

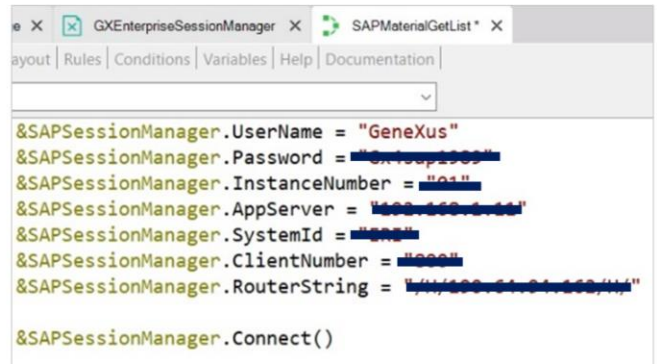
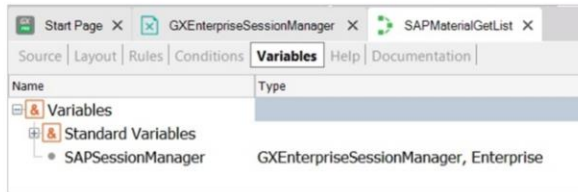
Integrating the KB to SAP ERP

Connecting with the ERP

After importing the corresponding BAPI to execute the Materials GetList method, we have seen the objects that were integrated into our knowledge base.

Now we are going to create a procedure that allows us to define the connection with the ERP and invoke the GetList method to receive the list of materials.

Connecting with the ERP



In this example, and for the sake of simplicity, we will use specific information. Later on, we will see other login options.

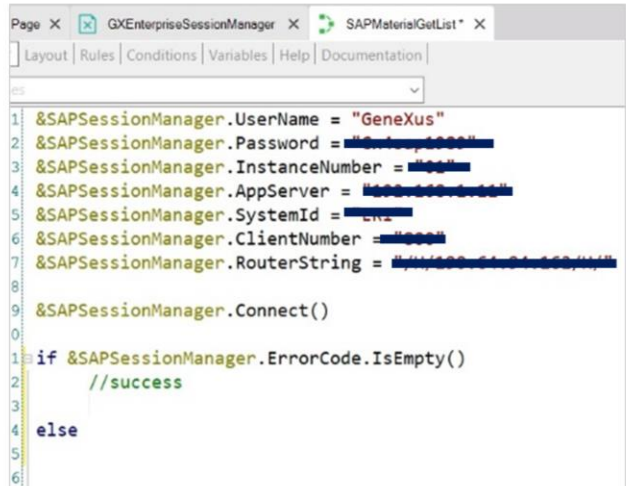
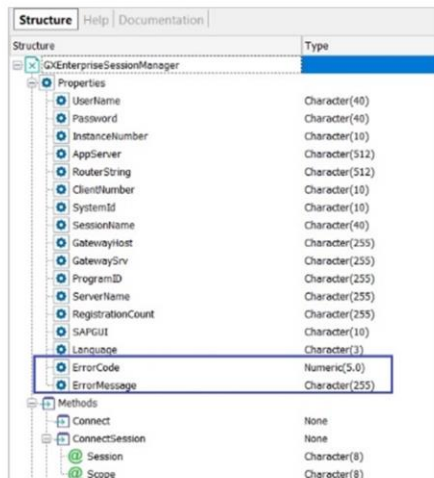
We create a procedure named SAPMaterialGetList. The first thing we will do is define the SAPSessionManager variable of the GXEnterpriseSessionManager data type. Remember that this is one of the external objects of the Enterprise module, automatically created in the BAPI import process.

Defining a variable based on the data type of an external object causes it to be defined with the data structure of that object. In this case, the properties UserName, Password, InstanceNumber, AppServer, etc., correspond to the ERP connection data and also make it possible to apply their methods on this instance.

So we go to the procedure Source, and assign a value to this variable for all the connection properties, starting with UserName, the user we will use to connect to the ERP. Then we enter its password, instance number, and so on until all the connection parameters are completed.

Once the values have been assigned to the variable's properties, we execute the Connect method.

Connecting with the ERP

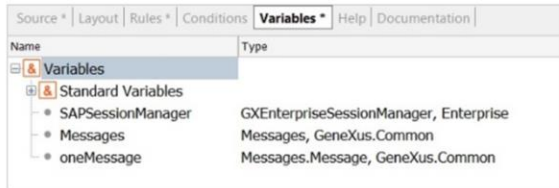


If the connection attempt fails, the generated error is loaded in the ErrorCode and ErrorMessage properties.

So we ask if there were no errors; that is, if ErrorCode is empty. There we will program the actions to take when the connection was successful: Otherwise—else—we want to somehow return and then display the error message, which is in the ErrorMessage property of the external object.

A good option is to use the Messages structured data type that is automatically created in every knowledge base, to be used precisely to store error and warning messages.

Connecting with the ERP



```

Page X  GXEnterpriseSessionManager X  SAPMaterialGetList X  sapZBUS1001 X  Domain
Layout | Rules | Conditions | Variables | Help | Documentation |
#5
1  &SAPSessionManager.UserName = "GeneXus"
2  &SAPSessionManager.Password = "GeneXus1000"
3  &SAPSessionManager.InstanceNumber = "0001"
4  &SAPSessionManager.AppServer = "SAP01"
5  &SAPSessionManager.SystemId = "001"
6  &SAPSessionManager.ClientNumber = "000"
7  &SAPSessionManager.RouterString = "http://sap01.sap.com:8000"
8
9  &SAPSessionManager.Connect()
10
11 if &SAPSessionManager.ErrorCode.IsEmpty()
12     //success
13
14 else
15     &oneMessage.Id = &SAPSessionManager.ErrorCode
16     &oneMessage.Description = &SAPSessionManager.ErrorMessage
17     &oneMessage.Type = MessageTypes.Error
18     &Messages.Add(&oneMessage)
19 Endif
20

```

Therefore, we need to define an output variable for our parameters, to return all the messages generated as a result of running the object:

We define it with the same name (Messages) so that it automatically takes the Messages data type. Now we have to load it into the procedure. Since it is a collection of items, which in turn is structured, we define another variable called &oneMessage to represent an item.

We return to the Source to load it. To the Id we assign the ErrorCode of &SAPSessionManager. But as we see, Id is of VarChar type and ErrorCode is Numeric, so we convert the Numeric type to String.

Then to Description we assign the ErrorMessage property. Lastly, we have to indicate the message type, which as we see has a predefined data type: MessageTypes, an enumerated domain.

All the domains defined in the knowledge base are seen here. The properties of MessageTypes have two values: Warning and Error.

The type in our case will be Error: What still needs to be done is to add this item to the message collection and close the block with Endif.

Since our objective is to obtain the list of materials, we must invoke the GETLIST method of the Material BAPI. That's what we will do next...

GeneXus[™]
by **Globant**

training.genexus.com