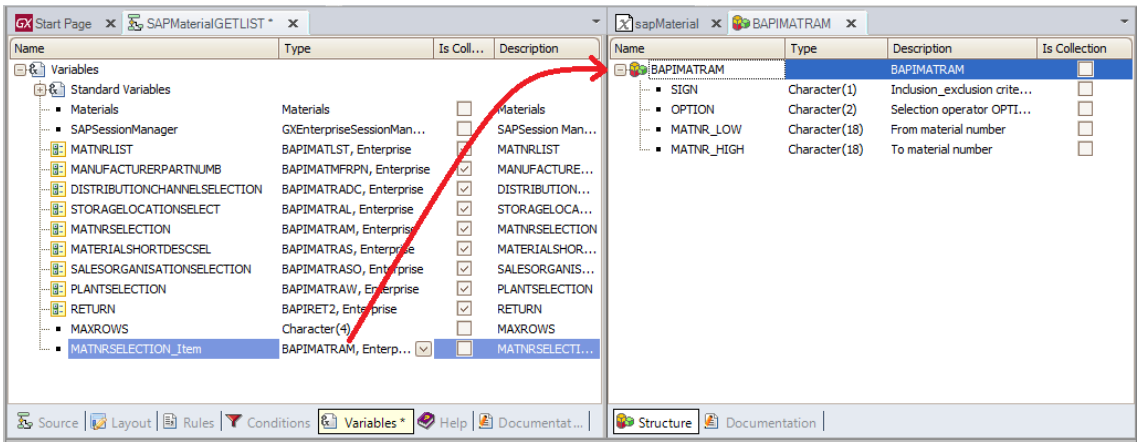
Agregamos, además, la variable MAXROWS, de tipo Character de 4.



Ya tenemos las variables definidas. Ahora, antes de llamar al método pasándole esas variables, tenemos que cargar la variable MATNRSELECTION, para especificar que queremos que nos devuelva la lista completa de materiales. Vemos que es una colección de ítems de tipo BAPIMATRAM, pues permite definir varios ítems con condiciones, para que las aplique en el filtro resultante.



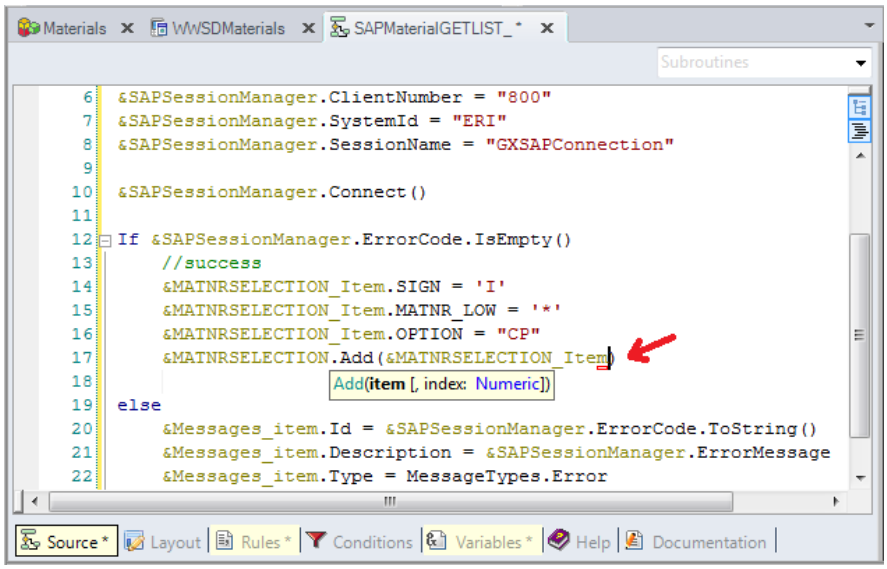
Nosotros sólo necesitaremos que la colección tenga un ítem. Nos tendremos que definir una variable de este tipo de datos (el del ítem de la colección). Por tanto no colección.



