

Dynamically Enable / Disable Fields in Table Maintenance Generator



Applies to:

SAP ABAP. For more information, visit the [ABAP homepage](#).

Summary

This article demonstrates on how to Enable / Disable fields in Table Maintenance Generator dynamically. It pops-up a screen with a set of radio buttons, requesting the user to enter their option of fields to be enabled / disabled.

Author: Dhanalakshmi Dhivanand

Company: Cognizant Technology Solutions, Chennai, India

Created on: 27 January 2011

Author Bio



Dhanalakshmi is an ABAP Consultant. Currently working with Cognizant Technology Solutions. She has 5 years of experience in SAP ABAP. She is also interested in working with Enterprise Portal, Content Development & Customization using Web Dynpro for Java, and ABAP.

Table of Contents

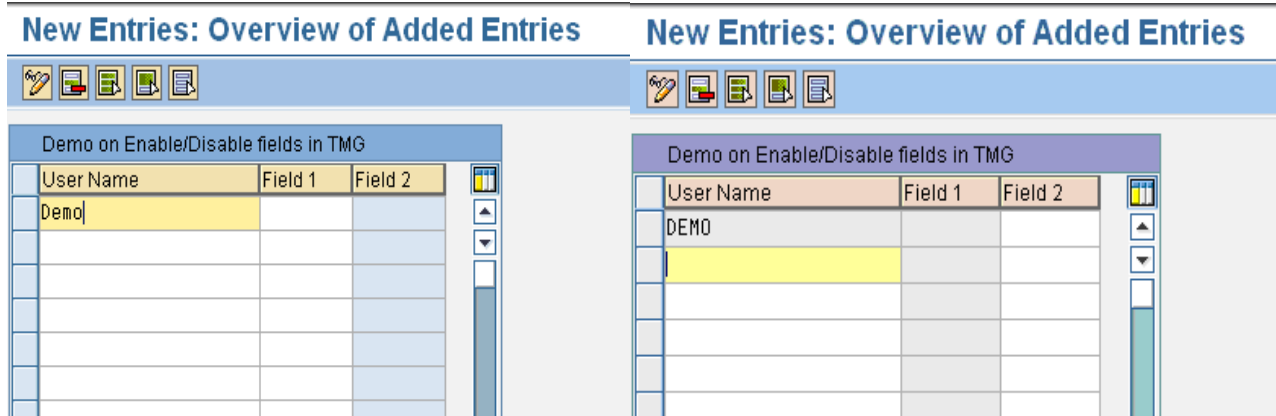
Introduction	3
Preview of Output	3
Merits and Demerits.....	3
Merits:.....	3
Demerits:.....	3
Procedure	3
Creation of Custom Table	4
Creation of Table Maintenance Generator in SE11 / SE54	6
Creation of Popup Screen.....	8
Working with TMG Events.....	17
Output.....	20
Related Content.....	23
Disclaimer and Liability Notice.....	24

Introduction

Fields in the table maintenance generator can be made Disabled at Design time. But this document demonstrates on enabling or disabling fields in table maintenance generator at runtime.

Preview of Output

At the end of this demonstration, the Table Maintenance screen in SM30 will look as show in the below screenshot.



Merits and Demerits

Merits:

Data can be maintained directly executing the table maintenance generator.

Demerits:

Once the Table Maintenance Generator screens are customized, the table should not be re-generated, as the screen changes will be lost.

Procedure

Below are the steps involved in to make the fields available / unavailable for input.

- Creation of Custom Table in SE11
- Creation of Table Maintenance Generator in SE11 / SE54
- Creation of popup screen
- Working with TMG (Table Maintenance Generator) Events
- Modification of auto generated code of TMG using SE80
- Output
- Merits and Demerits

Creation of Custom Table

Transaction Code: SE11

Transparent Table Name: ZTABLE

ABAP Dictionary: Initial Screen

Database table ZTABLE

View

Data type

Type Group

Domain

Search help

Lock object

Display Change Create

Maintain all the mandatory fields and Technical Settings.

Dictionary: Maintain Table

Transp. Table: ZTABLE New(Revised)

Short Description: Demo on Enable/Disable fields in TMG

Attributes **Delivery and Maintenance** Fields Entry help/check Currency/Quantity Fields

Delivery Class: C Customizing table, maintenance only by cust., not SAP import

Data Browser/Table View Maint.: Display/Maintenance Allowed

In the Fields tab, specify the following fields as mentioned in the below table.

Fields	Data Type
MANDT	MANDT
FIELD1	CHAR(8)
FIELD2	CHAR(8)

The activated table will look like the below mentioned screenshot.

Dictionary: Maintain Table

Transp. Table: ZTABLE Active
 Short Description: Demo on Enable/Disable fields in TMG

Attributes | Delivery and Maintenance | **Fields** | Entry help/check | Currency/Quantity Fields

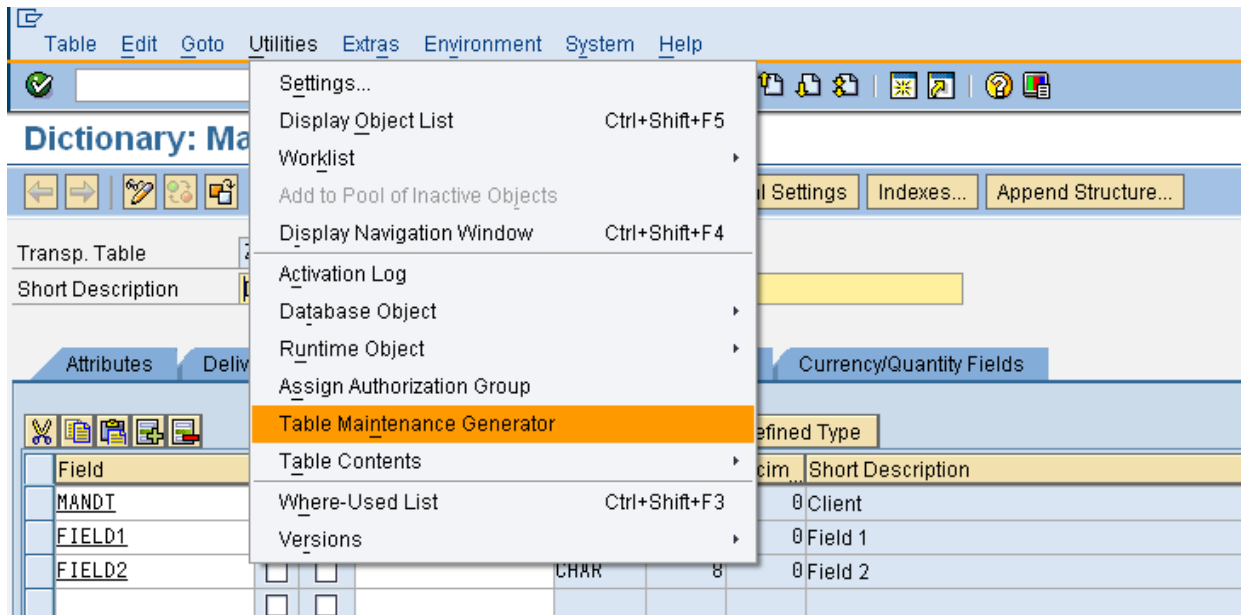
Technical Settings | Indexes... | Append Structure...

Srch Help | Data Element

Field	Key	Initi...	Data element	Data Ty...	Length	Decim...	Short Description
MANDT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	MANDT	CLNT	3		Client
FIELD1	<input type="checkbox"/>	<input type="checkbox"/>		CHAR	8		Field 1
FIELD2	<input type="checkbox"/>	<input type="checkbox"/>		CHAR	8		Field 2
	<input type="checkbox"/>	<input type="checkbox"/>					

Creation of Table Maintenance Generator in SE11 / SE54

To create table maintenance generator, go to SE11 transaction. Navigate to Menu Utilities → Table Maintenance Generator.



The below screen will appear.

