Module Pool Program

A Dialog program is a transaction which enters data in a Database System (or) retrieves data from the database tables and displays on the screen fields.

During a dialog program, System displays a screen on which user enters (or) request some information and selects push button (or) menu item then the result will be displaying the data onto the screen fields or updating the data to the database.

In other words a transaction is a sequence of screens that appears one after another depending upon user section.

A Dialog program contains two main components:

- Dynpro
- ABAP/4 module pool program

1. **Dynpro**: It's a dynamic program which contains screen and its flow logic and executes one dialog step.

Dialog Step: It is any screen change that is it includes all processing right from entering the data on to the first screen till you get the final screen.

There are four ways to trigger a dialog step.

1. press enter
2. select push button
3. Select Menu item.
4. press function key

Dynpro consists of four Components:

1. Screen flow logic
2. Screen layout
3. Screen Attribute
4. Field Attribute.

A. **Screen Flow Logic**: It is the code that processes a particular screen with the help of four events.

- ✓ PBO (Process Before Output)
- ✓ PAI (Process After Input)
- ✓ POH (Process on Help Request)
- ✓ POV (Process on Value Request)

1. **PBO (Process Before Output):** It is triggered before the screen is displayed. It is generally used to initialize the screen fields & is also used to attach the user interface & title.
2. **PAI (Process After Input):** This event is triggered when the user enters data onto the screen fields & clicks on push button or menu item. Then the result will be displaying the data on to the next screen or updating the data to the database.
3. **POH (Process on Help Request):** This event is triggered when the user press F1 function key on Screen field. This event is used to avoid or by-pass the standard help documentation provided by SAP through (data element). ABAP dictionary & provide the custom help documentation.
4. **POV (Process on Value Request):** It is triggered when the user press F4 function key on screen field. It is used to avoid the standard list of values provided by SAP through ABAP dictionary (Search Help) & provide the custom list of values.

B. **Screen Layout**: It is used to design the layout of the screen with the help of screen elements such as Input/output field, push button, check box, Radio buttons etc.

C. **Screen Attributes**: It defines the properties of the Screen such as normal screen, sub screen, model dialog box etc.

D. **Field Attributes**: It defines the properties of the field such as read only, required, optional, invisible etc.

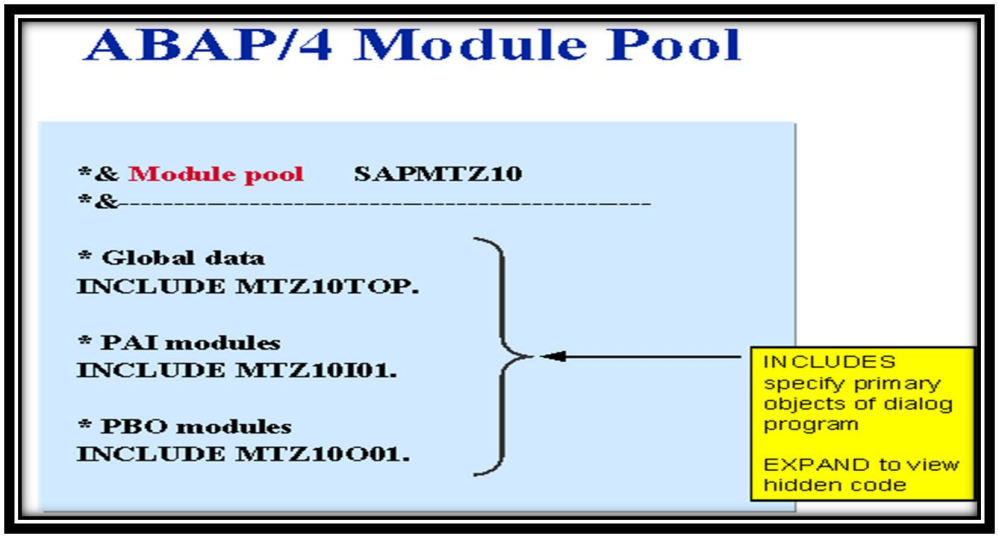
Screen: Screen is a graphical container where you can place screen elements such as Input/output field, text field, radio button, check box, push button etc.

- Screen is designed with the help of screen painter and the transaction code is SE51.
- Screen is represented by screen number & it ranges from 0 to 9999.
- Each screen is considered as a one logical unit of work (LUW). Each LUW of work will update the data to database.
- Group of logical unit of works is considered as SAPLUW.

2. ABAP/4 Module Pool: It contains data declaration & set of modules called by screen flow logic.

ABAP/4 module pool program is of program type 'M'. The naming convention for ABAP module pool program is SAPMZXXXX... or SAPMYXXXX…

Generally transaction or dialog program is created with the help of ABAP work bench tool known as Object Browser or Object Navigator & transaction Code is SE80.



In the Object Browser, the module pool code belongs to one of the following categories:

- **Global fields:** data declarations that can be used by all modules in the module pool.
- **PBO modules:** modules that are called before displaying the screen.
- **PAI modules:** modules that are called in response to the user input.
- **Subroutines:** subroutines that can be called from any position within the module pool.

By default, the system divides a module pool into one or several include programs. An include program can contain several modules of the same type (only PBO modules or only PAI modules).

The screen processor executes the screen where it executes the screen flow logic which contains two events PBO & PAI.

1. Screen processor executes the screen flow logic which triggers PBO event where it module is called so the control from screen flow logic is transferred to module pool program and the set of statements in the PBO module gets executed.

The set of statements can be initializing the screen fields, providing the user interface and also title bar etc.

2. Once end of PBO module encounter than the control from ABAP/4 module pool program is transferred back to the screen flow logic. After the PBO event triggered, screen is displayed to the user.
3. On to the available screen the user enters data and clicks on push button or menu item then the event triggered was PAI where its module is called.

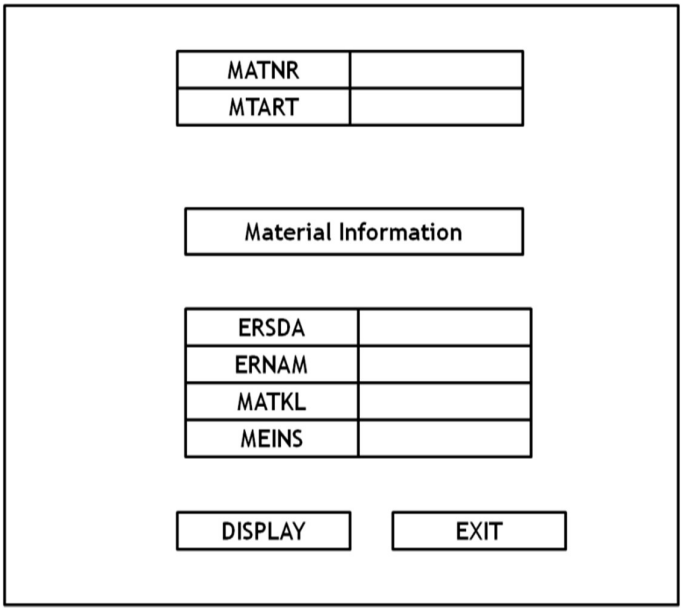
So, therefore the control from screen flow logic is transferred to module pool programs and the set of statements in that module gets executed.

The set of statements can be displaying the data on to the next screen or updating the data to the database.

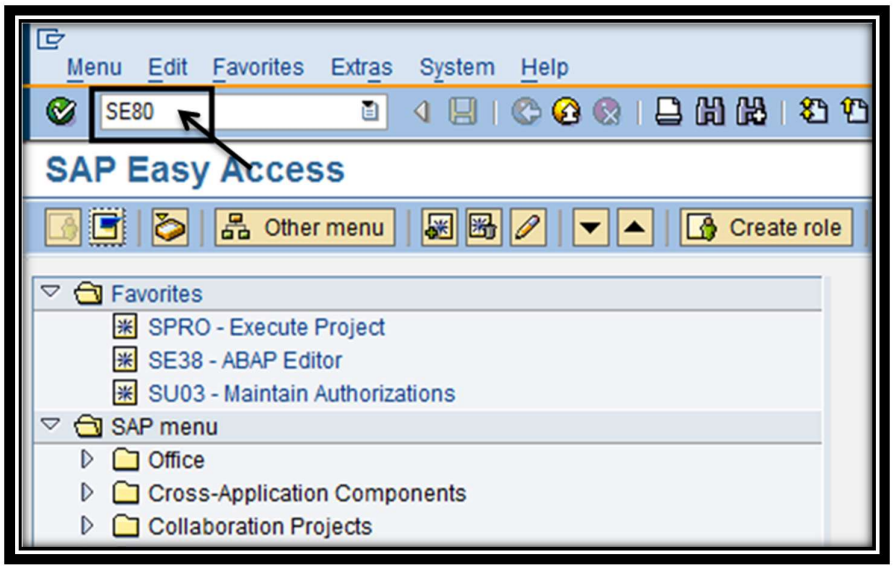
4. Once PAI's end module is encountered then the cursor position from ABAP/4 module pool program is transferred back to the screen flow logic, which will terminate the transaction.

Navigation Steps to create a Simple Module pool program or Transaction or Dialog programming

Scenario:

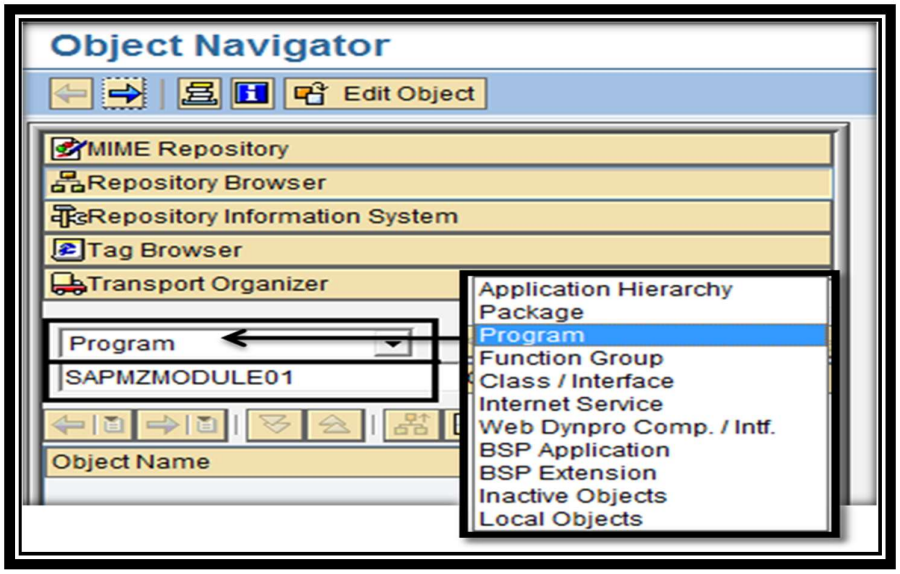


The above scenario depicts that if the user enters the material number and material type as an input and if he clicks on the Display push button then the details regarding that particular material such as material created date, created person, material group and unit of measure need to be display in the respective fields of the material. And if the user clicks on Exit push button then the user have to be come out of the screen SE80 of the program.

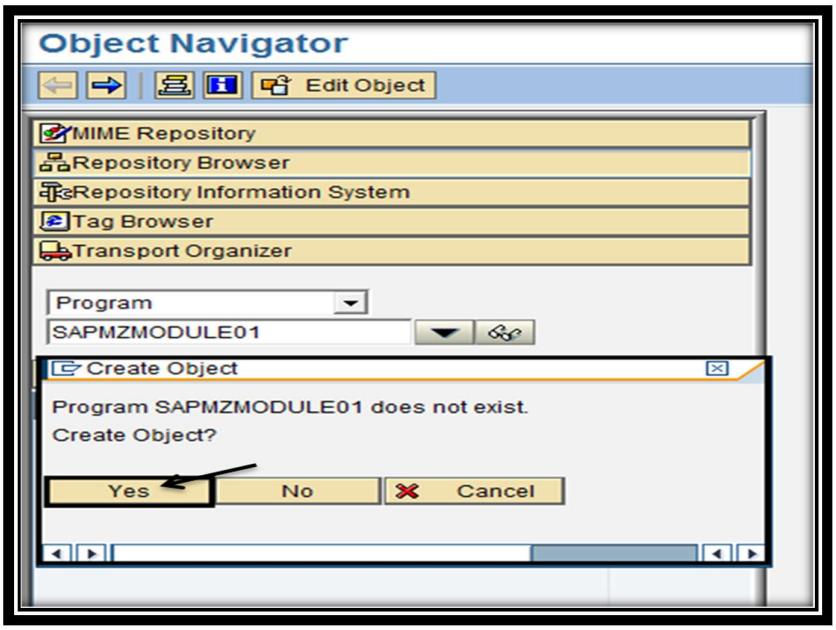


Transaction code **SE80** is an **Object Navigator** tool. Generally transaction or dialog program is created with the help of ABAP Work bench tool known as Object Browser or Object Navigator.

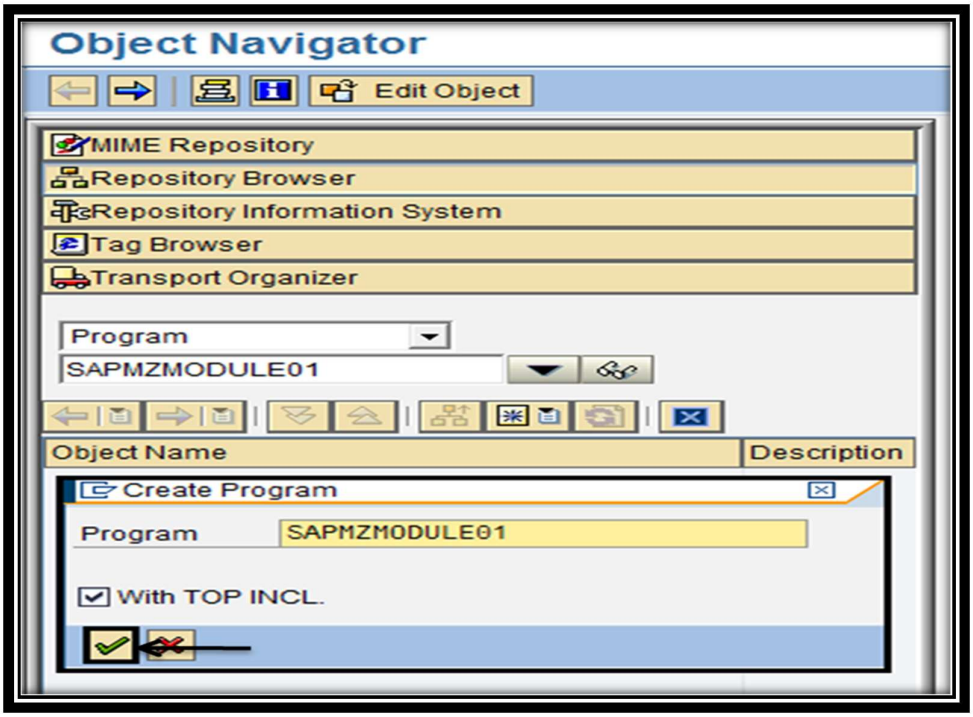
Enter **SE80** transaction code in the command field then press **ENTER**.