Luego en el Source del procedimiento, a esta variable, propiedad SIGN le asignamos “I” de inclusive, a la propiedad que contiene el From del material number, le asignamos el wildcard asterisco, y a la opción de filtro le asignamos CP, por “Contain Pattern”:

```
If &SAPSessionManager.ErrorCode.IsEmpty()
	//success
	&MATNRSELECTION_Item.SIGN = 'I'
	&MATNRSELECTION_Item.MATNR_LOW = '*'
	&MATNRSELECTION_Item.OPTION = "CP"
else
	&Messages_item.Id = &SAPSessionManager.ErrorCode.ToString()
	&Messages_item.Description = &SAPSessionManager.ErrorMessage
	&Messages_item.Type = MessageTypes.Error
	&Messages.Add(&Messages_item)
endif
```

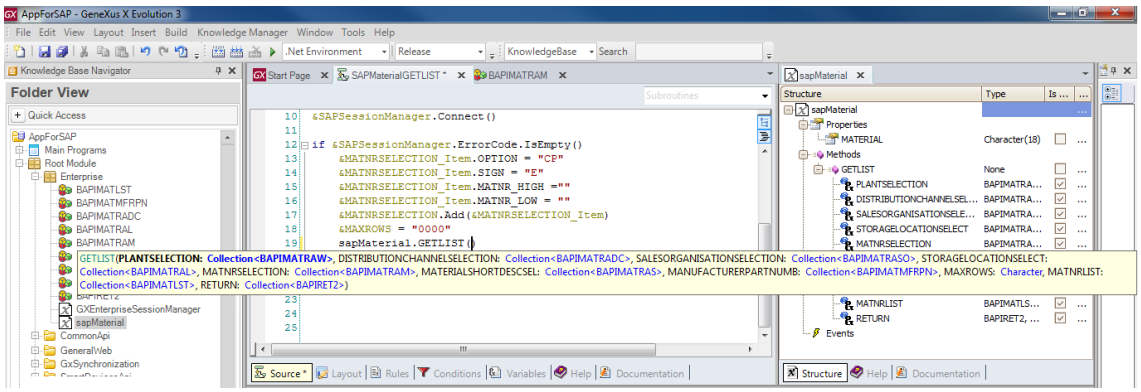
Por último, agregamos el contenido de esta variable a la variable colección (hasta ahora vacía) &MATNRSELECTION, que pasaremos por parámetro al método:



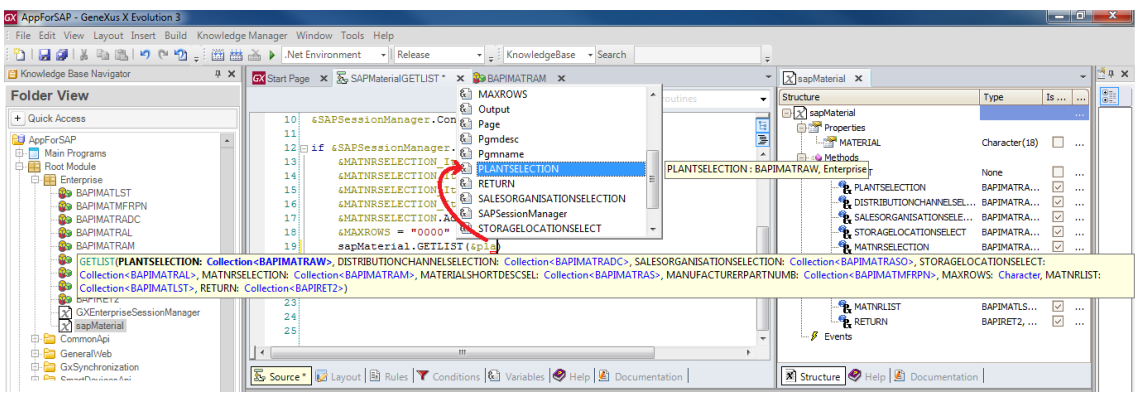
Cargamos también la variable &MAXROWS con una cadena de cuatro ceros, para que nos devuelva todos los materiales, sin restricciones:

```
&MATNRSELECTION_Item.SIGN = 'I'
&MATNRSELECTION_Item.MATNR_LOW = '*'
&MATNRSELECTION_Item.OPTION = "CP"
&MATNRSELECTION.Add(&MATNRSELECTION_Item)
&MAXROWS = "0000"
```

y ahora sí, estamos en condiciones de invocar al método de la BAPI. Arrastramos para no tener que digitar, y al escribir punto, GeneXus automáticamente escribe el único método que tiene el objeto externo y ya nos muestra en un intellitip todos los parámetros requeridos, resaltando el que debemos escribir en esta oportunidad, PLANTSELECTION:



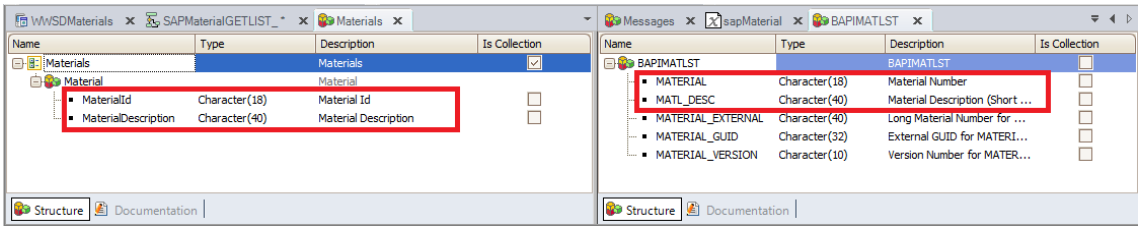
Elegimos la variable &PlantSelection:



Luego, la siguiente, &DISTRIBUTIONCHANNELSELECTION, y así…

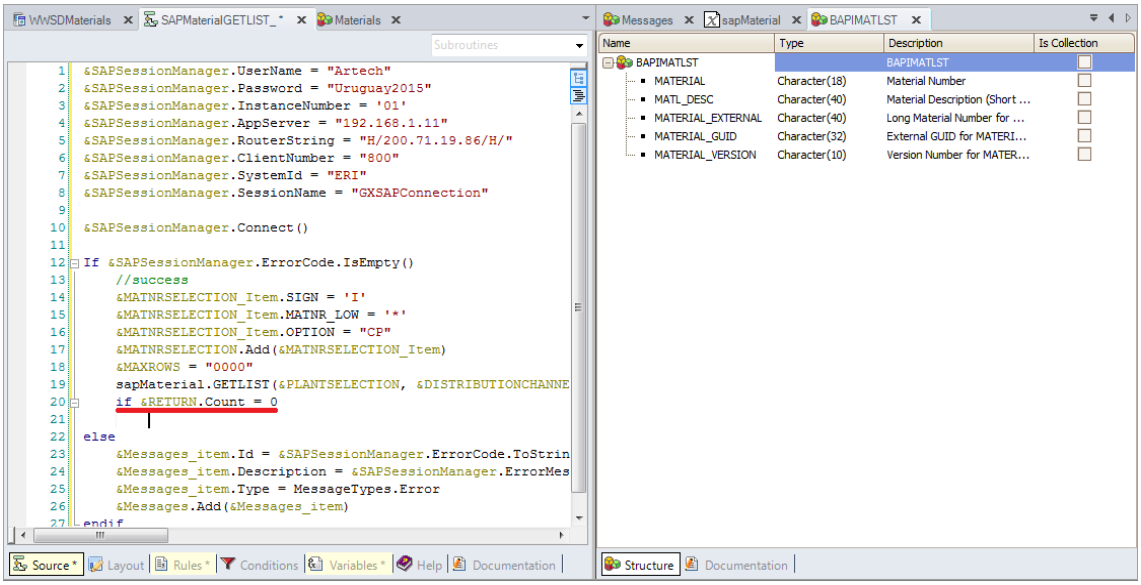
Las variables &MATNRLIST y &RETURN, al terminarse de ejecutar la bapi, quedarán cargadas con la colección de materiales y de mensajes, respectivamente.

&MATNRLIST es una colección de ítems de este tipo, BAPIMATLST:

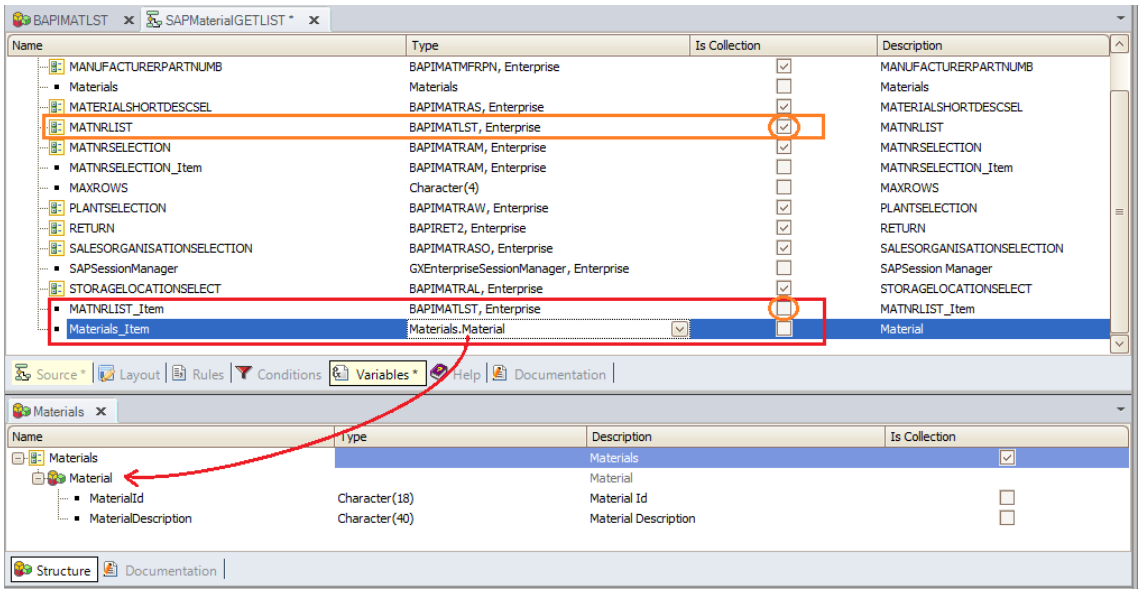


Y nuestra variable &Materials es del tipo de datos que creamos para quedarnos sólo con el id y la descripción.

Siendo esta la variable que devolverá el procedimiento, tenemos entonces que cargarla copiando los ítems de una colección a la otra, pero esto queremos hacerlo únicamente si el GETLIST fue exitoso. Lo sabemos, como vimos cuando testeamos la BAPI a través del conector, viendo si la colección de mensajes que vuelve cargada en la variable &RETURN, está vacía, es decir, si no tiene ítems. Entonces:



Tenemos que recorrer cada ítem de la colección de materiales devuelta y crear un nuevo ítem en la de &Materials. Por lo que tenemos que crearnos dos variables más, cuya estructura sea la de los ítems... MATNRLIST_Item, del tipo el mismo que MATNRLIST pero simple, no colección. Y Materials_Item, del tipo Materials.Material.



Utilizaremos la variable MATNRLIST_Item para iterar sobre la colección de materiales devuelta. Para cada ítem de la colección:

```
for &MATNRLIST_Item in &MATNRLIST
```

a la variable vacía &Materials_Item, para el MaterialId le asignamos el valor correspondiente del ítem actual en el que estamos posicionados:

```
for &MATNRLIST_Item in &MATNRLIST
      &Materials_Item.MaterialId = &MATNRLIST_Item.MATERIAL
```

Y a MaterialDescription le asignamos la descripción que se encuentra en este campo del ítem actual:

```
for &MATNRLIST_Item in &MATNRLIST
      &Materials_Item.MaterialId = &MATNRLIST_Item.MATERIAL
      &Materials_Item.MaterialDescription = &MATNRLIST_Item.MATL_DESC
```

Ya podemos agregar a la colección &Materials que se inicializa vacía, el contenido de la variable que acabamos de cargar:

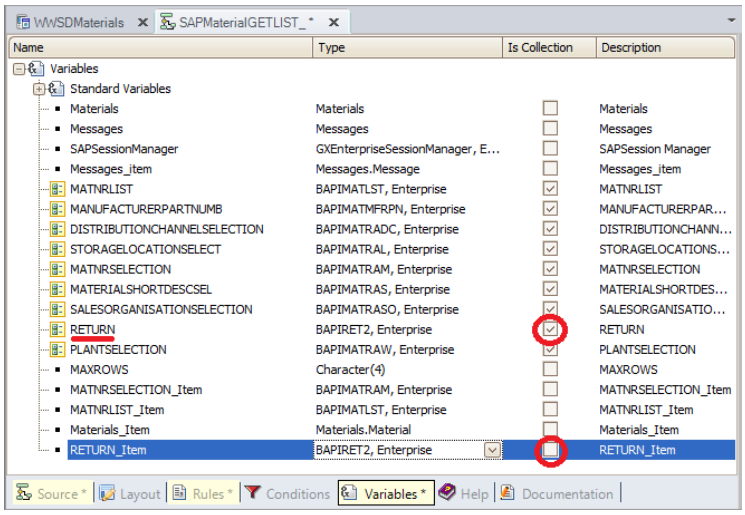
```
for &MATNRLIST_Item in &MATNRLIST
    &Materials_Item.MaterialId = &MATNRLIST_Item.MATERIAL
    &Materials_Item.MaterialDescription = &MATNRLIST_Item.MATL_DESC
    &Materials.Add(&Materials_Item)
```