Generate Table Maintenance Dialog: Generation Environment

Find Scr. Number(s)

Table/View: ZTABLE

Technical Dialog Details

Authorization Group: &NC& w/o auth. group
 Authorization object: S_TABU_DIS
 Function group: ZTABLE
 Package:

Maintenance Screens

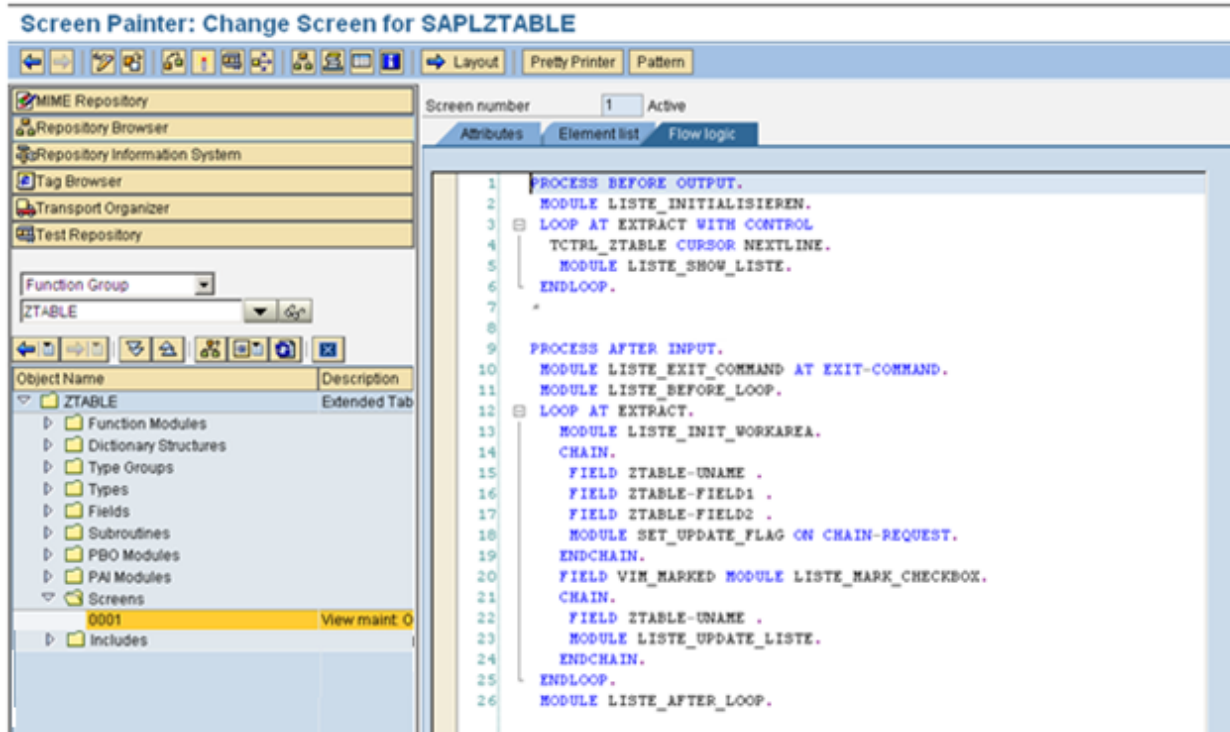
Maintenance type: one step
 two step
 Maint. Screen No.: Overview screen: 1
 Single screen:

Dialog Data Transport Details

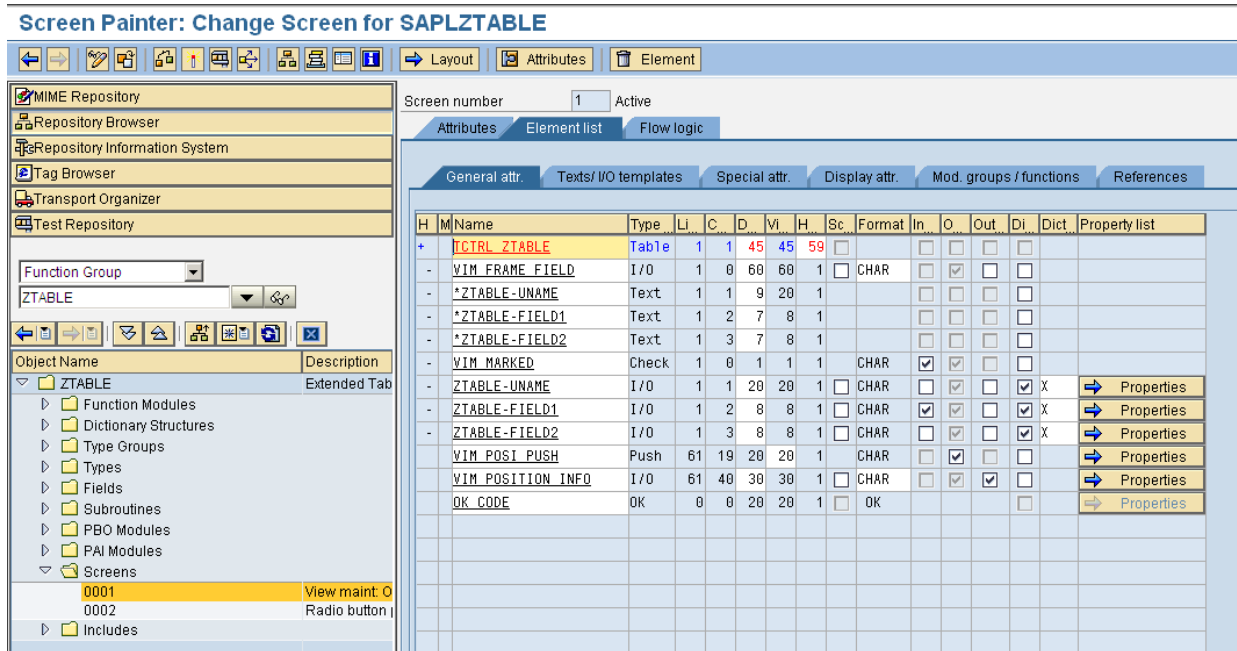
Recording routine: Standard recording routine
 no, or user, recording routine
 Compare Flag: Automatically Adjustable Note

Enter all the required fields and click on Create button in maintenance screen

Double Click on the Overview Screen Number, which will take you to the screen painter.



Click on the Element list tab and uncheck the INPUT check box of the field ZTABLE-FIELD2, so as to set the FIELD2 disabled initially when the table maintenance screen is loaded in the screen.



Creation of Popup Screen

Create another screen '0002' by right clicking on the Screen folder node.

Screen Painter: Change Screen for SAPLZTABLE

Repository Browser

Function Group: ZTABLE

Object Name: ZTABLE | Description: Extended Tab

Screen number: 2 | Inactive

Attributes | Element list | Flow logic

Short Description: Radio Button Popup

Original Language: EN | English | Package: \$TMP

Last changed on/at: 12/03/2010 02:00:49

Last Generation: 0:00:00

Screen Type:

- Normal
- Subscreen
- Modal dialog box
- Selection screen

Settings:

- Hold Data
- Switch Off Runtime Compress
- Template - non-executable
- Hold Scroll Position
- Without Application Toolbar

Other Attributes:

Next Screen: 2

Cursor Position: []

Screen Group: []

Lines/Columns:

Occupied	5	26
Mainten.	5	33

Context Menu FORM ON CTMENU: []

Properties

A dialog appears requesting for the screen number. Provide '0002' as the screen number.