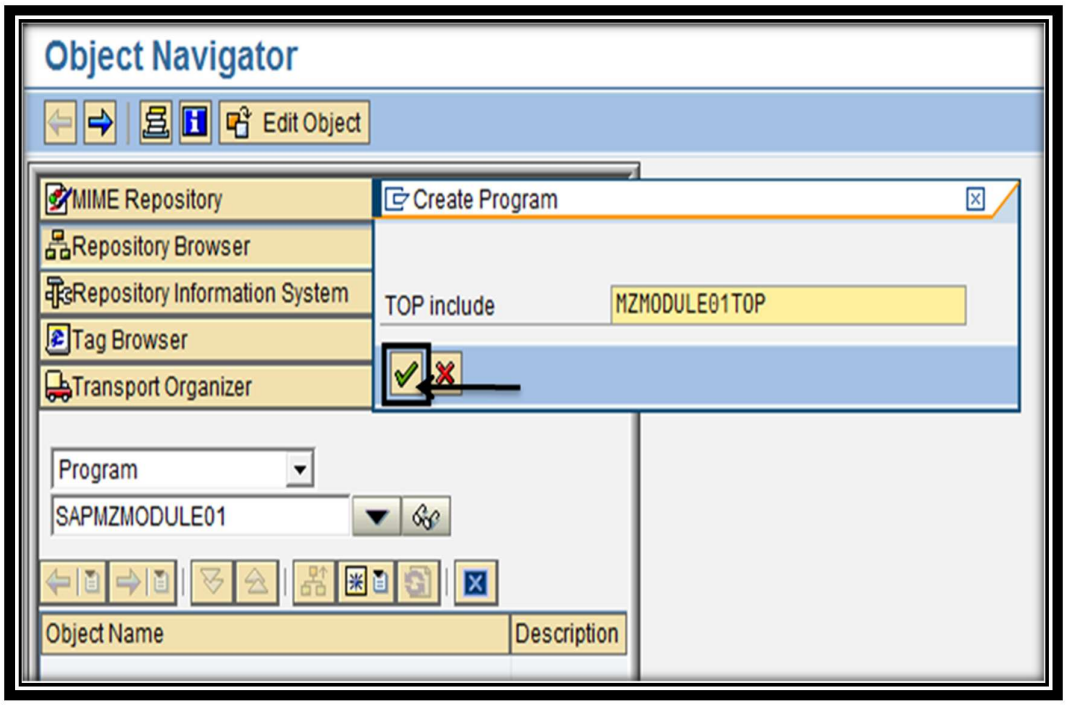
In dropdown list select program & provide module pool program name starts with SAPMY or SAPMZ as SAPMZMODULE01 then press ENTER.



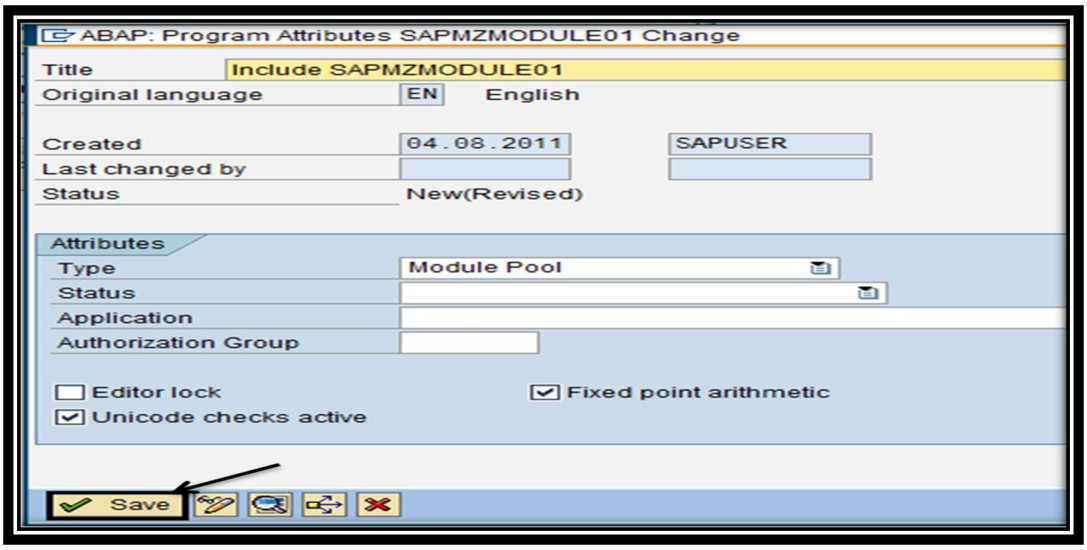
Click on yes for creating object for the program.

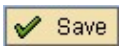


It will ask you to create program with TOP INCLUDE, and then press enter.

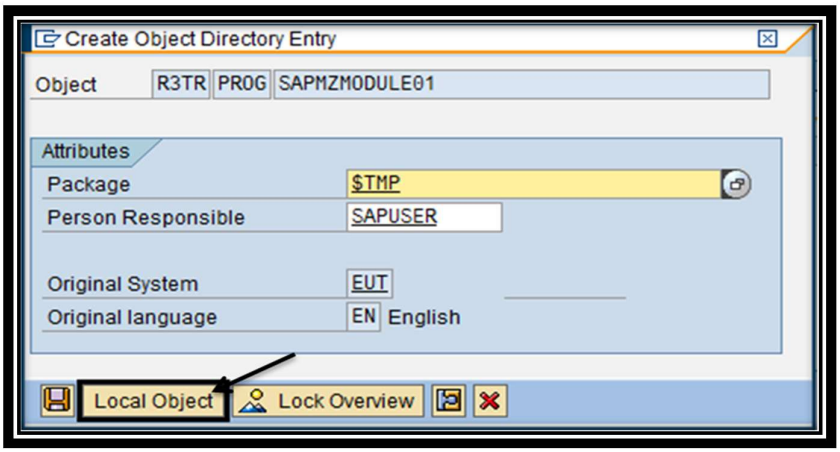


Then it will propose the include name as MZMODULE01TOP then press enter.

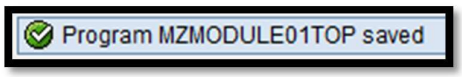


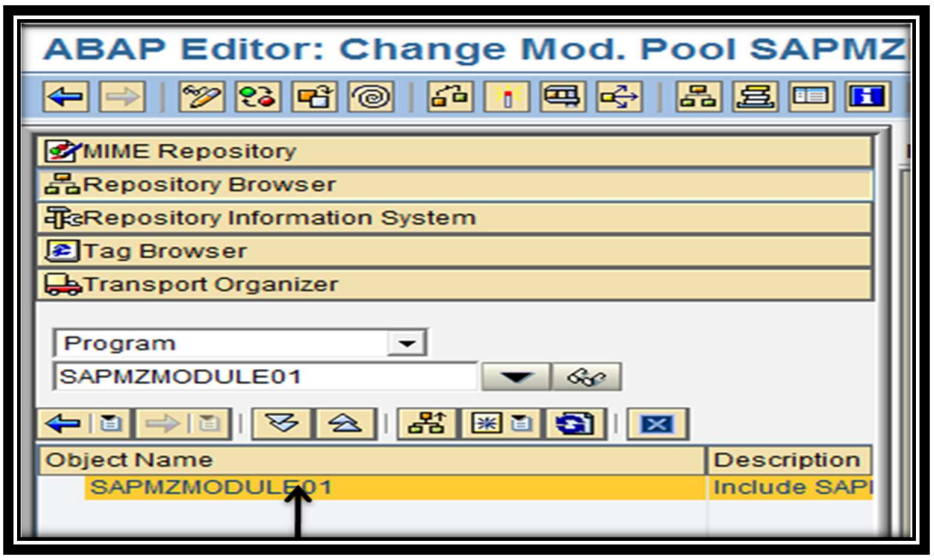
Opens attributes information. Then click on 

button.

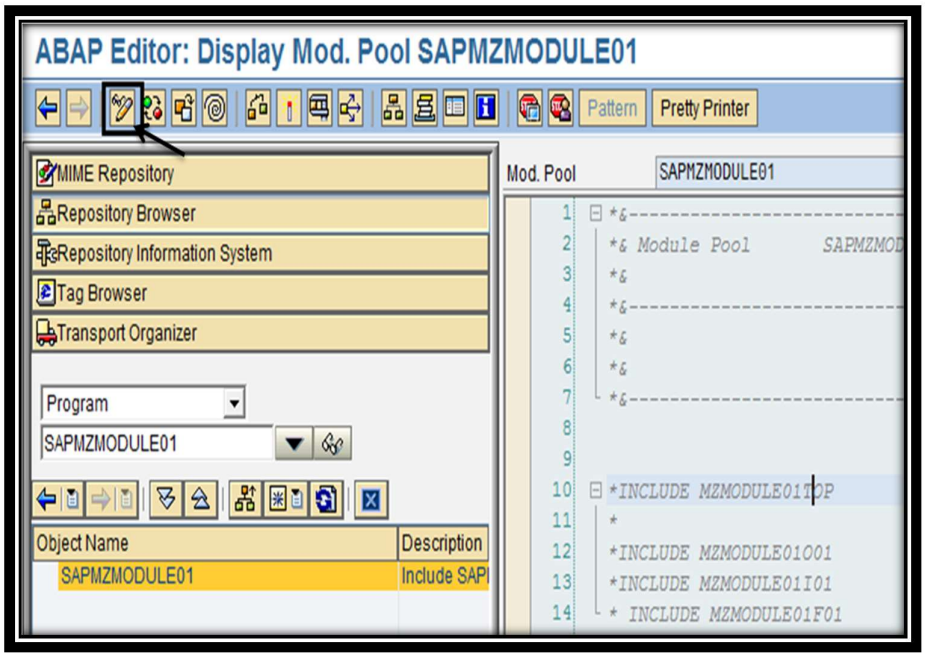


Save it as **LOCAL OBJECT ($TMP)**. Then the following success message will appear on the status bar.

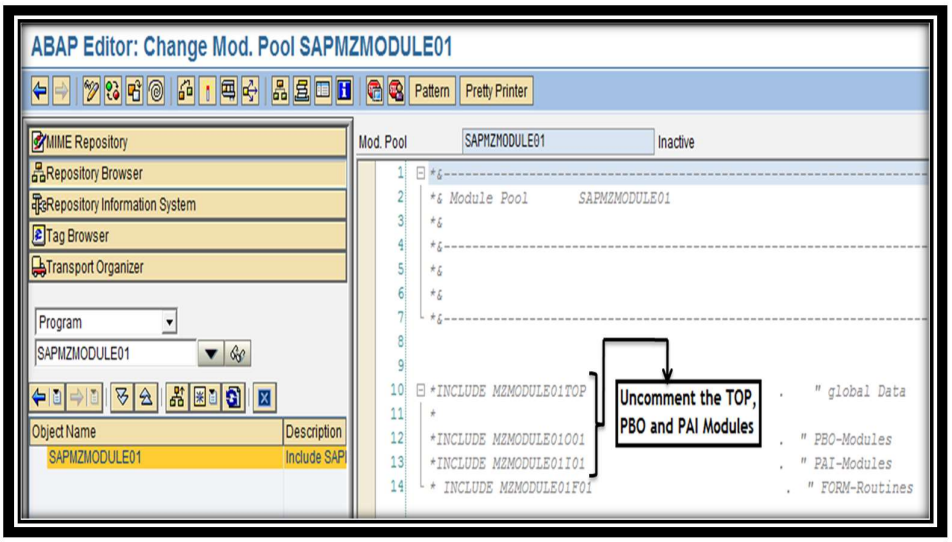




Double click on the object **SAPMZMODULE01** from the left panel.

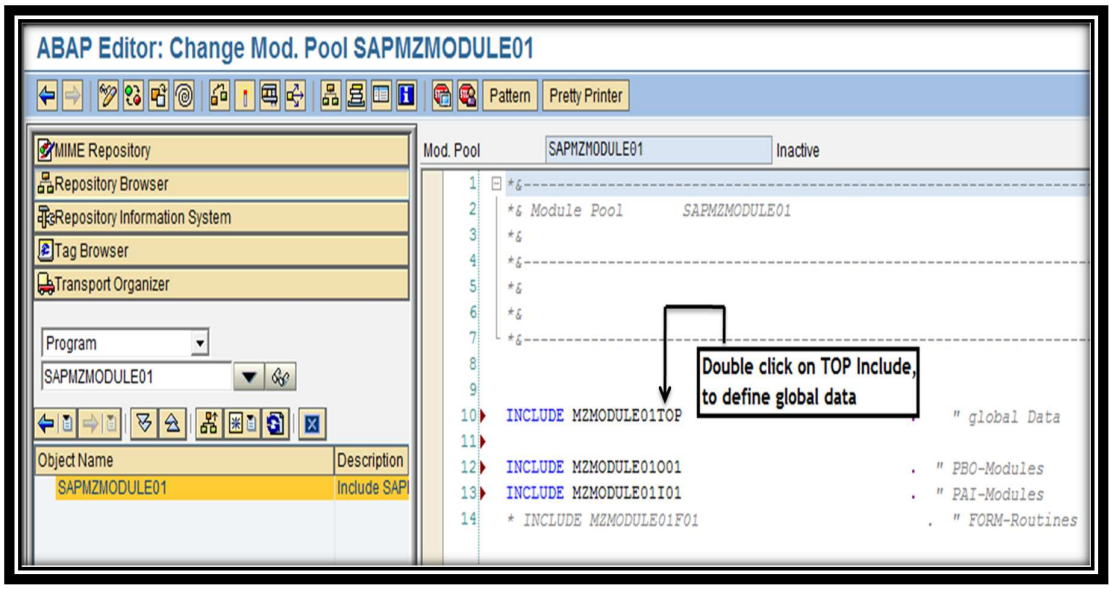


Click on change mode from application tool bar or (Ctrl+F1) to make the workbench screen enable.