Add(**item** [, index: Numeric])

Reservamos nueva área de memoria para la variable &Materials_Item, para el siguiente material a ser agregado… y finalizamos el comando for.

```
for &MATNRLIST_Item in &MATNRLIST
    &Materials_Item.MaterialId = &MATNRLIST_Item.MATERIAL
    &Materials_Item.MaterialDescription = &MATNRLIST_Item.MATL_DESC
    &Materials.Add(&Materials_Item)
    &Materials_Item = new()
endfor
```

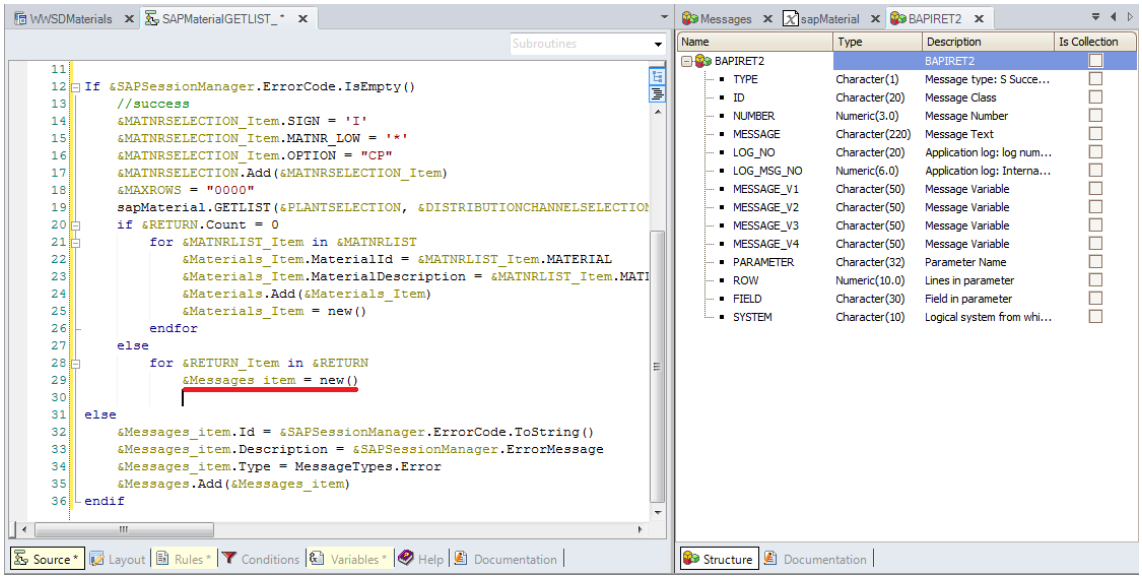
Sólo nos resta programar qué queremos hacer si hubo mensajes. Al igual que hicimos antes, los agregaremos en la variable de salida &Messages. Los mensajes vuelven cargados del método GETLIST de la bapi en la variable colección &RETURN. Así que tendremos que recorrer esa colección, utilizando una variable del tipo de datos de los ítems:



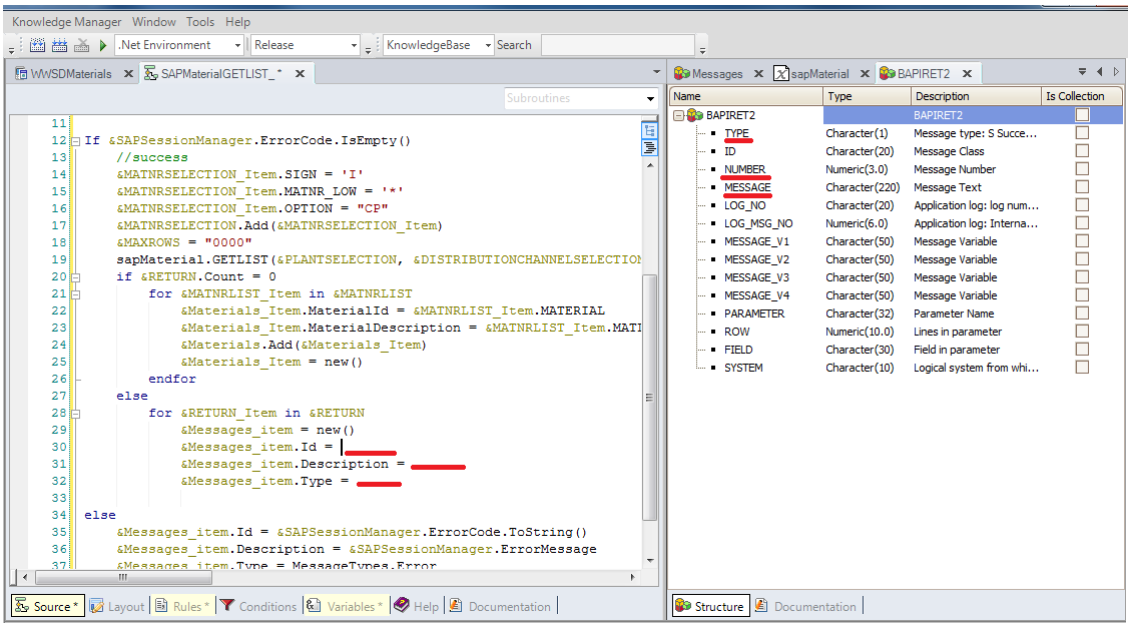
Para cada ítem de la colección:

```
else
        for &RETURN_Item in &RETURN
```

reservamos espacio de memoria para la variable &Messages_item que ya habíamos utilizado:

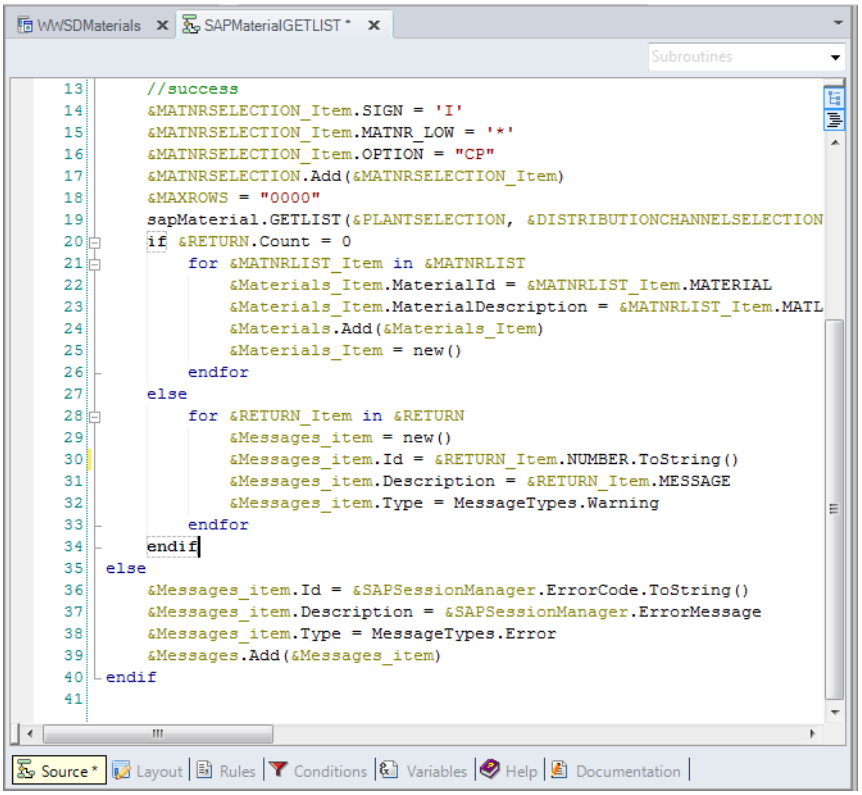


Y asignamos valor a sus propiedades:



de acuerdo a las del ítem actual. Como Id es varchar y NUMBER es numérico, tenemos que transformarlo a string.

Para el Type deberíamos reducir los tipos posibles del return a Error o Warning. Por simplicidad aquí pusimos directamente Warning. Cerramos el for y cerramos el if, y grabamos:



En el video siguiente completaremos el panel que invoca a este procedimiento y lo veremos en ejecución.

Video filmado con GeneXus X Evolution 3 – upgrade 3