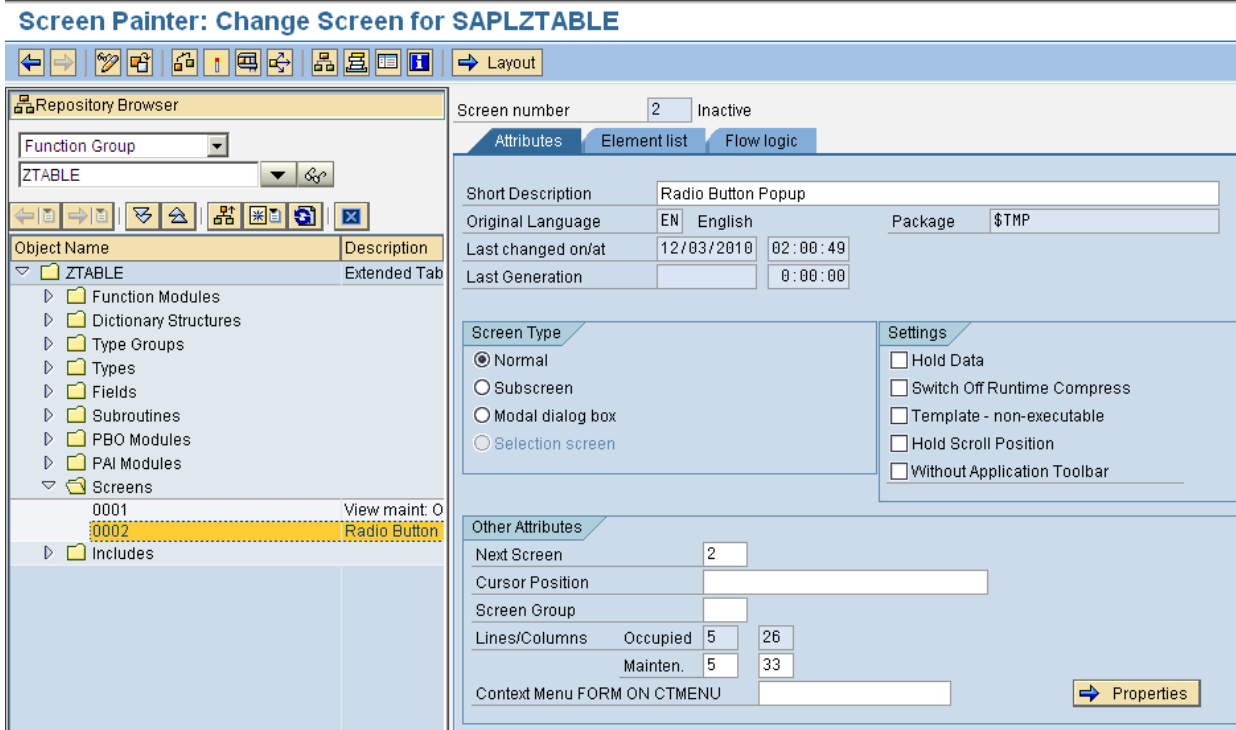
Create Screen

Program: SAPLZTABLE

Screen number: 0002

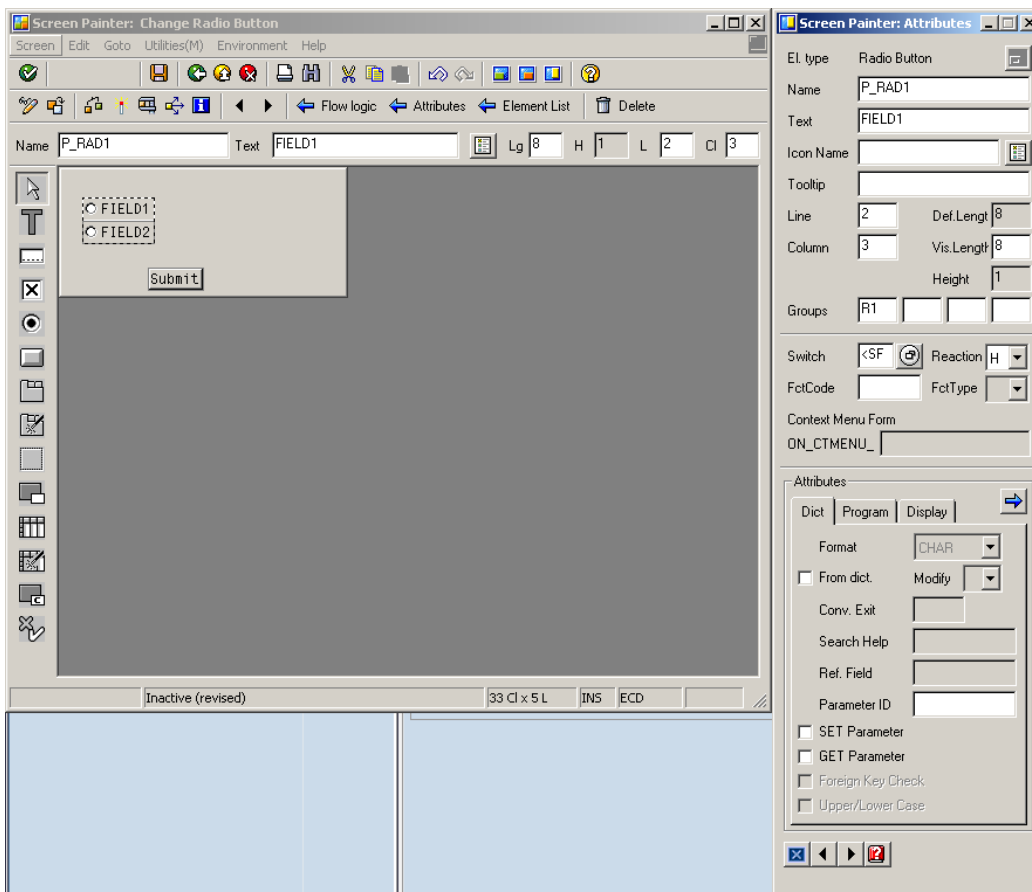
OK Cancel

Provide all the required inputs as shown in the below screenshot.



Click on the Layout Button in the top of the screen painter.

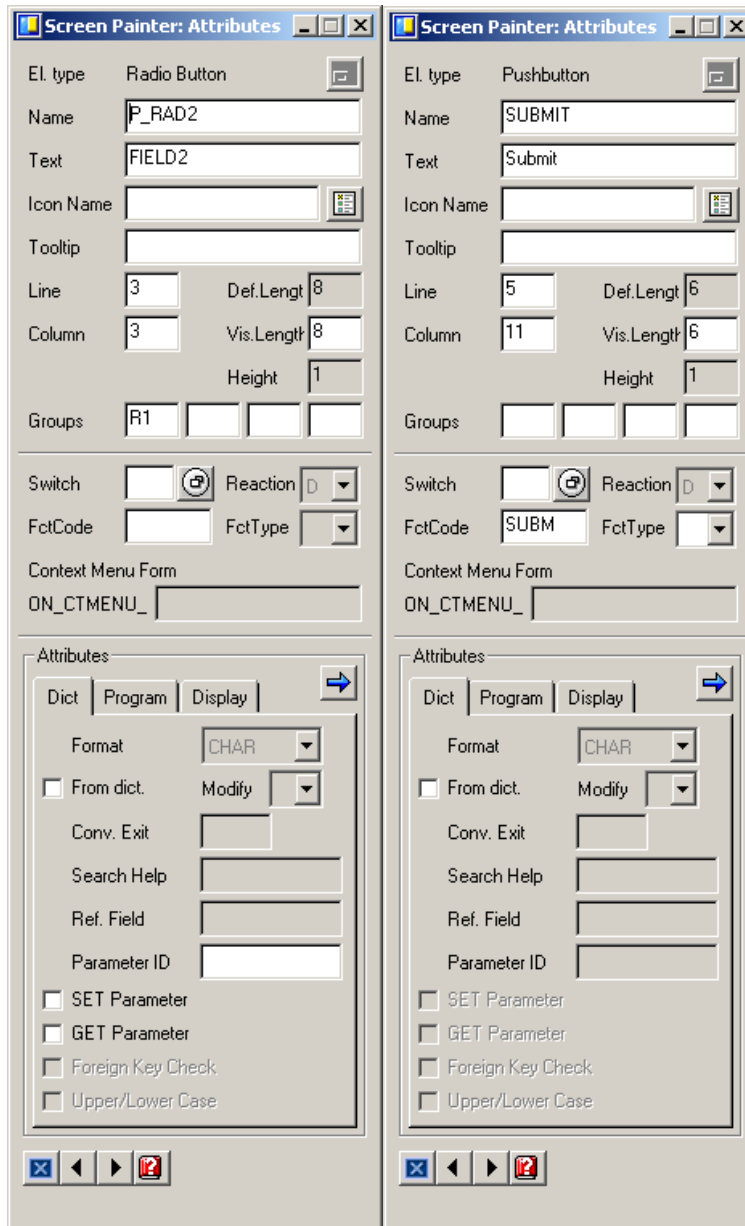
The Layout of Screen No. '0002' appears. Place 2 radio buttons to make fields enable / disable according to the selection of radio button.



Name the elements as show in the below table.