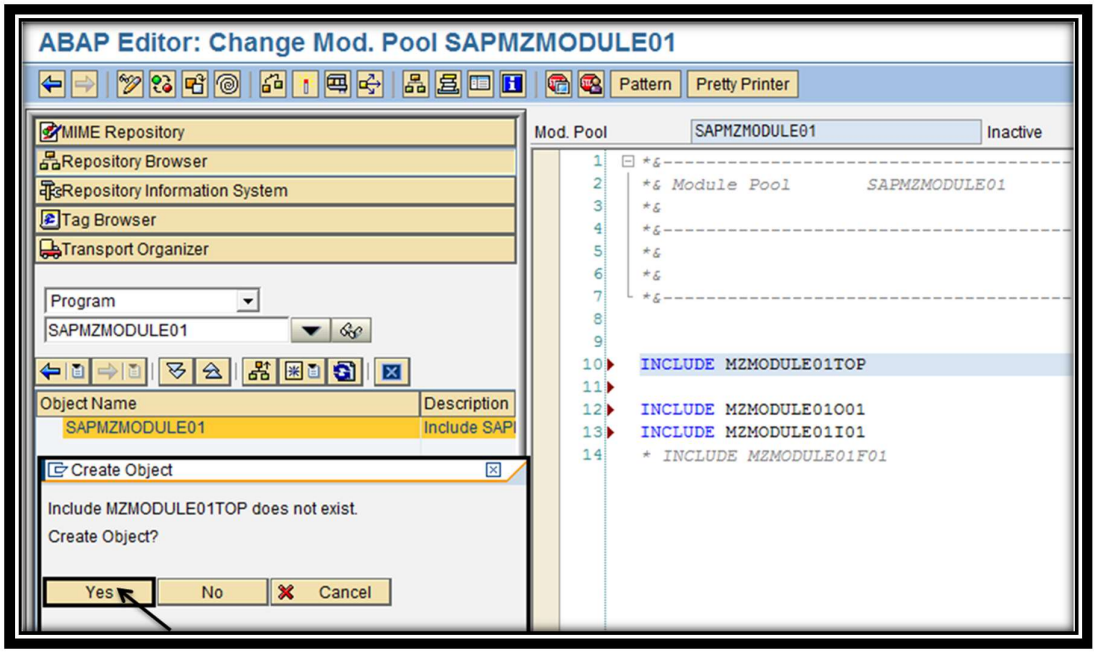


Uncomment the below Includes:

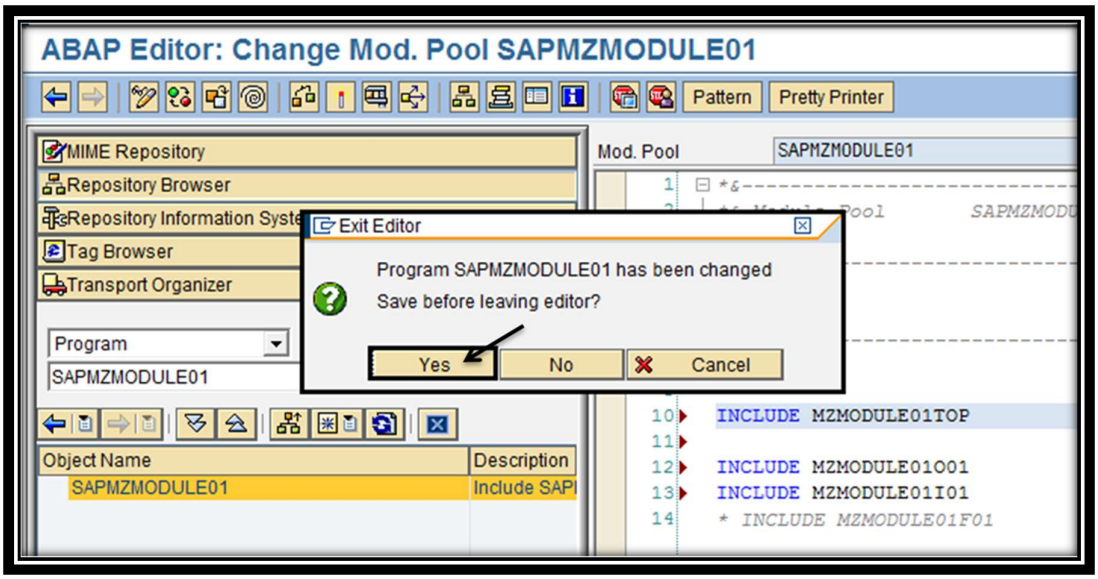
- ✓ **INCLUDE MZMODULE01O01**
- ✓ **INCLUDE MZMODULE01I01**



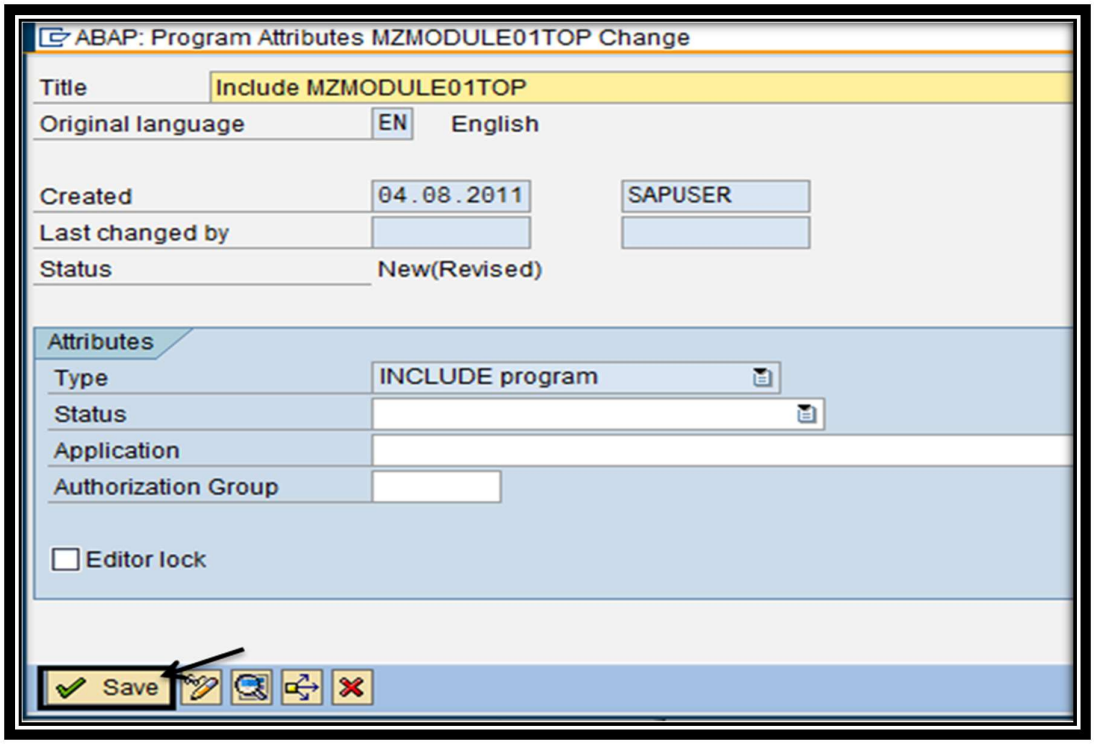
Double click on INCLUDE MZMODULE01TOP, which is the global data of the transaction program.

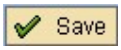


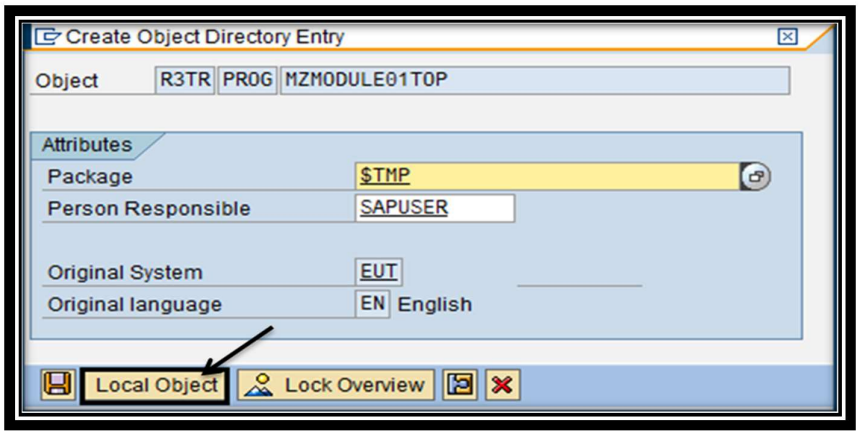
Click on **Yes** to create the object for the TOP Include.



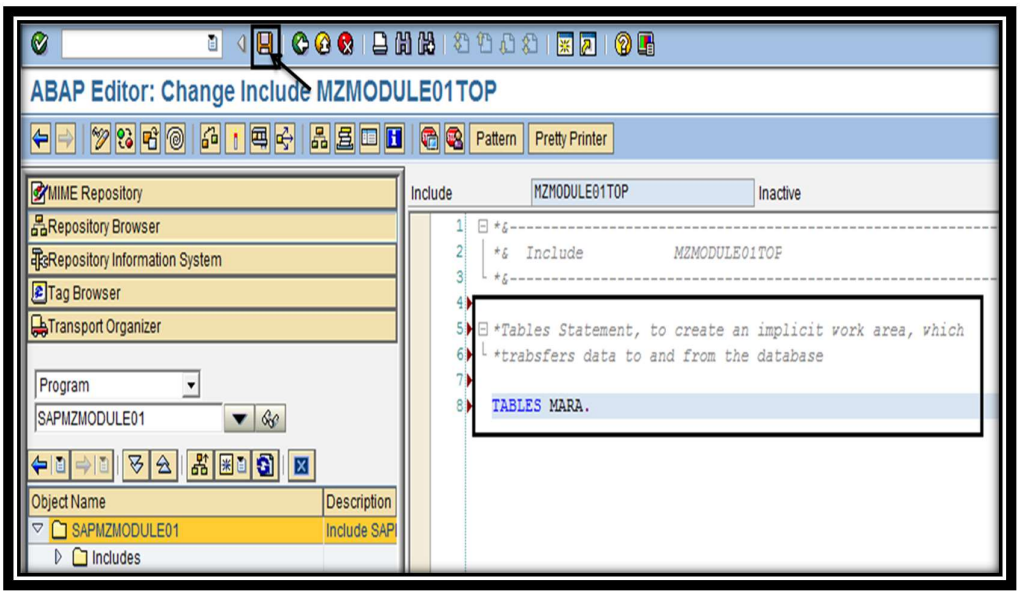
Again click on **Yes** to save the program before creating the object for TOP Include.



Then click on  button.



Save it as **LOCAL OBJECT ($TMP) or in a Package**.



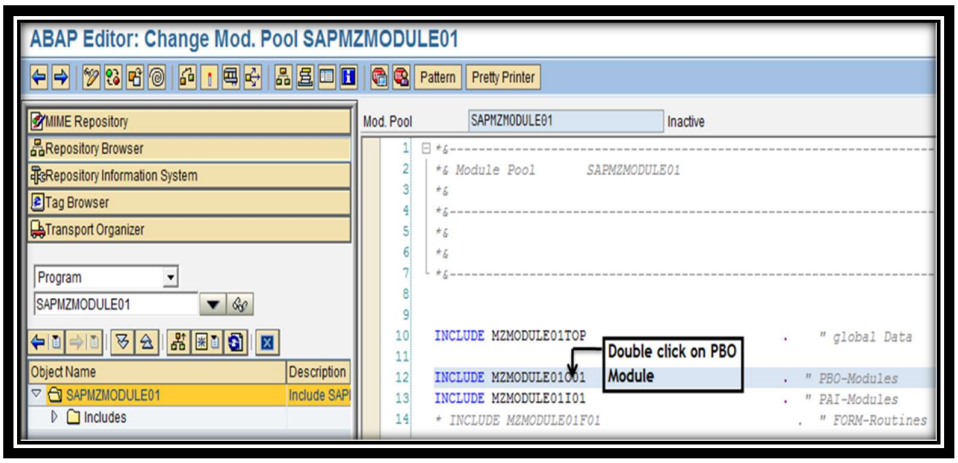
Double click on Top Include.

Provide the following code on the above Editor, which is used to write the program logic.

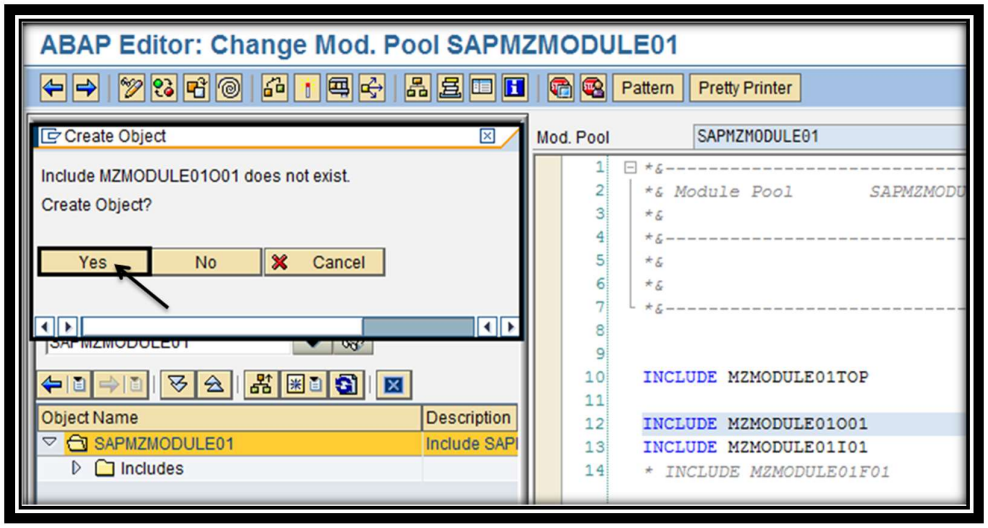
```
*Tables Statement, to create an implicit work area, which
*transfers data to and from the database

TABLES MARA.
```

Then click on **SAVE (CTRL+S)** button from standard tool bar and save it in a **Local Object**. It will display the following success message on the status bar.

