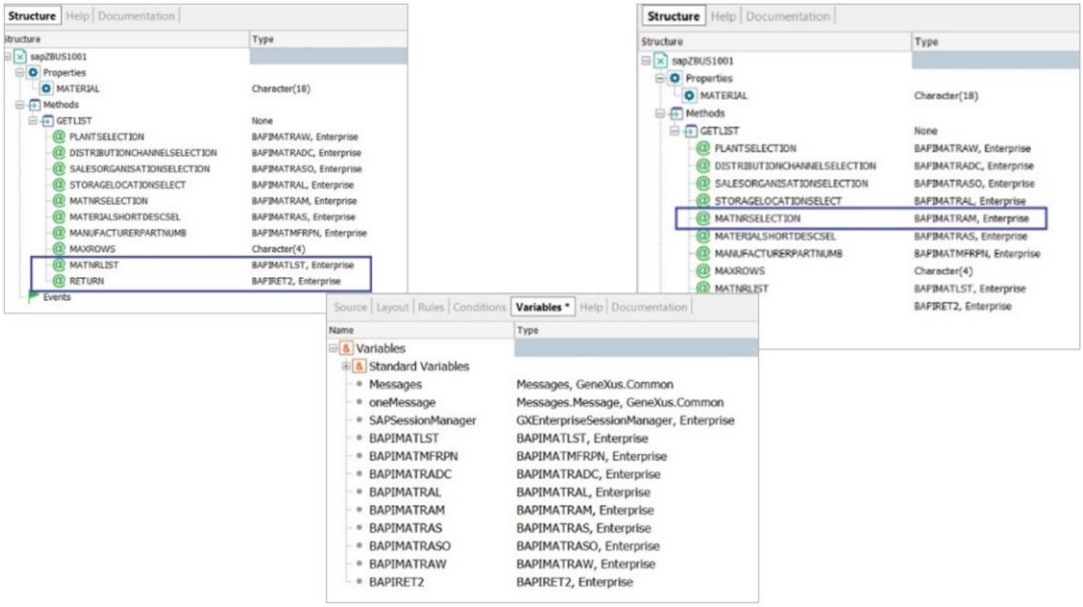
GeneXus by Globant

Integrating the KB to SAP ERP

Calling the GetList method

Moving forward in our example, if there were no errors in the connection with the ERP, we now want to get the list of materials. Therefore, we will have to invoke the GETLIST method of the external object corresponding to the Material BAPI.

Calling the GetList method



We will have to pass all these parameters to the method... Also, remember that the last ones are output ones.

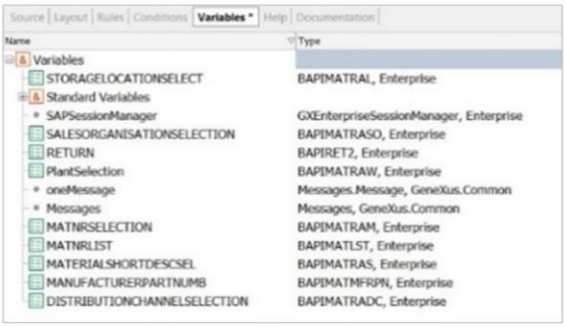
The second to last one will be the list of materials and the last one will be the collection of generated warning and error messages, among others.

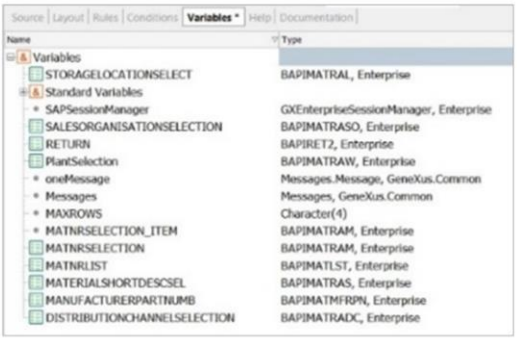
The other parameters will be input parameters. In this case, they will be almost all empty, because we don't want to filter by plant, distribution channel, organization, and so on. However, since we want to retrieve all the materials, we must load the MATNRSELECTION parameter as we did when we tested the method through the connector.

Then in our procedure, we will define variables for each parameter of the method. As we see, they are of the data types imported with the BAPI plus a character data type of length 4.

So, from the variables section of our procedure, we drag the structured data types and see that variables with the same name are automatically created.

Calling the GetList method





OK. The parameters of the GetList method are based on structured data types, but not as simple elements but as collections, and so we must also indicate this in our variables.

We are also going to change the variables' names to make our code clearer:

In addition, we add the &MAXROWS variable, of Character (4) data type.

Good. We already have the variables defined. Now, before calling the method by passing these variables, we have to load the MATNRSELECTION variable to specify that we want the entire list of materials to be returned. We see that it is a collection of items of BAPIMATRAM type because it allows defining several items with conditions, in order to apply them in the resulting filter.

But we only need the collection to have one item. We will have to define a variable of this data type (that of the item of the collection). That is to say, no collection.

Then in the procedure Source we assign a value to the properties of this variable:

- To the SIGN property, we assign the value "I" for inclusive.
- To the property containing the From of the material number, we assign the wildcard "*".
- And to the filter option we assign "CP", which stands for "Contain pattern".

Finally, we add the content of this variable to the &MATNRSELECTION collection variable (empty so far), which we will pass by parameter to the method.

Calling the GetList method

We also load the &MAXROWS variable with a string of four zeros, so that it returns all the materials without restrictions.

Now we are ready to invoke the BAPI method. We drag to avoid typing, and when we enter a period, GeneXus automatically writes the only method of the external object. Also, it shows all the required parameters in an intellitip, highlighting the ones we have to type in each time.

When the BAPI execution is finished, the &MATNRLIST and &RETURN variables will be loaded with the collection of materials and messages, respectively.

We only have to program what we want to do if there were any messages. As we did before, we will add them in the &Messages output variable. The returned messages are loaded from the GETLIST method of the BAPI in the &RETURN collection variable. So we will have to run through that collection, using a variable of the items data type:

We define this variable, and do the iteration by assigning the values to the properties according to the data type of the current item.

For the Type, and for the sake of simplicity, we only set "Warning". We close the for, close the if, and save:

Finally, in this procedure we define the corresponding Parm rule to return the collection of materials of the &MATNRLIST variable, and the collection of messages of the &Messages variable.

Next, we will implement the necessary process to receive the collection of materials returned by this procedure and save it in the Product table of our application.

GeneXus by Globant

GeneXus™
by Globant

training.genexus.com