ELEMENT	NAME	TEXT
Radio button 1	P_RAD1	FIELD1
Radio button 2	P_RAD2	FIELD2
Command Button	SUBMIT	Submit

Place a command button and name as 'SUBMIT'. Provide 'SUBM' as function code in FctCode field.



The above screenshot shows the attributes of the UI elements placed in the screen '0002'.

Go to Element list of the screen '0002'. Include OK_CODE in the element list.

Screen Painter: Change Screen for SAPLZTABLE

Screen number: 2 Inactive

H	M	Name	Type	Li	C	D	Vi	H	Sc	Format	In	O	Out	Di	Dict	Property list
		P_RAD1	Radio	2	3	1	1	1		CHAR	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		Properties
		P_RAD1	Radio	2	5	6	6	1			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		Properties
		P_RAD2	Radio	3	3	1	1	1		CHAR	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		Properties
		P_RAD2	Radio	3	5	6	6	1			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		Properties
		SUBMIT	Push	5	11	6	6	1			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		Properties
		OK_CODE	OK	0	0	20	20	1		OK	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		Properties

Declare these 2 radio buttons in the TOP include of the TMG program.

ABAP Editor: Change Include LZTABLETOP

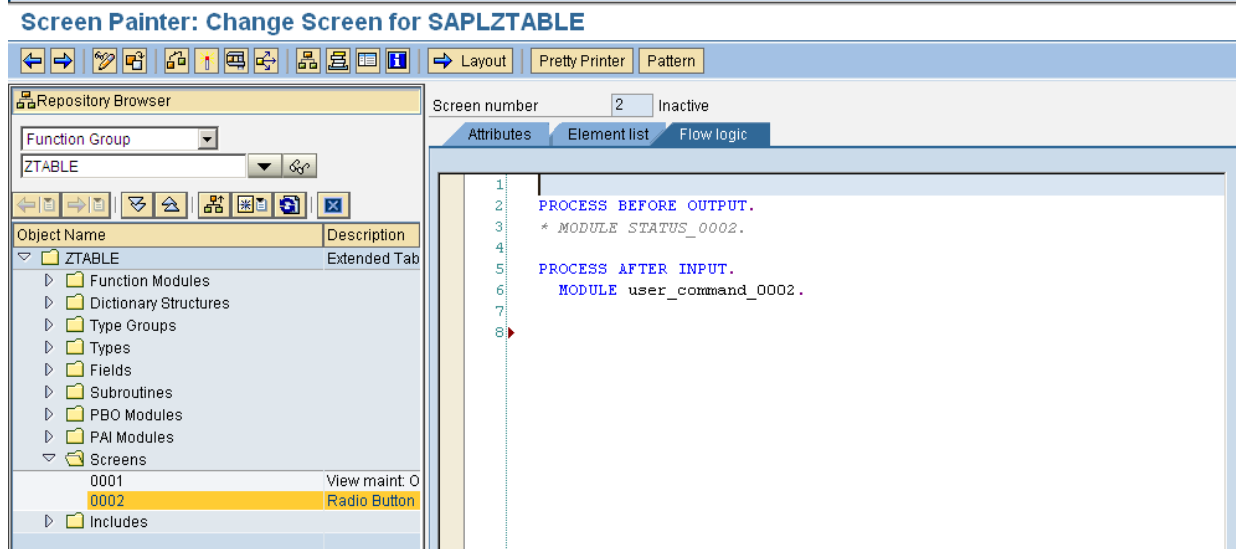
```

1  *-----*
2  * generated viewmaintenance function pool top
3  * generation date: 12/03/2010 at 01:55:51 by user DD47
4  * view maintenance generator version: #001407#
5  *-----*
6  FUNCTION-POOL ZTABLE                MESSAGE-ID SV.
7
8  INCLUDE LSVIMDAT                      . "general data decl.
9  INCLUDE LZTABLETOO                    . "view rel. data decl.
10
11 DATA: P_RAD1 TYPE C,
12         P_RAD2 TYPE C.
    
```

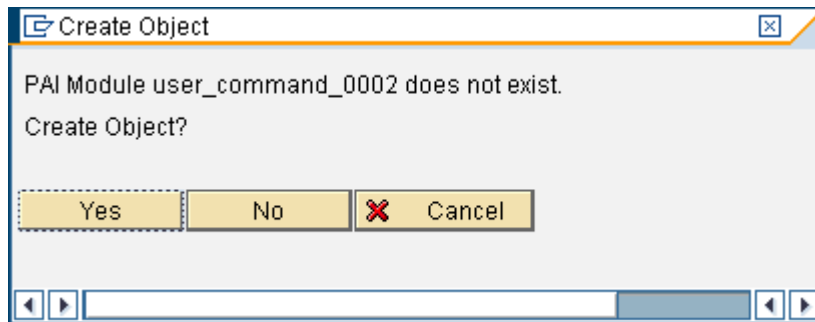
```

DATA: P_RAD1 TYPE C,
      P_RAD2 TYPE C.
    
```

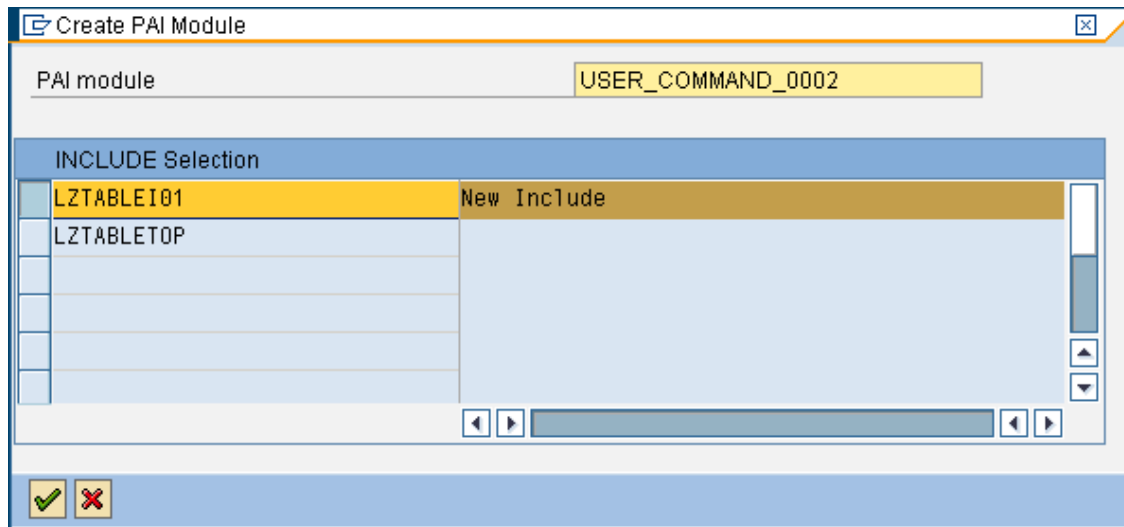
In the flow logic part, include a module USER_COMMAND_0002 and double click on the module.



A popup appears as shown below. Click on 'Yes' in the appearing screen.



Create this module in the new INCLUDE program LZTABLEI01.



An editor opens to insert code in the module USER_COMMAND_0002.

ABAP Editor: Change Include LZTABLEI01

The screenshot shows the ABAP Editor interface. The title bar reads 'ABAP Editor: Change Include LZTABLEI01'. The left sidebar contains a 'Repository Browser' with various navigation options like 'MIME Repository', 'Repository Information System', 'Tag Browser', 'Transport Organizer', and 'Test Repository'. Below this is a 'Function Group' dropdown set to 'ZTABLE'. The main area is split into two panes. The left pane shows a list of objects with columns 'Object Name' and 'Description'. The right pane shows the code for the include 'LZTABLEI01', which is currently 'Inactive (Revised)'. The code includes a comment '***INCLUDE LZTABLEI01 .', a module definition 'Module USER_COMMAND_0002 INPUT', and a 'text' section. The code is enclosed in a 'module USER_COMMAND_0002 input.' block.