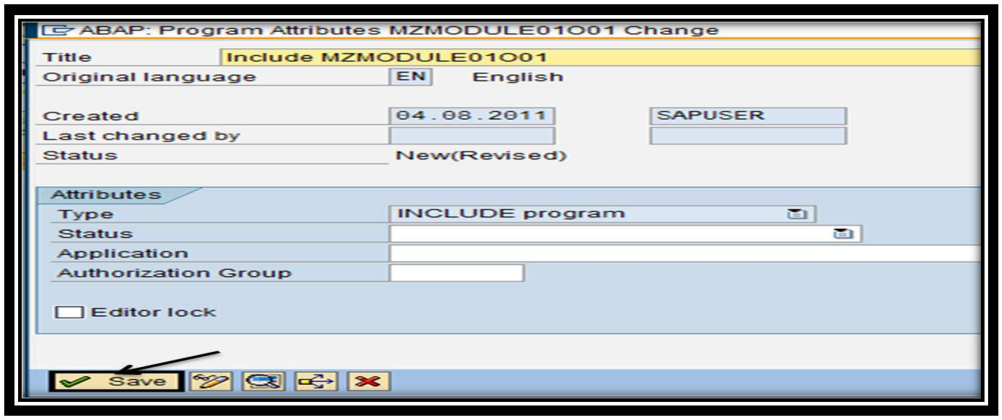
Click on back button from standard tool bar then the following screen will come.

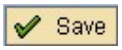


Double click on **INCLUDE MZMODULE01O01,** which is PBO Module of the transaction program.

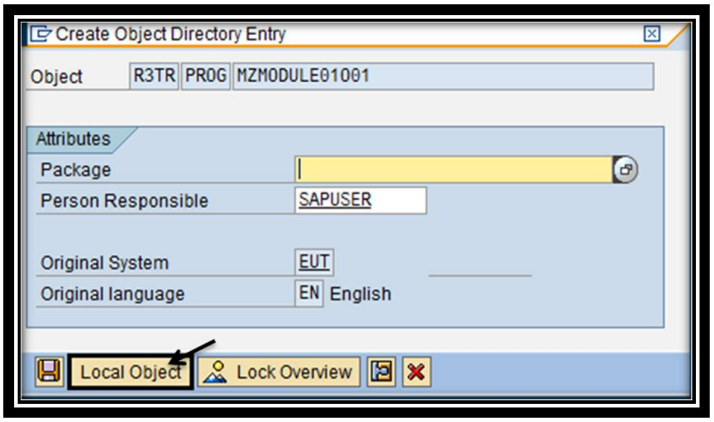


Click on **Yes** to create the object for the O01 Include, which is the PBO module.

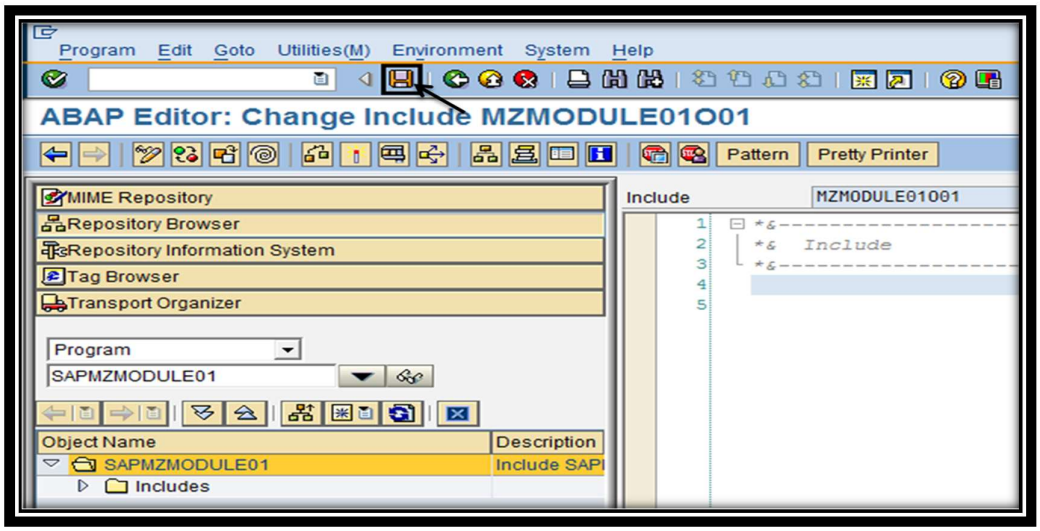


Then click on 

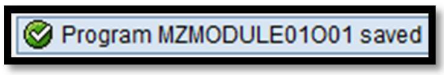
button.



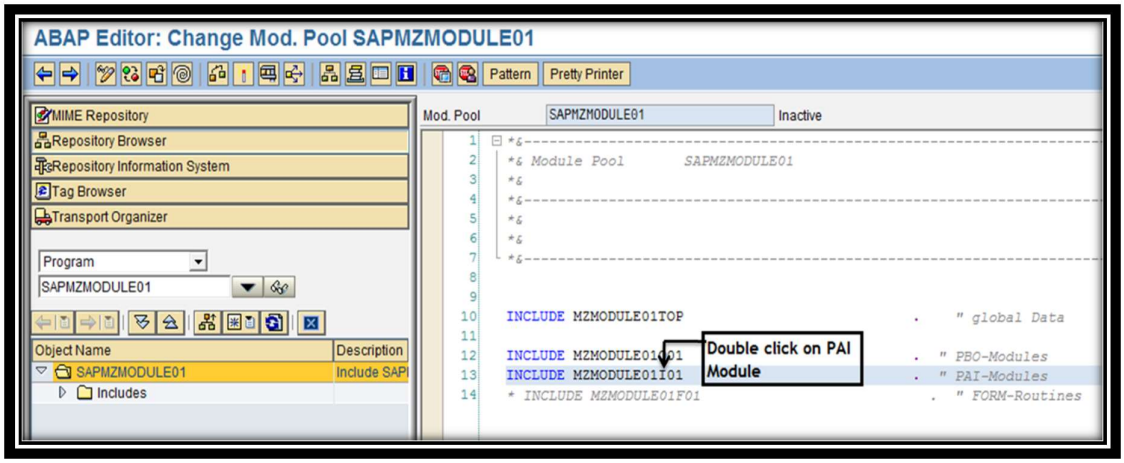
Save it as **LOCAL OBJECT ($TMP) or in a Package.**



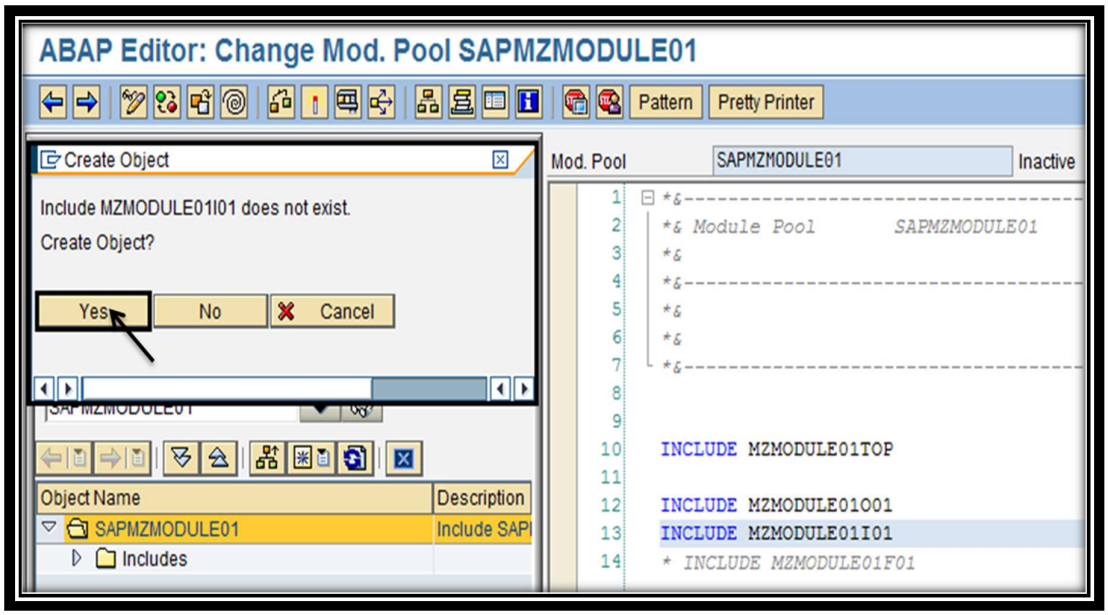
Then click on **SAVE (CTRL+S)** button from standard tool bar and save it in a **Local Object**. It will display the following success message on the status bar.



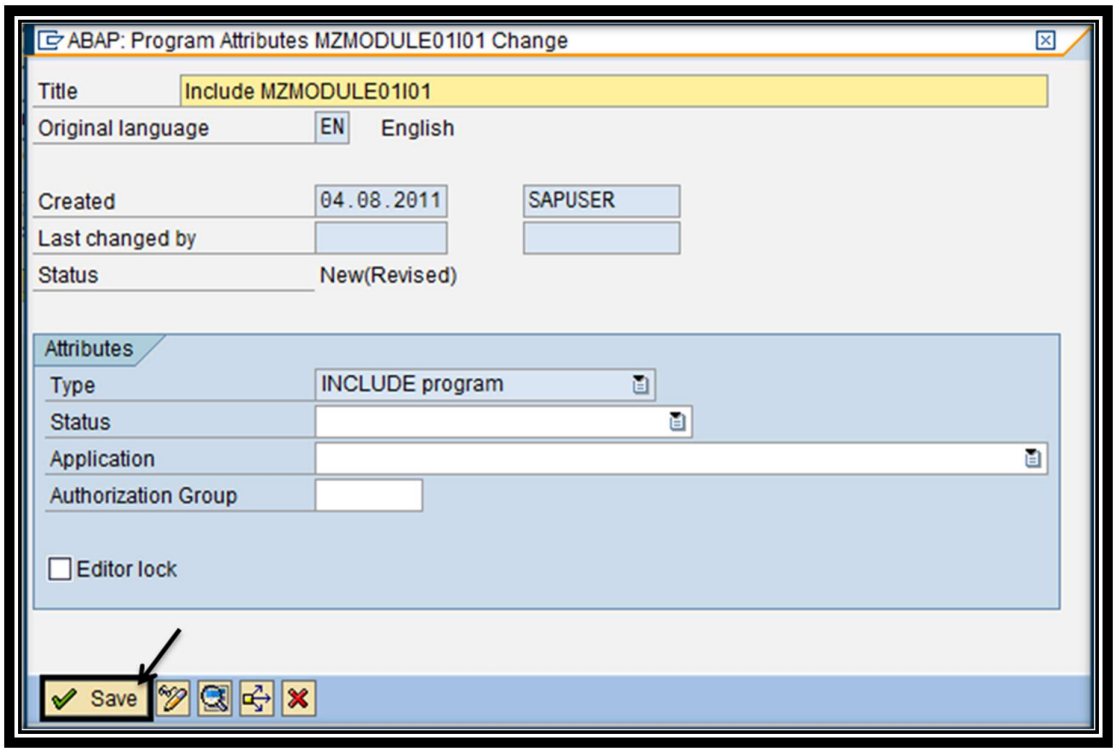
Click on back button from standard tool bar then the following screen will come.

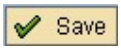


Double click on **INCLUDE MZMODULE01I01,** which is PAI Module of the transaction program.

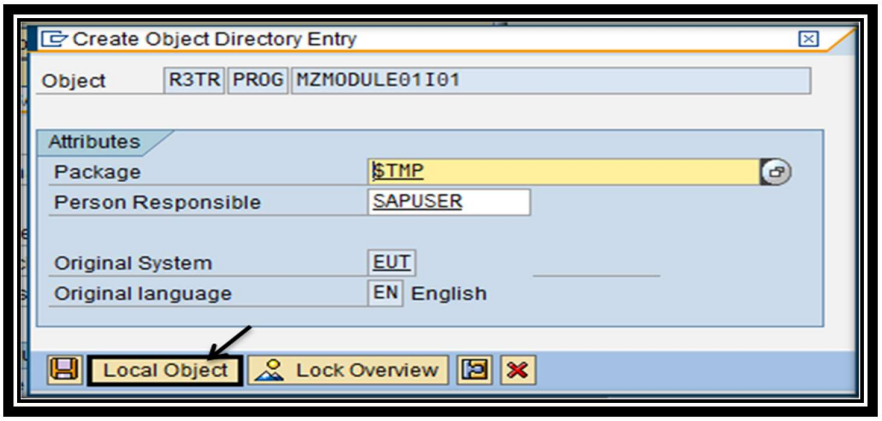


Click on **Yes** to create the object for the I01 Include, which is the PAI module.