Write the below code in the USER_COMMAND_0002 INPUT module.

```

*-----*
***INCLUDE LZTABLEI01 .
*-----*
*&-----*
*&      Module  USER_COMMAND_0002  INPUT
*&-----*
*      text
*-----*
module USER_COMMAND_0002 input.

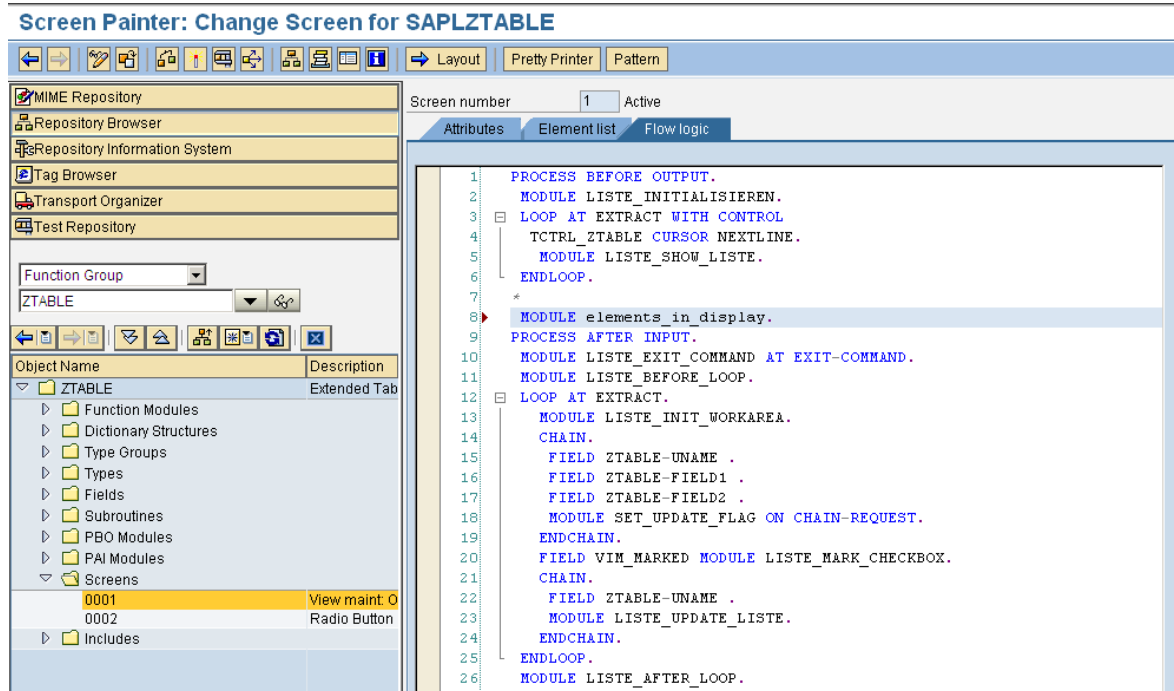
  CASE ok_code.
    WHEN 'SUBM'.
      SET SCREEN 0.
      LEAVE SCREEN.
  ENDCASE.

endmodule.          " USER_COMMAND_0002  INPUT

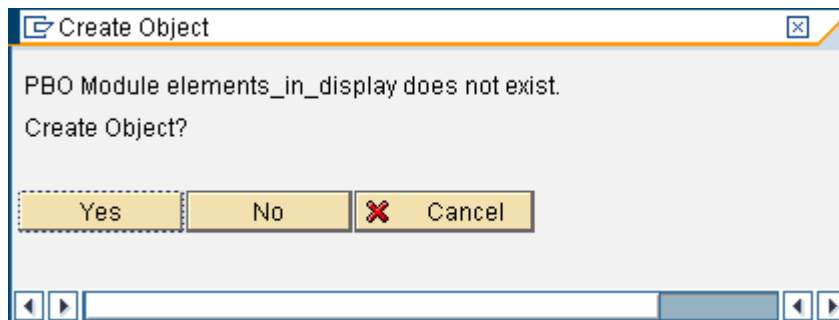
```

Modification of auto generated code of TMG using SE80

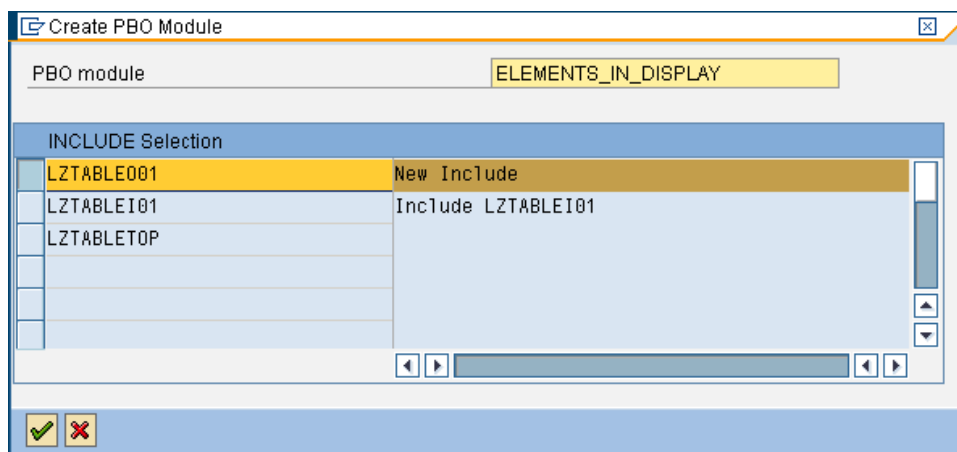
In order to enable / disable fields in the TMG according to the radio button selection, create a PBO module ELEMENTS_IN_DISPLAY in the auto generated code of TMG as show in the below screenshot.



Now double click on the PBO module, which brings pop up as shown below.



Click on 'Yes' Button in the appearing dialog.



Create an include 'LZTABLE001' to incorporate the code of PBO module 'ELEMENTS_IN_DISPLAY'. The Screen appears as shown in the below screenshot.

ABAP Editor: Change Include LZTABLE001

Include LZTABLE001 Inactive (Revised)

```

1  *-----*
2  ***INCLUDE LZTABLE001 .
3  *-----*
4  *&-----*
5  *&      Module  ELEMENTS_IN_DISPLAY  OUTPUT
6  *&-----*
7  *      text
8  *-----*
9  module ELEMENTS_IN_DISPLAY output.
10
11  endmodule.
12  " ELEMENTS IN DISPLAY OUTPUT

```

Object Name | Description

LSVIMF46	Include LS
LSVIMF50	Include LS
LSVIMF51	Include LS
LSVIMF5A	Include LS
LSVIMF5B	Include LS
LSVIMF5C	VIM_SET_
LSVIMF5D	VIM_MAIN
LSVIMF5E	maintain
LSVIMF5F	VIM_READ
LSVIMF60	Include LS
LSVIMF61	map keys
LSVIMF62	Include LS
LSVIMF63	Include LS
LSVIMF64	Include LS
LSVIMF65	Include LS
LSVIMF1	database
LSVIMFX	extended
LSVIMF2	Form rout
LSVIMF3	form rout
LSVIMFXL	Subprogr
LSVIMFXP	Subroutine
LSVIMFXX	?
LSVIMTX	
LSVIMXX	
LSVIMXX	
LSVIMDT	Table-spe
LZTABLE00	
LZTABLE00	
LZTABLE01	Include LZ
LZTABLE01	Include LZ

Write the below code in the module ELEMENTS_IN_DISPLAY OUTPUT.

ABAP Editor: Change Include LZTABLE001

Include LZTABLE001 Active

```

1  *-----*
2  ***INCLUDE LZTABLE001 .
3  *-----*
4  *&-----*
5  *&      Module  ELEMENTS_IN_DISPLAY  OUTPUT
6  *&-----*
7  *      text
8  *-----*
9  module ELEMENTS_IN_DISPLAY output.
10  IF p_rad1 EQ 'X'.
11  LOOP AT SCREEN.
12  IF screen-name EQ 'ZTABLE-FIELD1'.
13  screen-input = 1.
14  MODIFY SCREEN.
15  ENDIF.
16  IF screen-name EQ 'ZTABLE-FIELD2'.
17  screen-input = 0.
18  MODIFY SCREEN.
19  ENDIF.
20  ENDOLOOP.
21  ELSE.
22  LOOP AT SCREEN.
23  IF screen-name EQ 'ZTABLE-FIELD2'.
24  screen-input = 1.
25  MODIFY SCREEN.
26  ENDIF.
27  IF screen-name EQ 'ZTABLE-FIELD1'.
28  screen-input = 0.
29  MODIFY SCREEN.
30  ENDIF.
31  ENDOLOOP.
32  ENDIF.
33
34  endmodule.