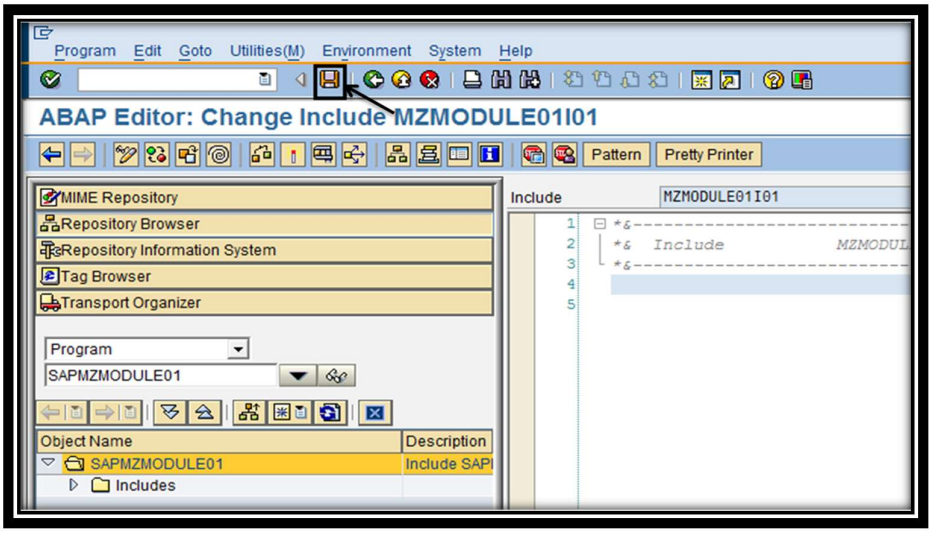


Then click on 

button.

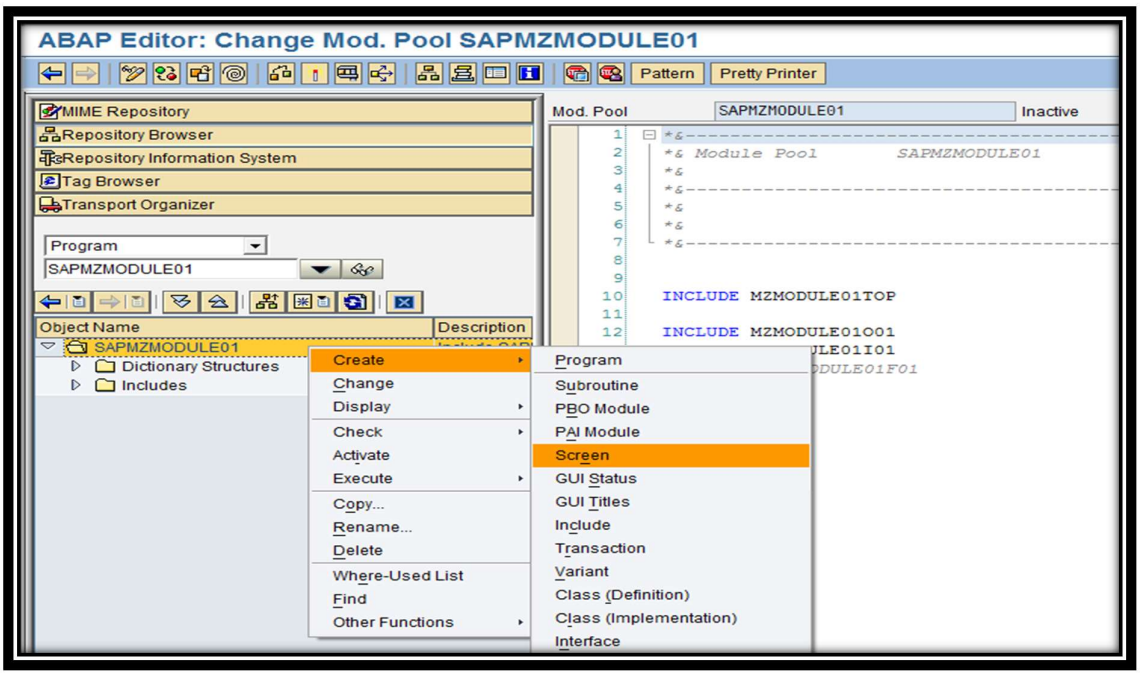


Save it as **LOCAL OBJECT ($TMP) or in a Package.**

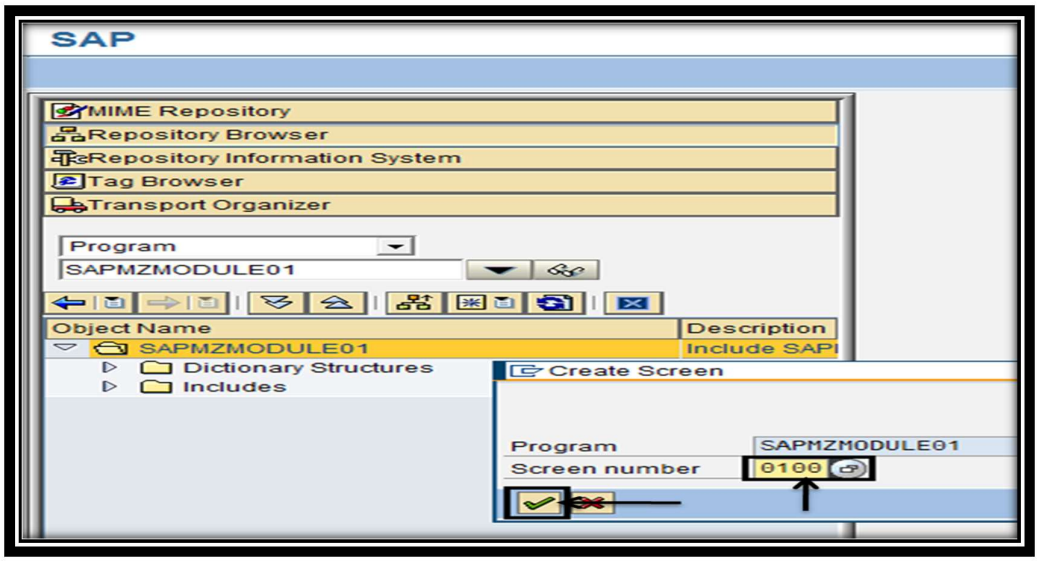


Then click on **SAVE (CTRL+S)** button from standard tool bar and save it in a **Local Object**. It will display the following success message on the status bar.

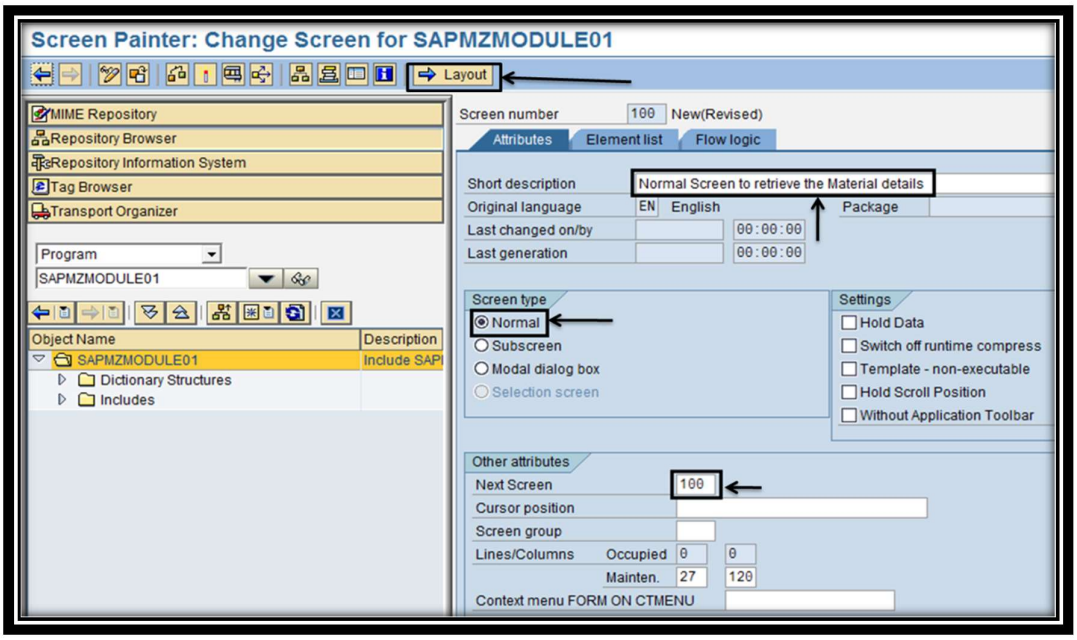
Click on back button from standard tool bar then the following screen will come.



Select the object from the left panel, right click then select **Create → Screen.**



Provide the screen number **0100** then press **ENTER.**

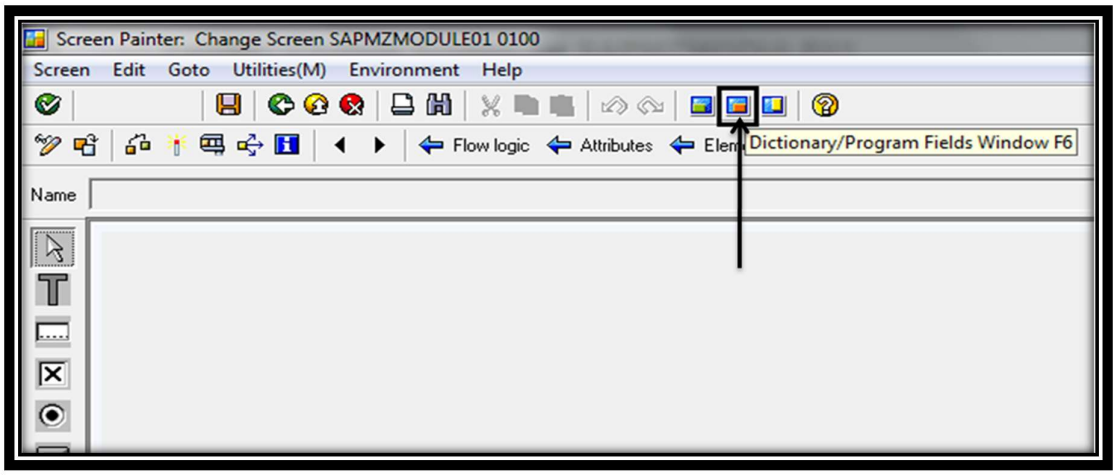


Maintain the following field values on the above screen fields.

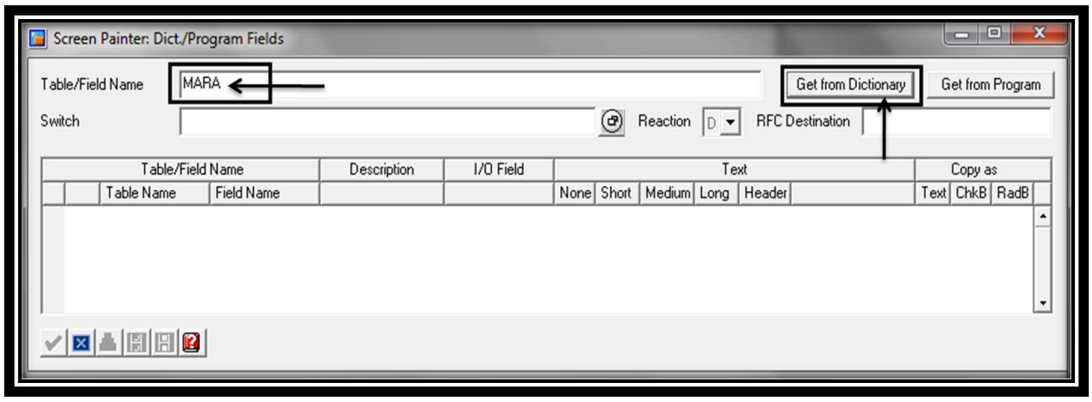
Short description	**-**	**Normal Screen to retrieve the Material details**
Screen type	**-**	**Normal**
Next Screen	**-**	**100**

Click on

button from application tool bar to design the screen.



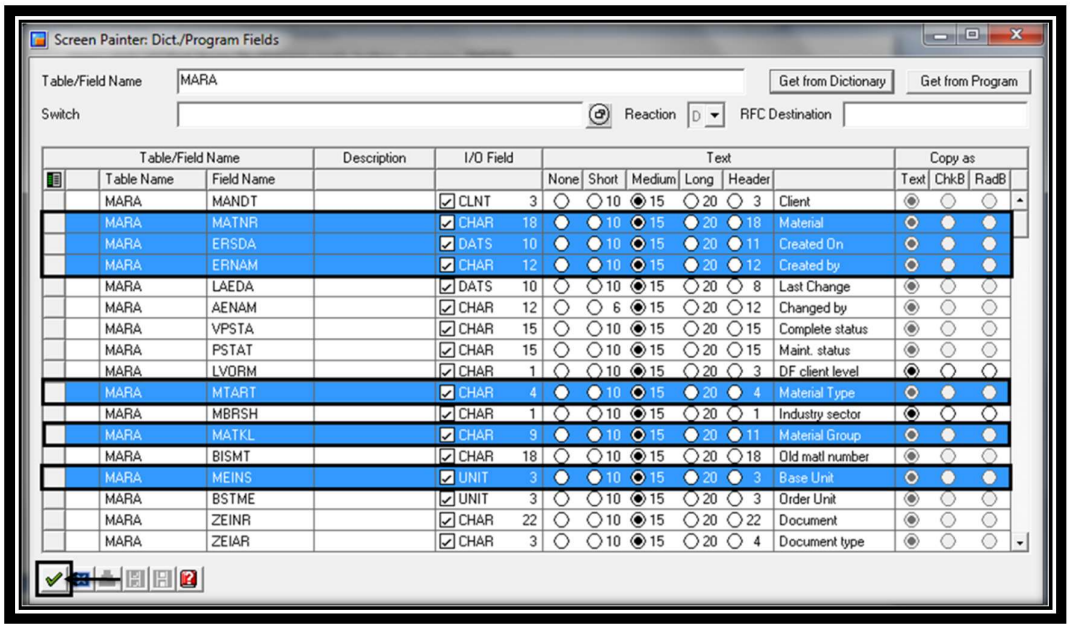
Click on Dictionary Fields Icon from standard tool bar (or) press F6.