```

Object Name | Description

ZTABLE	Extended Tab
Function Modules	
Dictionary Structures	
Type Groups	
Types	
Fields	
Subroutines	
PBO Modules	
PAI Modules	
Screens	
0001	View maint: 0
0002	Radio button
Includes	

```

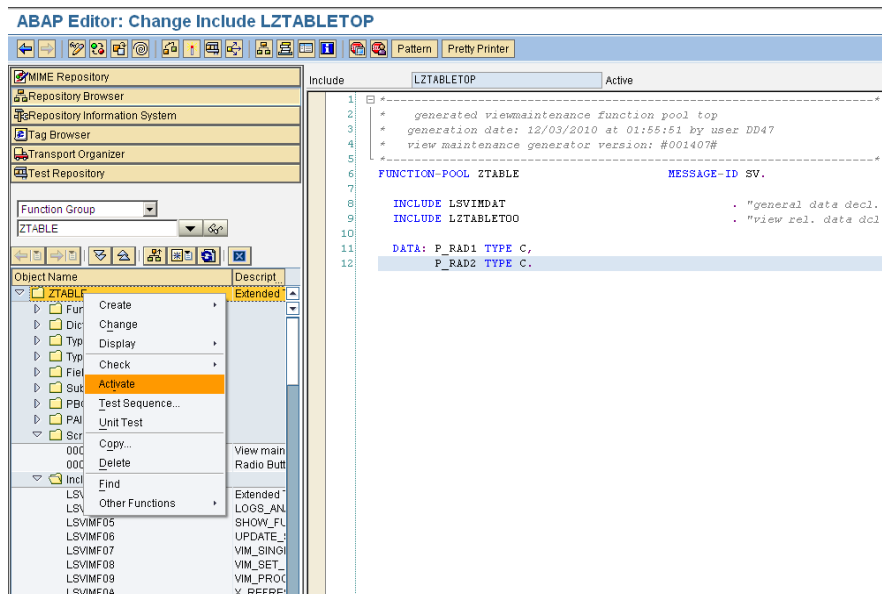
*-----*
***INCLUDE LZTABLE001 .
*-----*
*&-----*
*&      Module  ELEMENTS_IN_DISPLAY  OUTPUT
*&-----*
*      text
*-----*

module ELEMENTS_IN_DISPLAY output.
  IF p_rad1 EQ 'X'.
    LOOP AT SCREEN.
      IF screen-name EQ 'ZTABLE-FIELD1'.
        screen-input = 1.
        MODIFY SCREEN.
      ENDIF.
      IF screen-name EQ 'ZTABLE-FIELD2'.
        screen-input = 0.
        MODIFY SCREEN.
      ENDIF.
    ENDLOOP.
  ELSE.
    LOOP AT SCREEN.
      IF screen-name EQ 'ZTABLE-FIELD2'.
        screen-input = 1.
        MODIFY SCREEN.
      ENDIF.
      IF screen-name EQ 'ZTABLE-FIELD1'.
        screen-input = 0.
        MODIFY SCREEN.
      ENDIF.
    ENDLOOP.
  ENDIF.

endmodule.                " ELEMENTS_IN_DISPLAY  OUTPUT

```

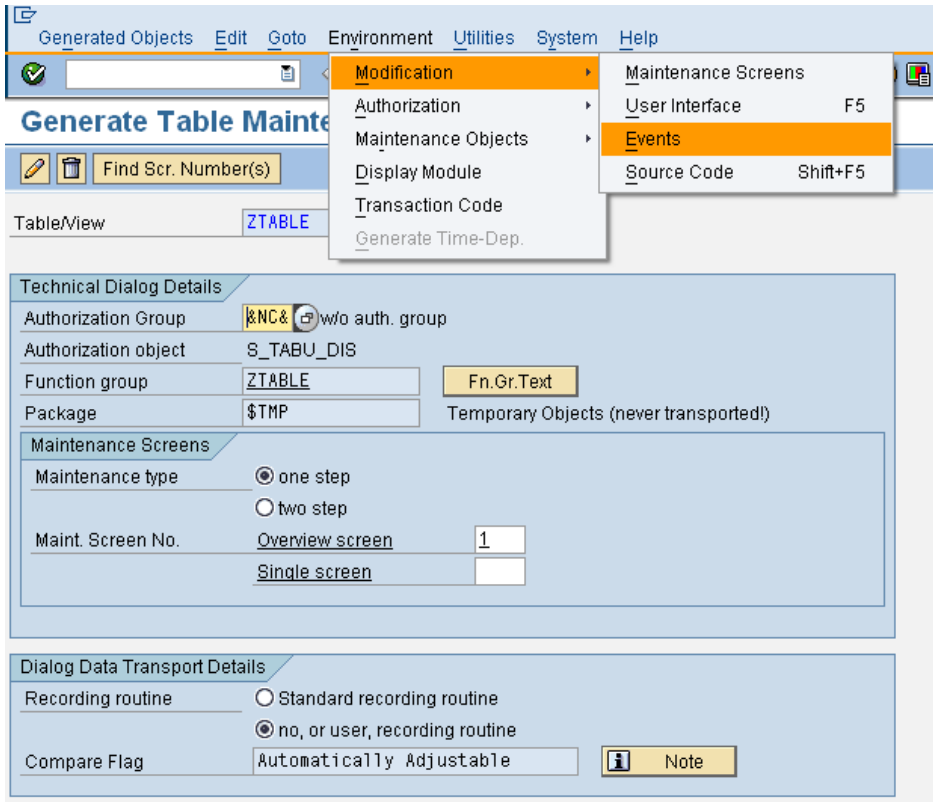
Click on Activate, but right clicking on the main program.



Working with TMG Events

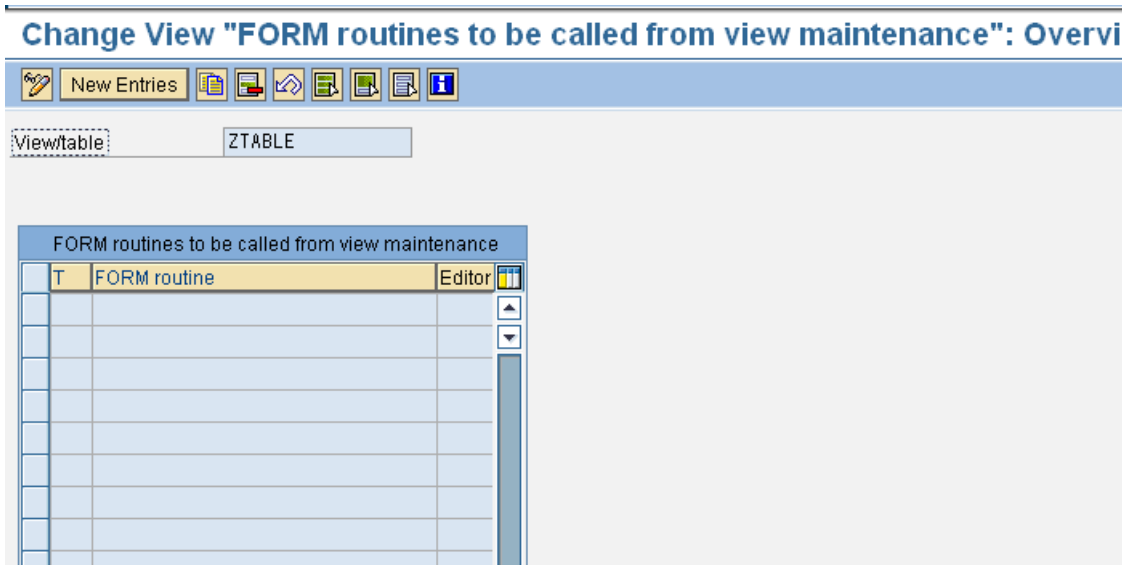
To call the pop screen, create an event in the table maintenance generator. To achieve this. Go to Transaction code SE11, and enter 'ZTABLE'.

Navigate to the menu section, Utilities → Table Maintenance Generator.

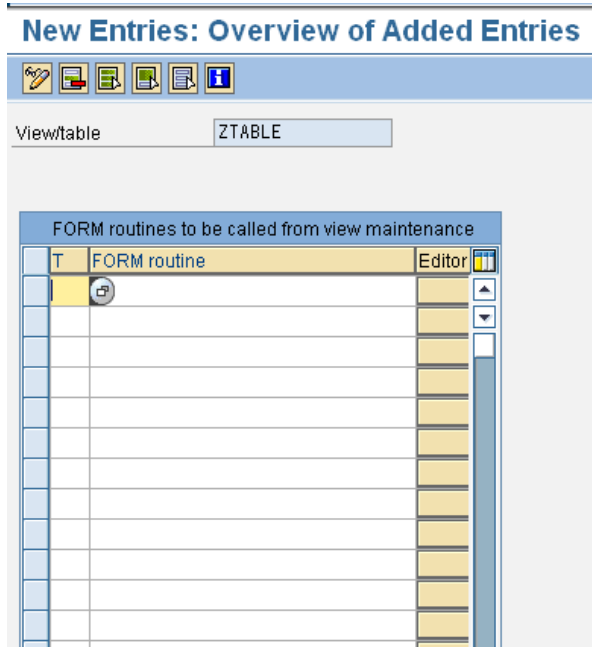


Now again navigate to Environment → Modification → Events.

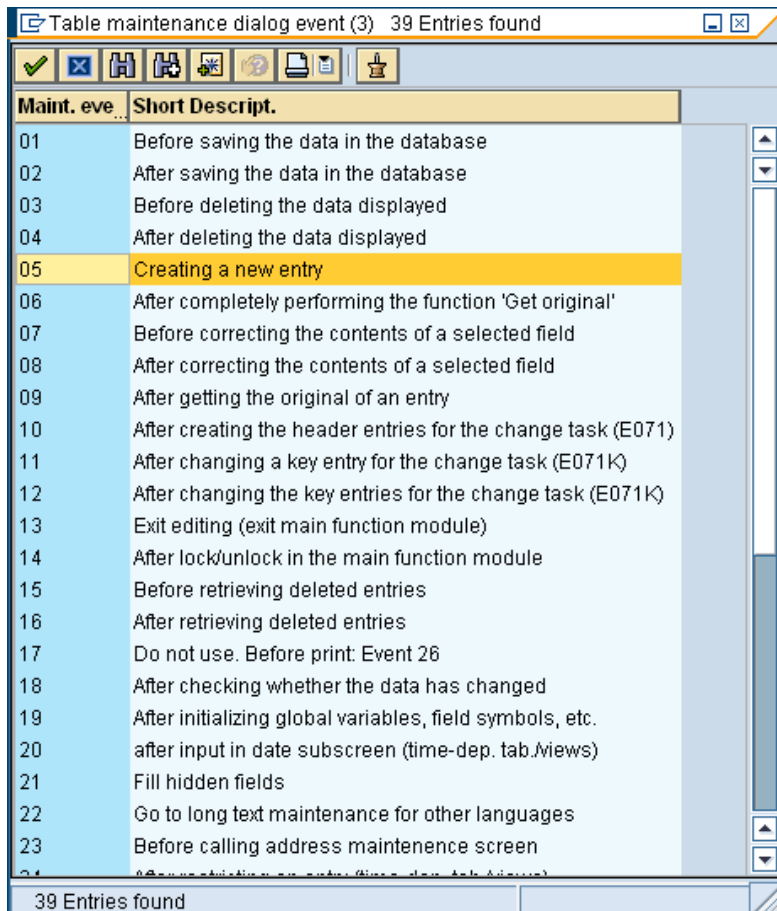
A screen appears as shown in the below screenshot.



Click on 'New Entries' button.



Press F4 function key to check the available events of TMG.



Select '05' from the appearing list, to incorporate code for calling the popup screen.

New Entries: Overview of Added Entries

View/table: ZTABLE

FORM routines to be called from view maintenance		
T	FORM routine	Editor
05	AFTER_INPUT	

Create your own routine. In this case, 'AFTER_INPUT' is the routine.

When this routine is double clicked, a dialog appears as shown below.

Create Subroutine

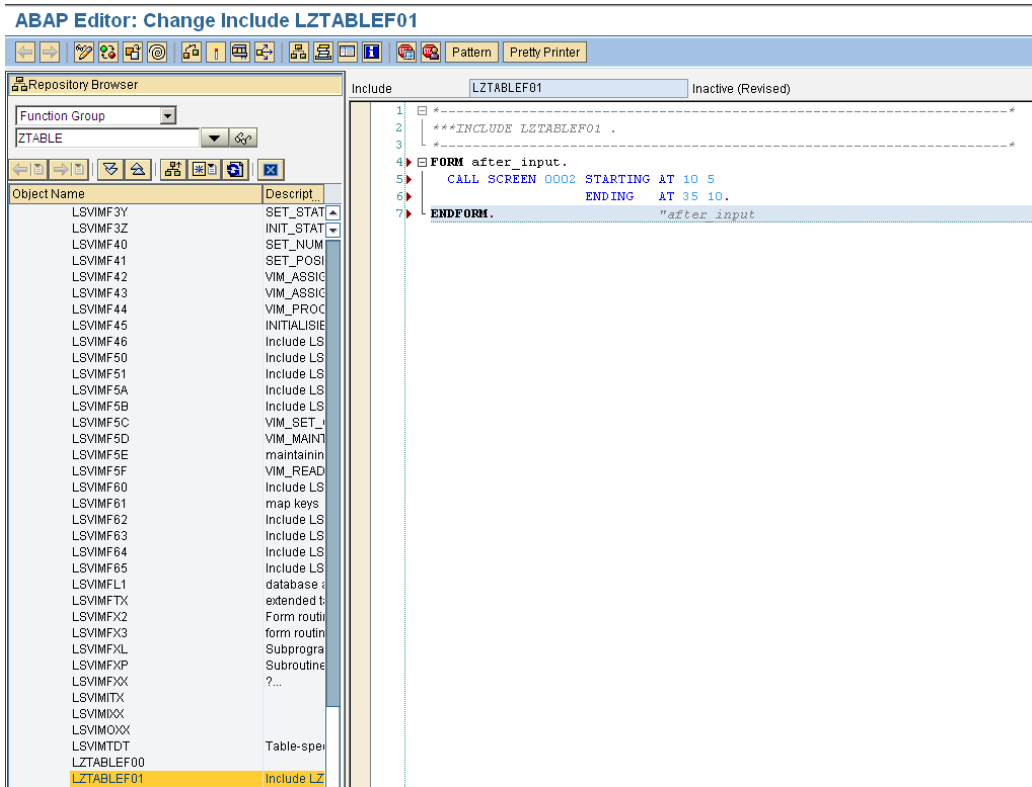
Subroutine: AFTER_INPUT

INCLUDE Selection	
LZTABLEF01	New Include
LZTABLEI01	Include LZTABLEI01
LZTABLE001	Include LZTABLE001
LZTABLETOP	

Create this routine in the new include 'LZTABLEF01'.

Write the below piece of code in the appearing window.