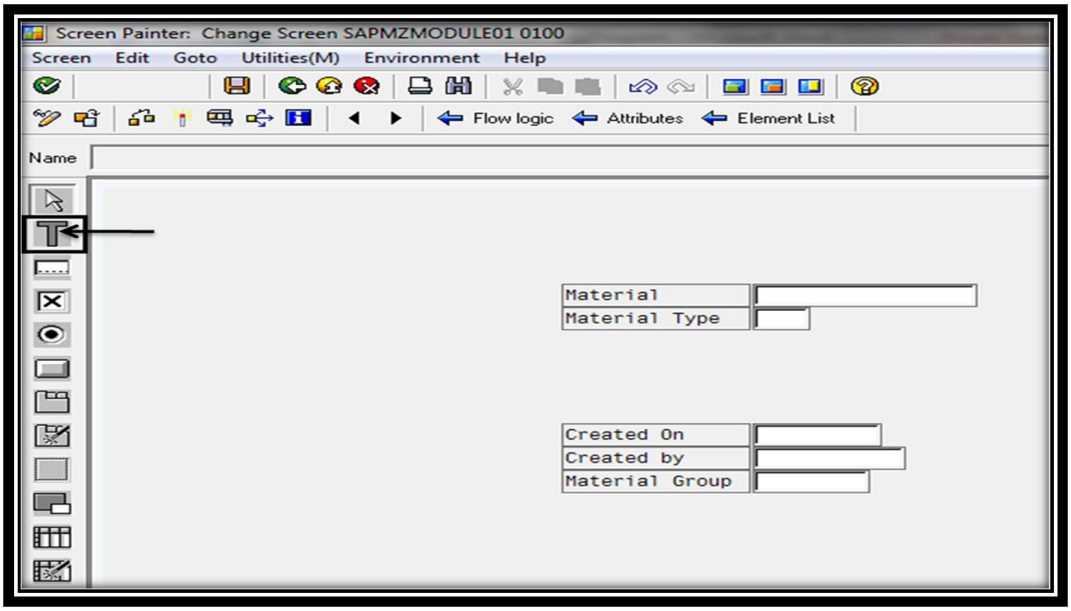
Maintain the following field values on the above screen fields.

Table/Field Name - MARA

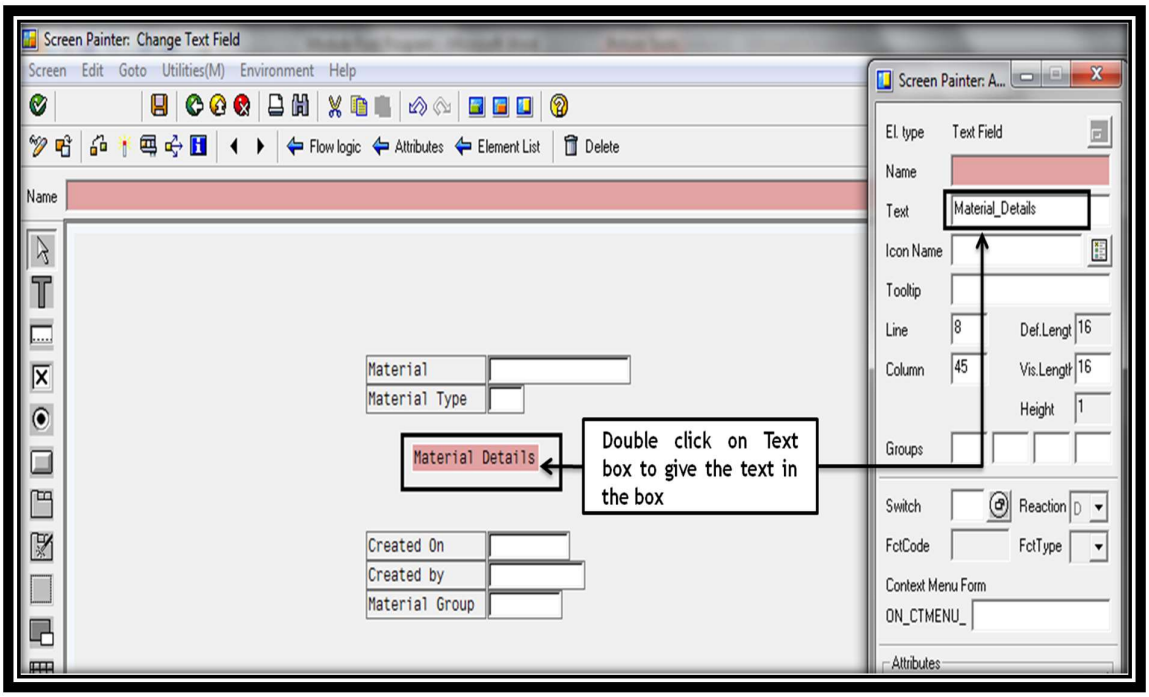
Then click on **Get from Dictionary** push button or press **ENTER**.



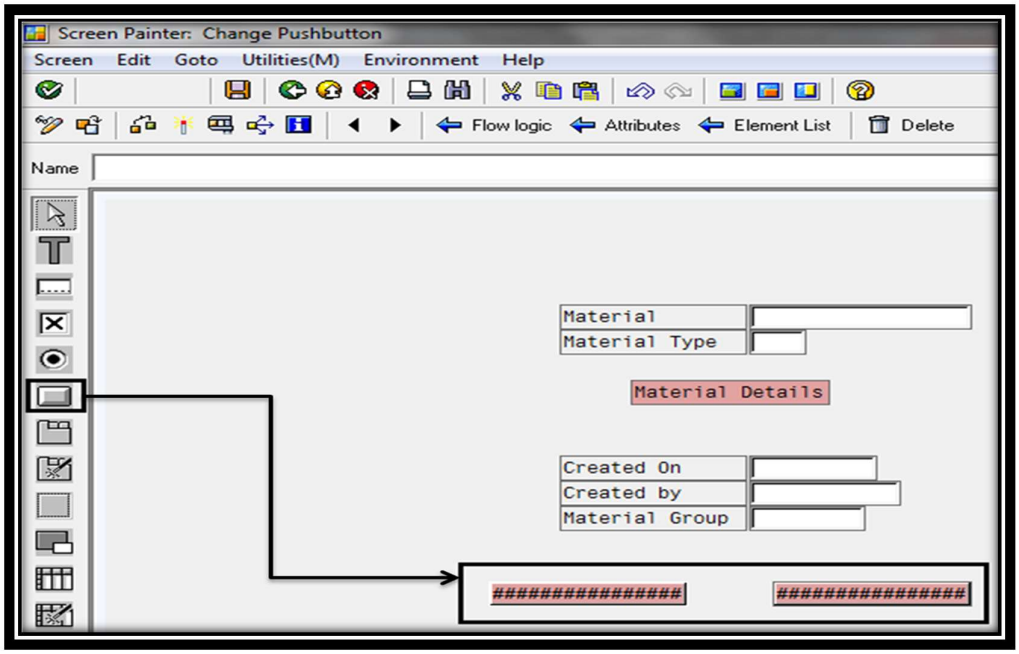
Select the fields **MATNR, ERSDA, ERNAM, MTART, MATKL** and **MEINS** and drag & drop onto the layout.



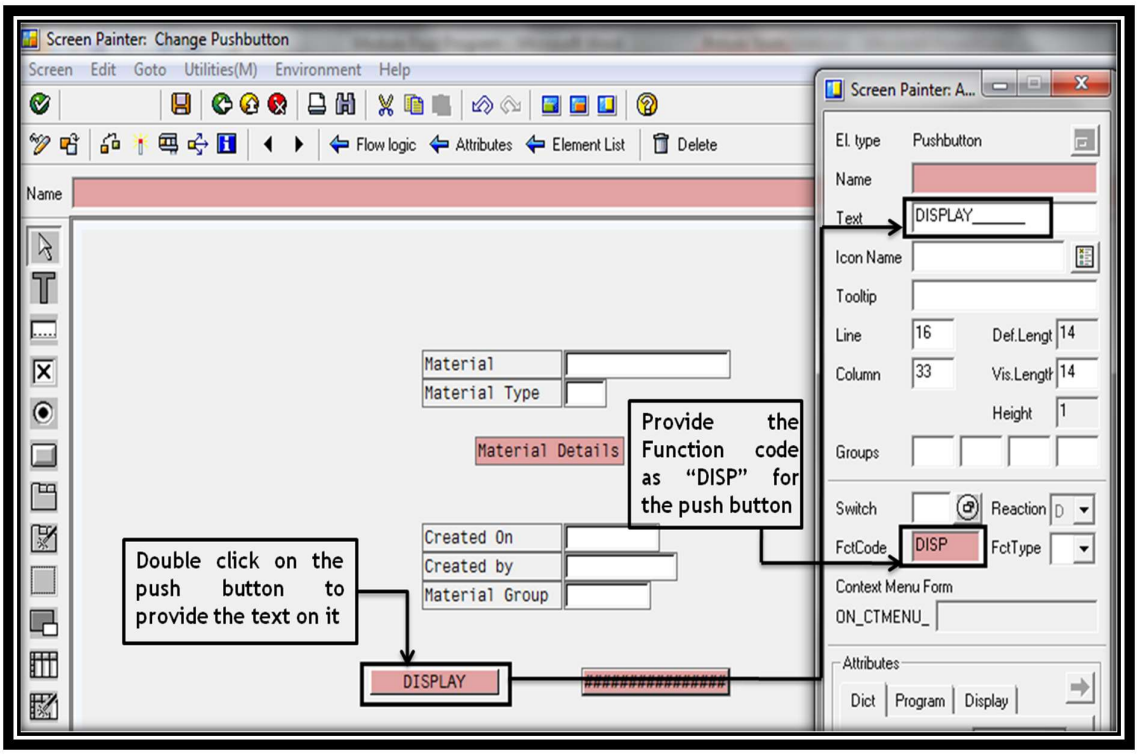
Click on Text field (T) from left panel & drag & drop on to the layout.



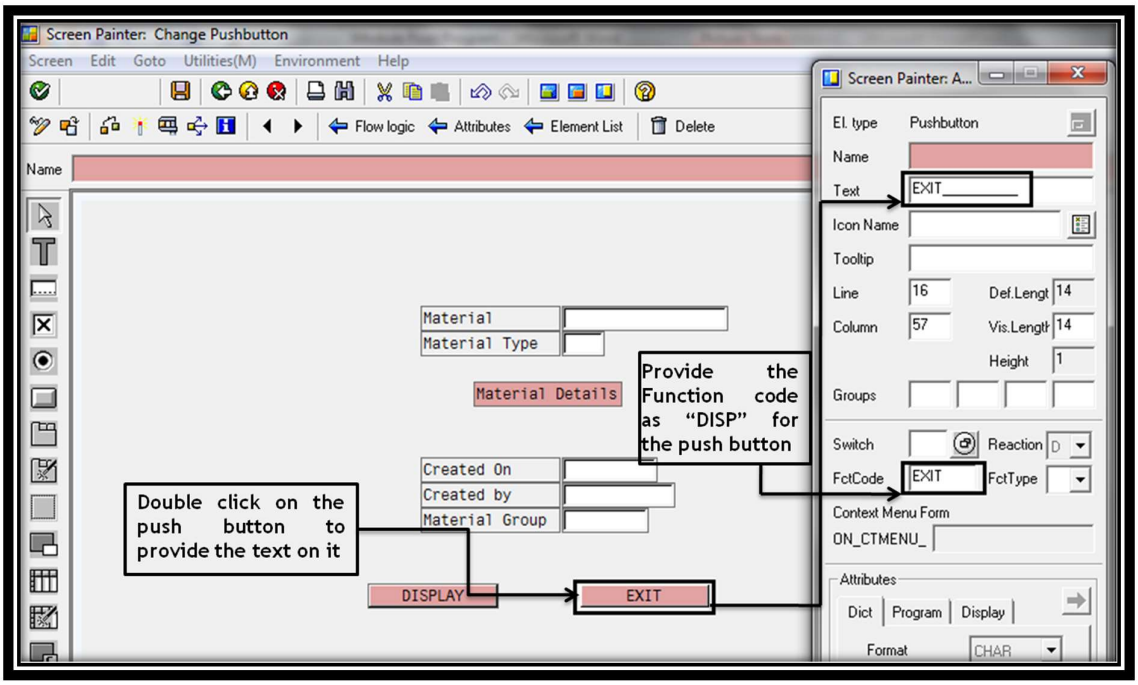
Double click on text field provide text as **Material Details.**



Select the push button from the left panel then drag & drop on to the layout.

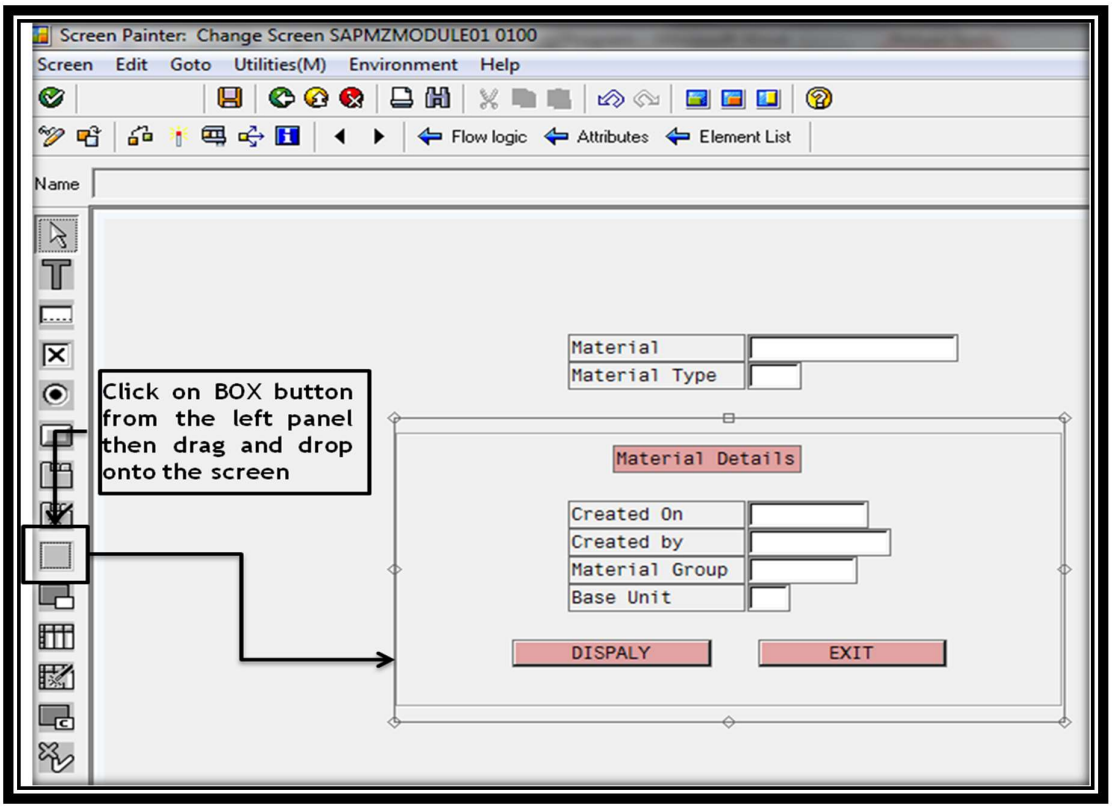


Then double click on push button then provide text as **DISPLAY**, provide function code as **DISP**.

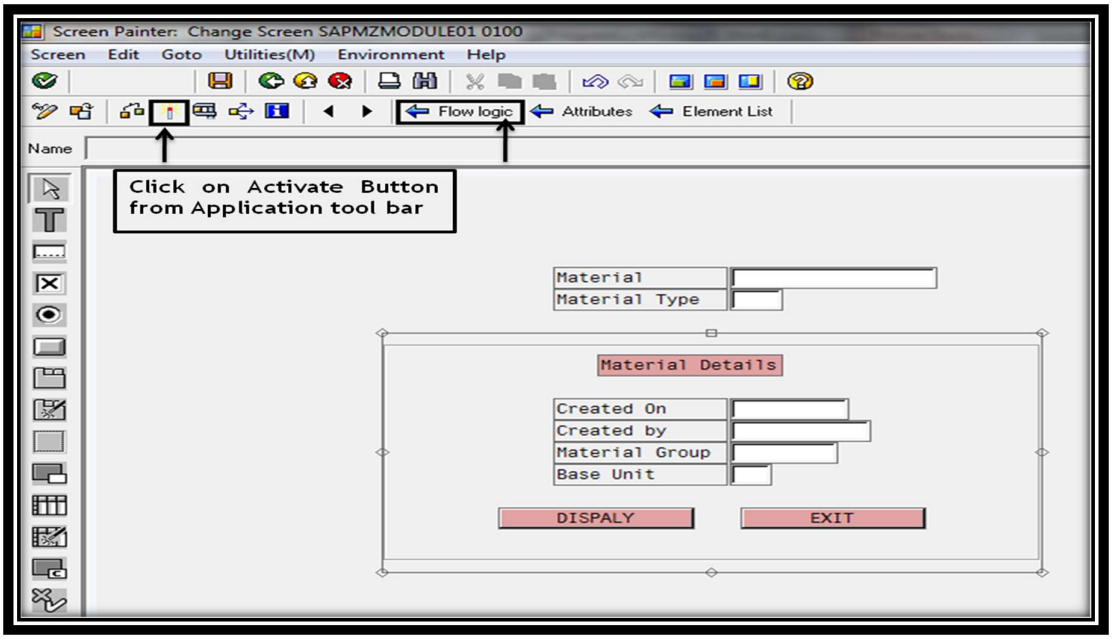


Select another push button from the left panel then drag & drop on to the layout.

Double click on push button then provide text as **EXIT**, provide icon provide function code as **EXIT**.

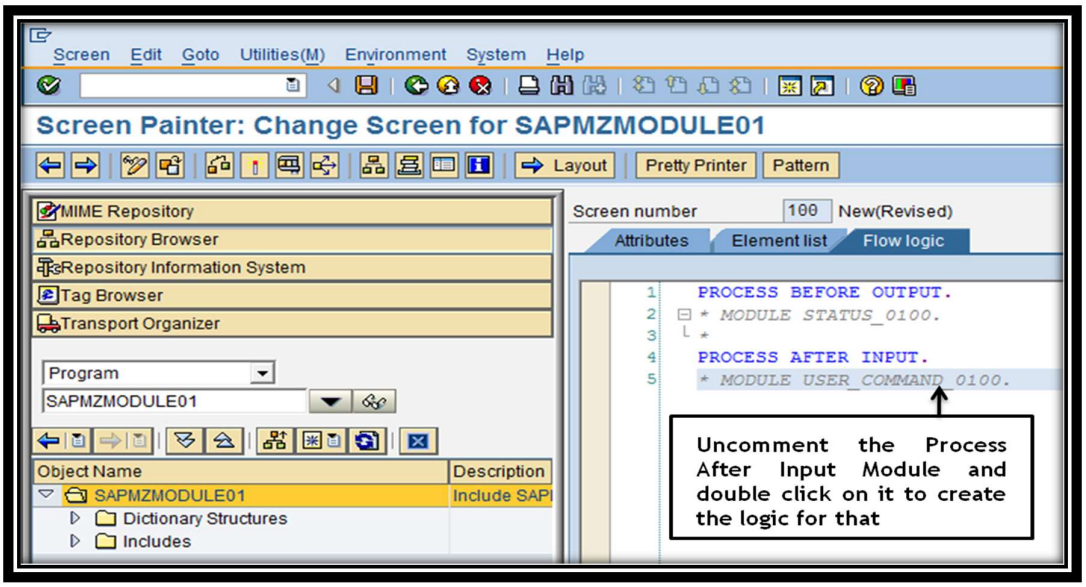


Click on **Logical Box** icon from the left panel onto the layout.

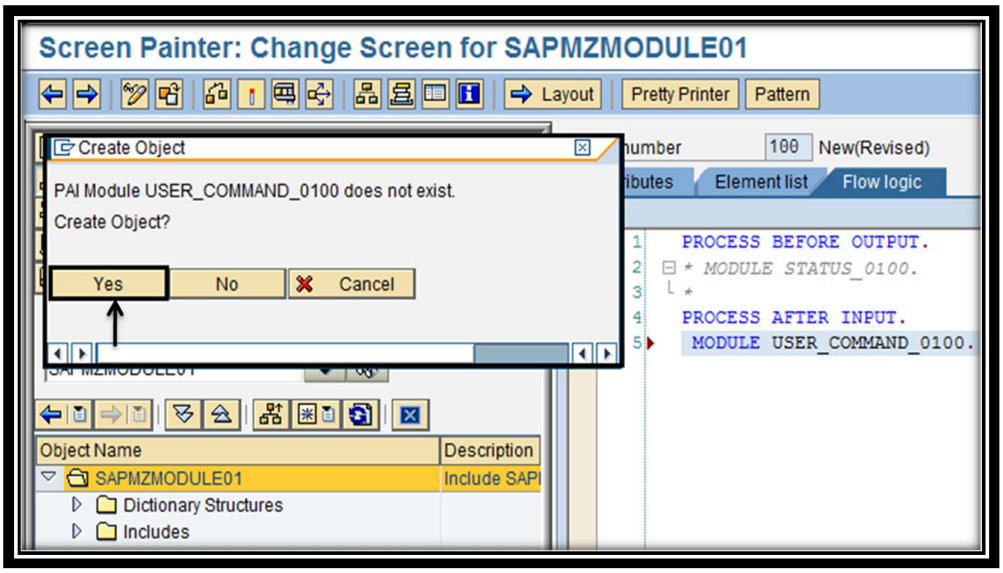


Then click on **Activate (CTRL+F3)** button from application tool bar to activate the screen.

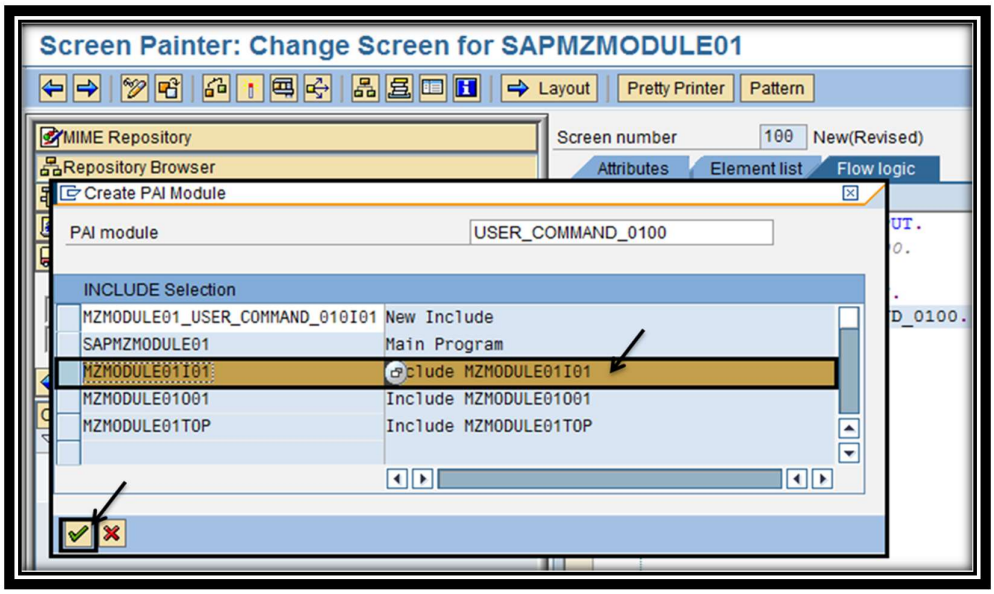
Click on **Flow Logic** button from application tool bar, to provide program logic for the dialog program



Uncomment the **MODULE USER_COMMAND_0100** for Process After Input.



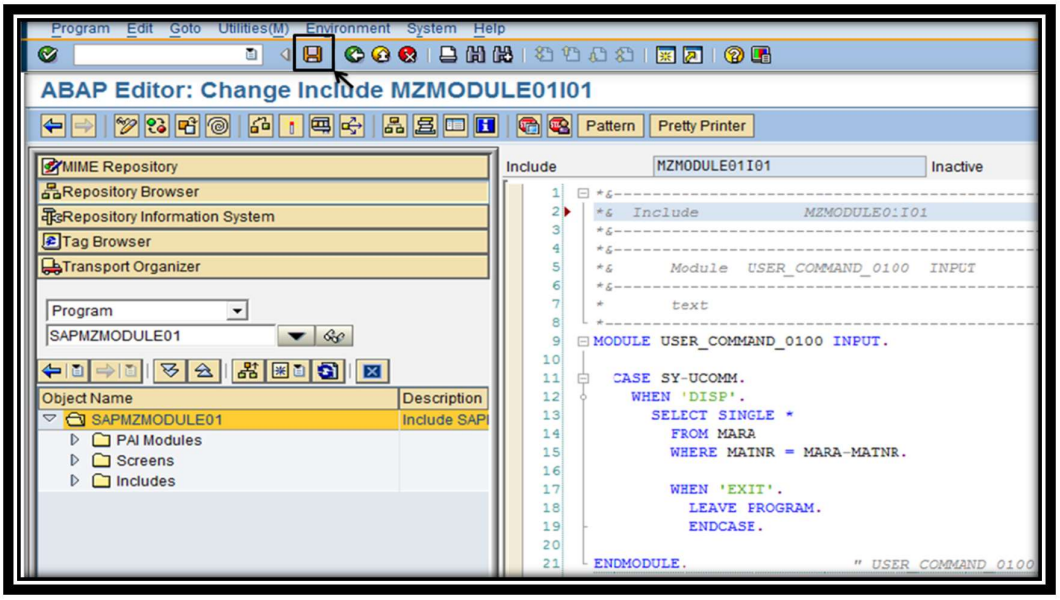
Double click on **USER_COMMAND_0100**, and then click on **Yes** to creating object for the PAI module.



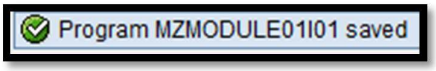
Select MZMODULE01I01 because this is for PAI module Enter & Save it.

Enter the following code on the Editor of program logic.

```
MODULE USER_COMMAND_0100 INPUT.
  CASE SY-UCOMM.
    WHEN 'DISP'.
      SELECT SINGLE *
        FROM MARA
        WHERE MATNR = MARA-MATNR AND MTART = MARA-MTART.
    WHEN 'EXIT'.
      LEAVE PROGRAM.
      ENDCASE.
ENDMODULE.              " USER_COMMAND_0100  INPUT
```



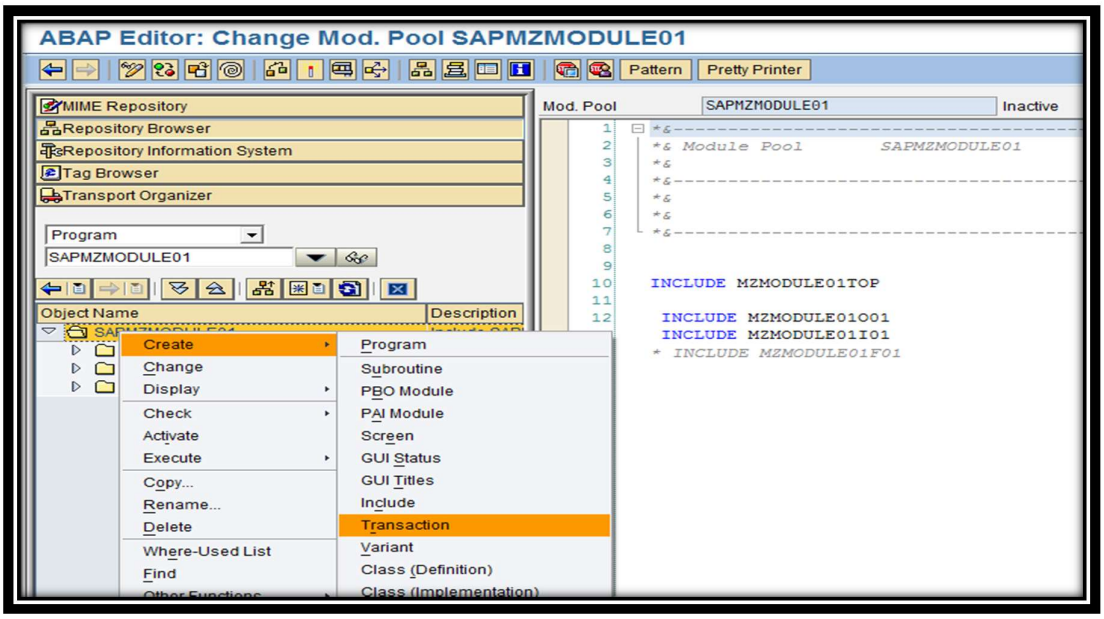
Then click on **SAVE (CTRL+S)** button from standard tool bar and save it in a **Local Object**. It will display the following success message on the status bar.