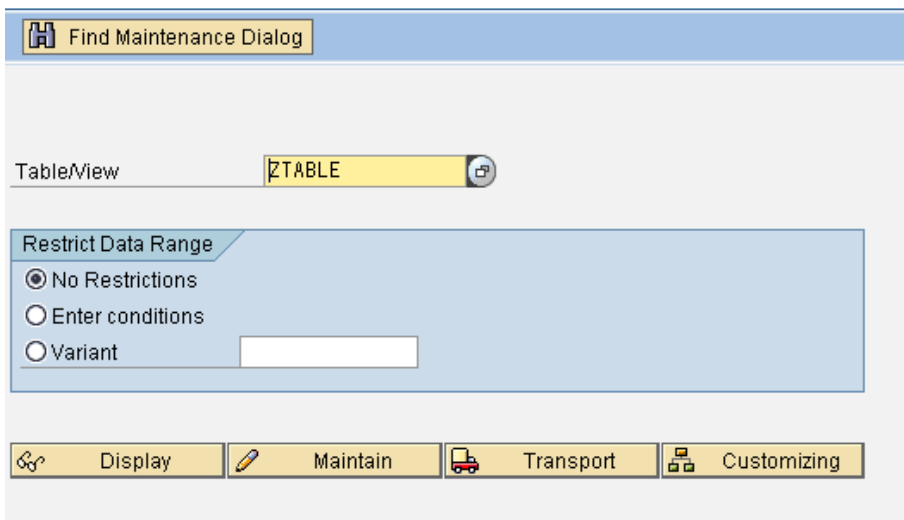
```
FORM after_input.
  CALL SCREEN 0002 STARTING AT 10 5
                        ENDING   AT 35 10.
ENDFORM.               "after_input
```



Output

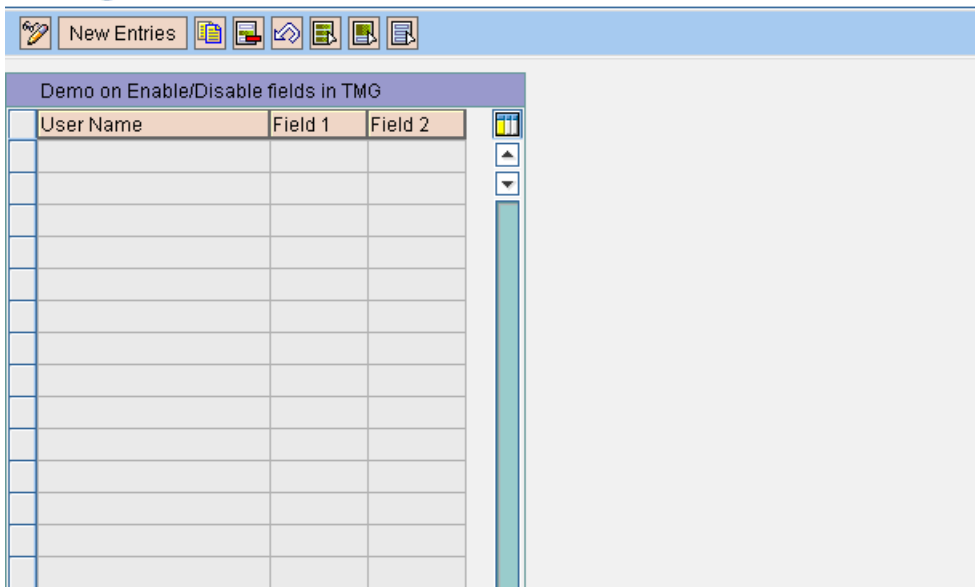
To test the environment, go to SM30 and enter the table name.

Maintain Table Views: Initial Screen



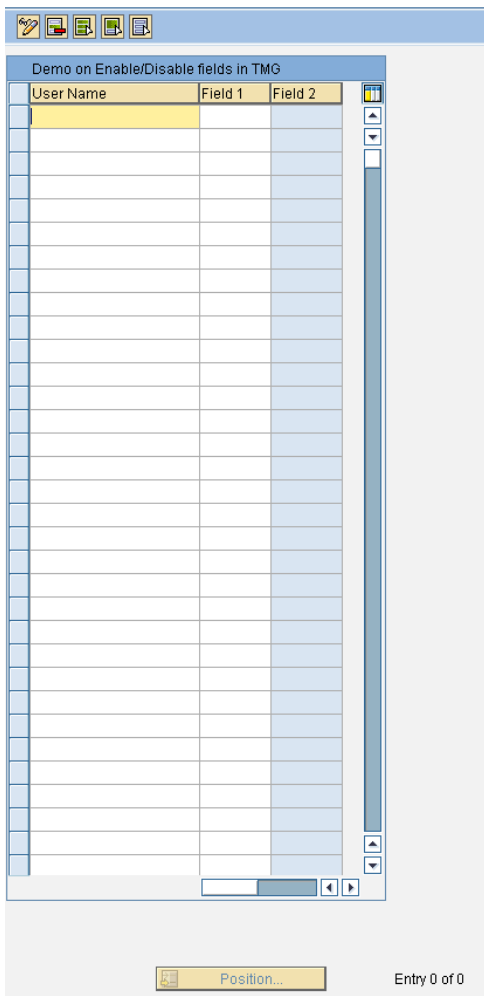
Click on the 'Maintain' button.

Change View "Demo on Enable/Disable fields in TMG": Overview



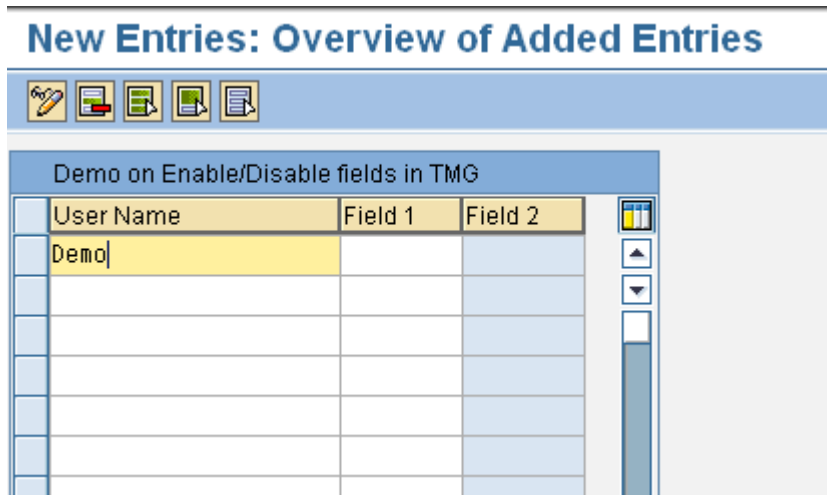
Click on the 'New Entries' button.

New Entries: Overview of Added Entries

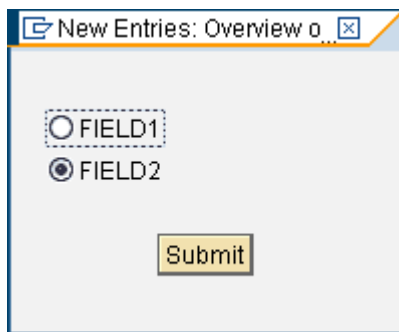


In the above screen, the field 'FIELD2' is disabled initially from input.

Enter a record and hit 'ENTER'.

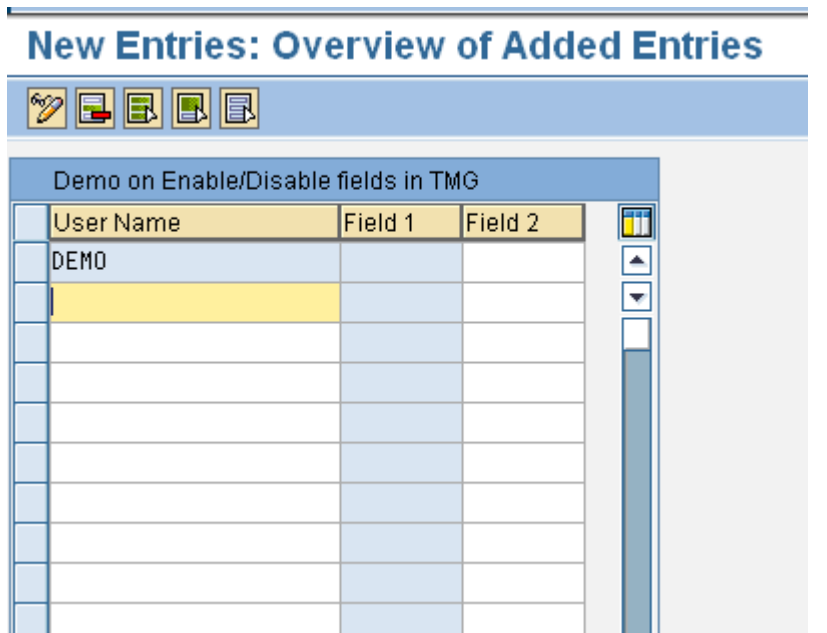


A popup screen appears as shown below.



Select the radio button 'FIELD2' from the appearing popup screen and click on SUBMIT button.

The TMG screen appears as shown below.



Thus the fields can be made enable / disable at runtime.

Related Content

For more information, visit the [ABAP homepage](#)

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.