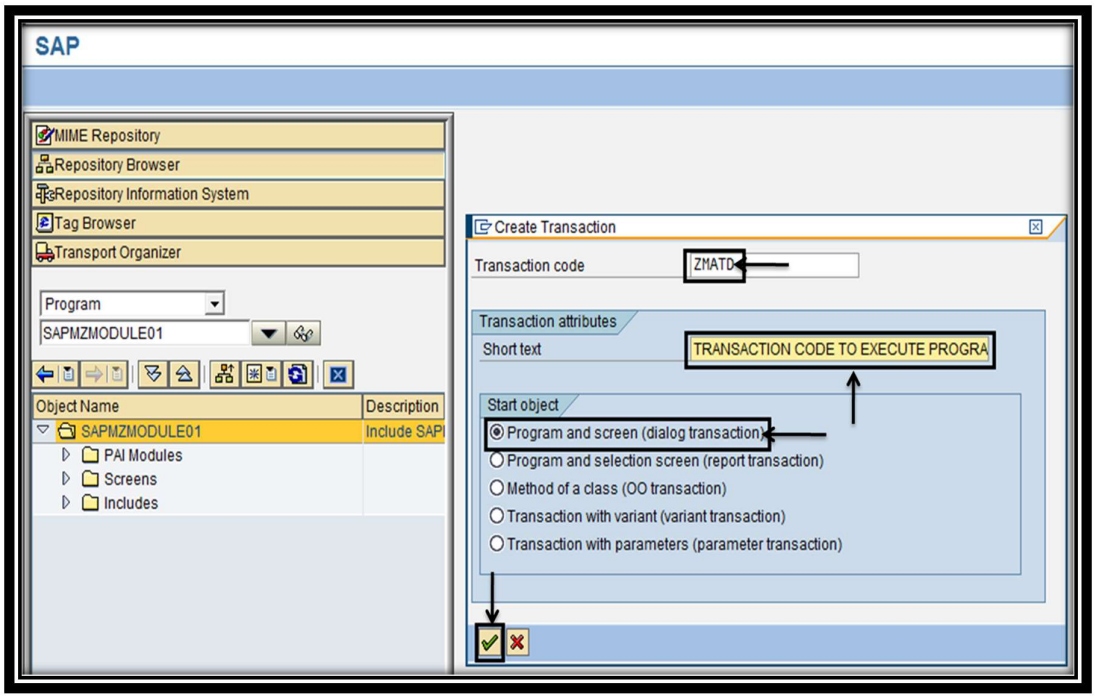
Then click on **Check (CTRL+F2)** button from application tool bar then the following message will come on the status bar.

Then click on **Activate (CTRL+F3)** button from application tool bar then press on **ENTER** to activate the dialog program.

Click on back button from standard tool bar then the following screen will come.



Select the object from the left panel right click, and then select **Create → Transaction**.

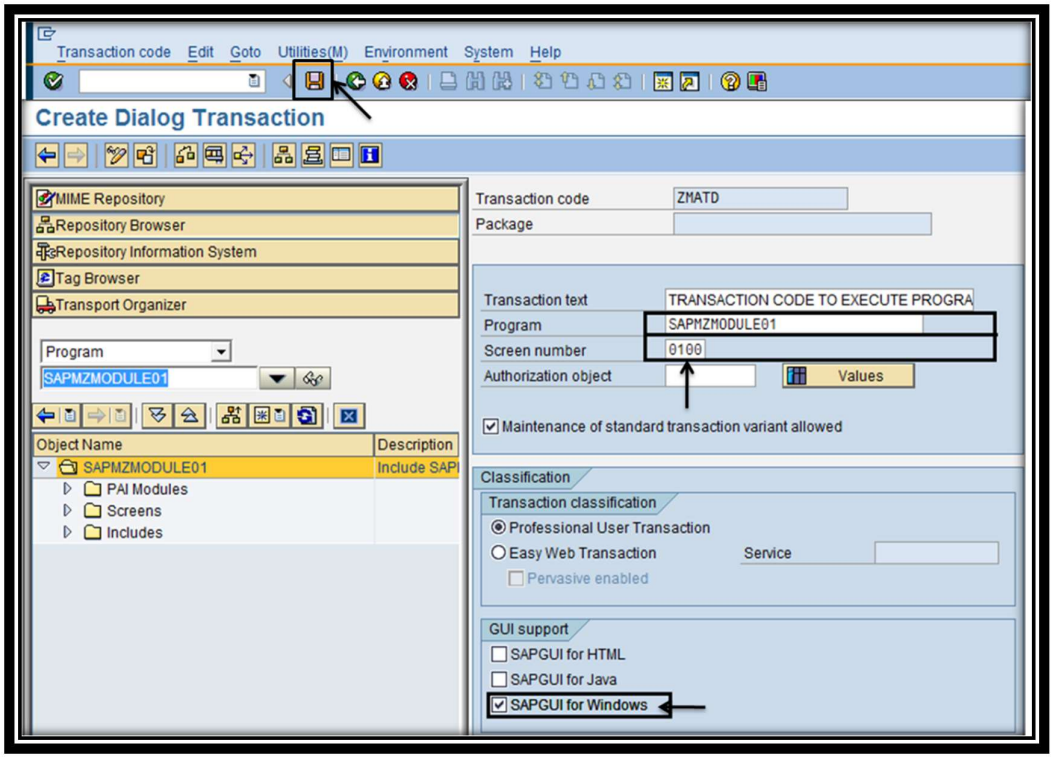


Maintain the following field values on the above screen fields.

Transaction code - **ZMATD**

Short text - **Transaction Code to execute program**

Default ◉ Program & Screen radio button is selected then Press **ENTER.**



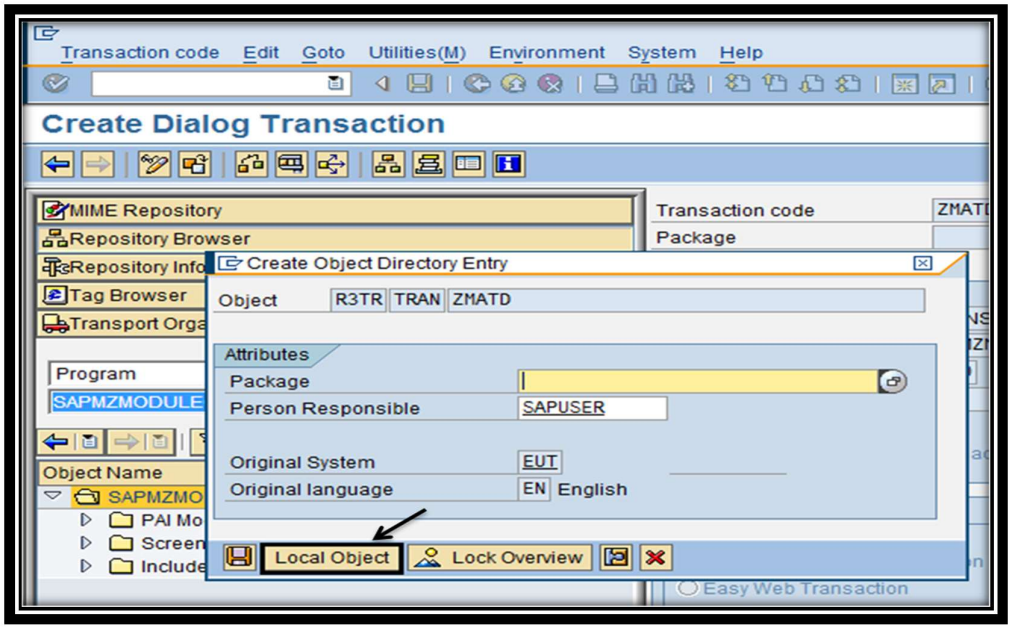
Maintain the following field values on the above screen fields.

Program - SAPMZMODULE01

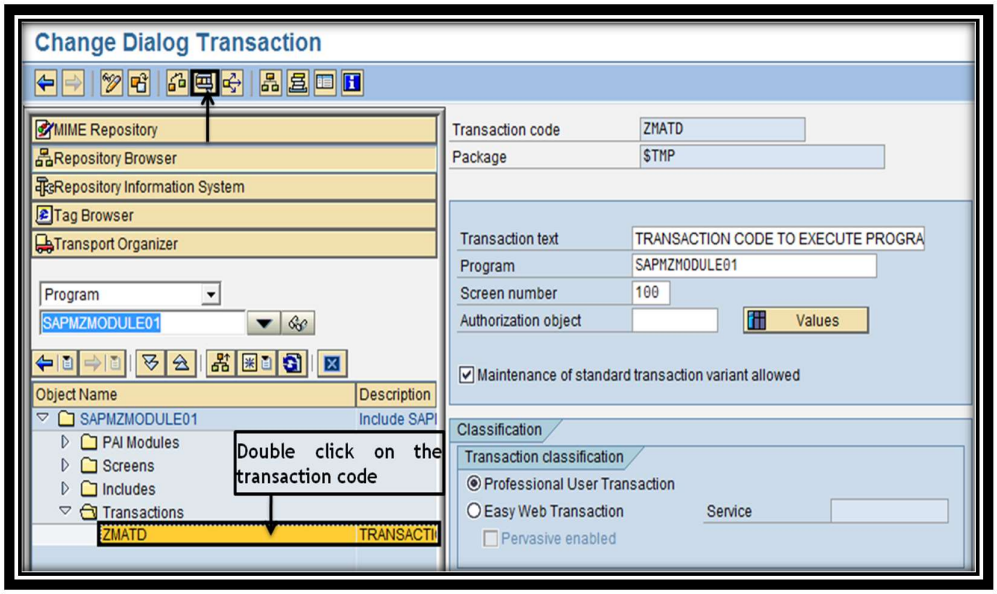
Screen number- 100

GUI Support as

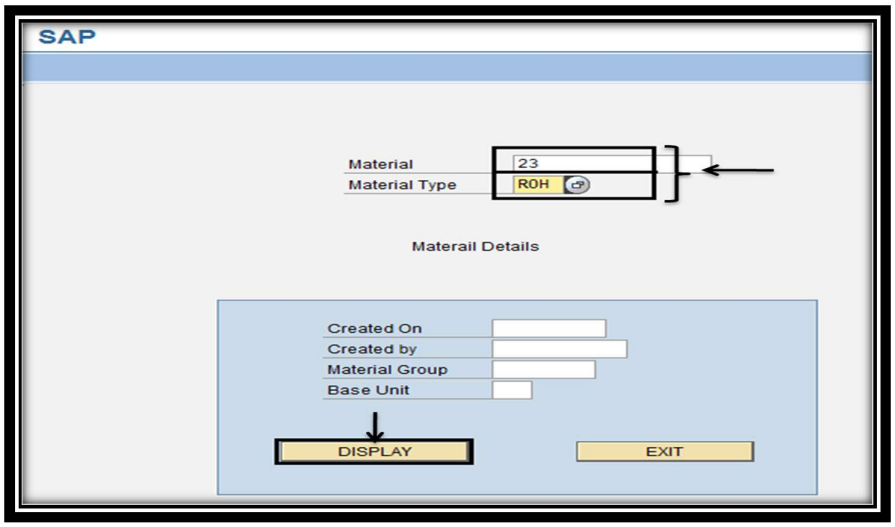
☑ SAPGUI for Windows then Save it.



Save it in the local object then the following message will appear in the status bar.



Select transaction **ZMATD** and click on **Execute (F8)** button from application tool bar to display the data onto screen fields.

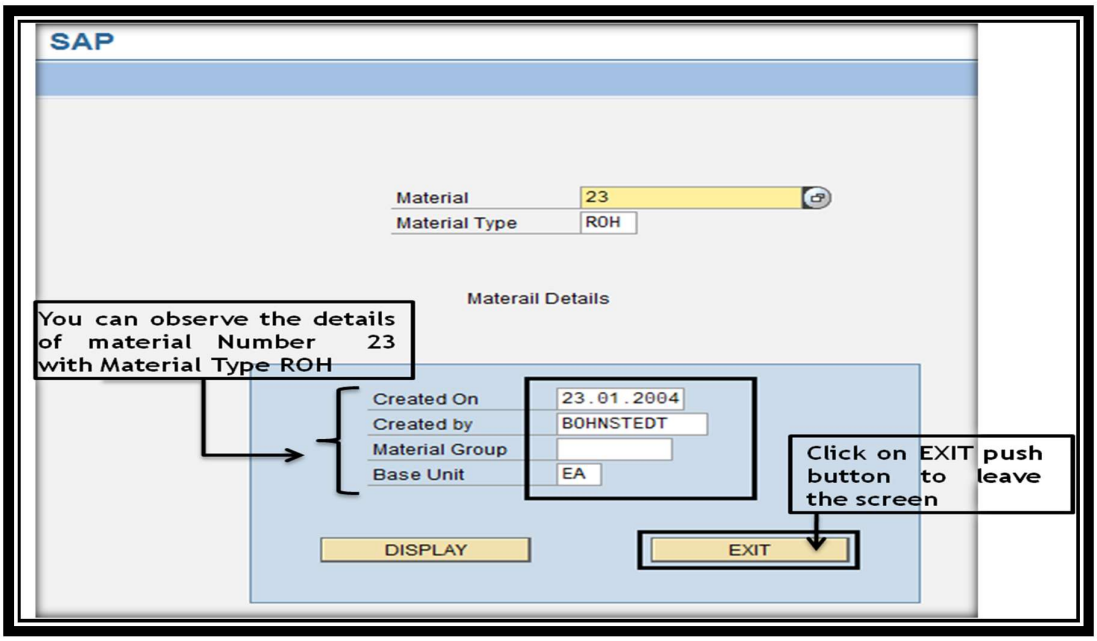


Maintain the following field values on the above screen fields.

Material - 23
Material Type - ROH

Then click on **DISPLAY** push button to display the details of the Material number 23 on the respective screen fields.

Note: If the program field names & screen field names are same then system will automatically transfer the data from module pool program to screen & vice-versa.



Then click on **EXIT** push button to leave the screen back to dialog programming initial screen.