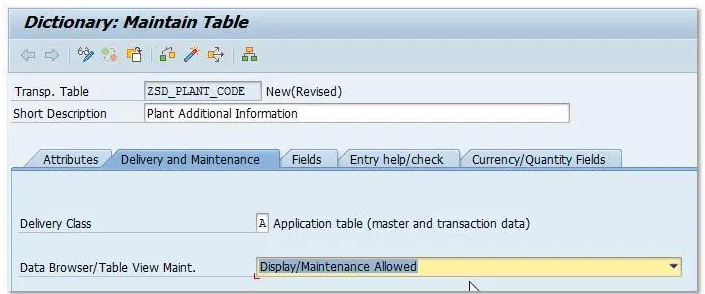
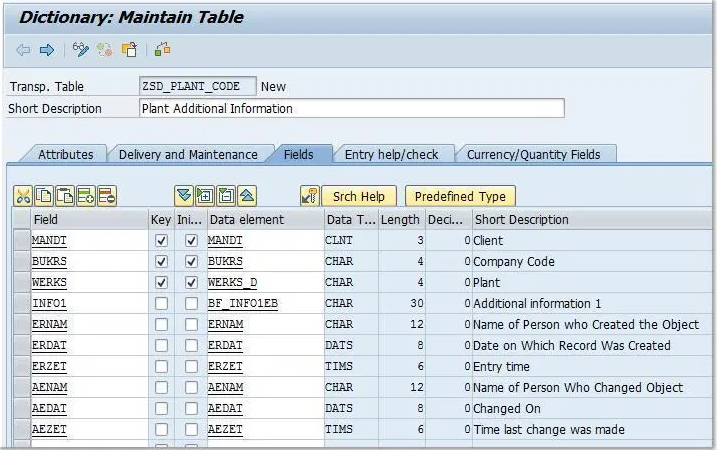
# Enhanced Table Maintenance with Automatic Change Recording

Table maintenance generator is very powerful tool to generate simple data entry screen for tables. For tables which store configuration data i.e. rarely modified it not worth spending time creating module pool program for data entry. Table maintenance generator also gives programmer opportunity to modify standard behaviour by including additional logic. In this blog I will enhance standard table maintenance to automatically record created by and created date/time and also last modified by and last modified date/time.  
  
I have this table which record additional information per plant. Instead of hard-coding this text we decided to create a table and put these values there in case it requires change in future.

Create table with ‘Display/Maintenance Allowed’



I have fields ERNAM, ERDAT, ERZET, AENAM, AEDAT, AEZET apart from fields which will hold actual data.

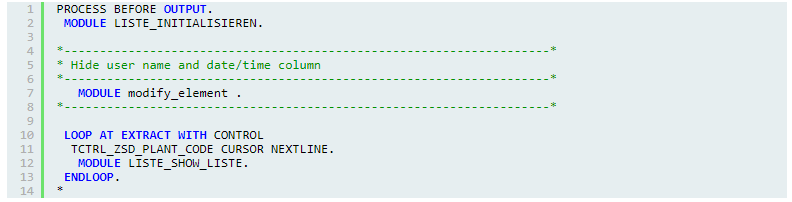


Create table maintenance from menu option Utilities->Table Maintenance Generator. Make sure to select ‘one step’ in Maintenance type.

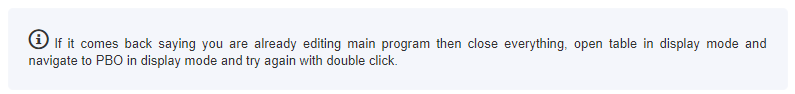
With standard table maintenance you will have all field active and ready for input. Since we username and date times field to be filled automatically by program logic we need to hide them from table maintenance screen. To do that, in table maintenance screen double click on screen number.



It will take you to the flow logic of screen. Add module modify\_element in PBO of screen.



Save and double click on modify\_element module to create module in new include.



In module modify\_element copy following code. This code is going through the table control structure to hide columns.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | PROCESS BEFORE OUTPUT.   MODULE LISTE\_INITIALISIEREN.    \*--------------------------------------------------------------------\*  \* Hide user name and date/time column  \*--------------------------------------------------------------------\*     MODULE modify\_element .  \*--------------------------------------------------------------------\*     LOOP AT EXTRACT WITH CONTROL    TCTRL\_ZSD\_PLANT\_CODE CURSOR NEXTLINE.     MODULE LISTE\_SHOW\_LISTE.   ENDLOOP.  \* |

Save and double click on modify\_element module to create module in new include.

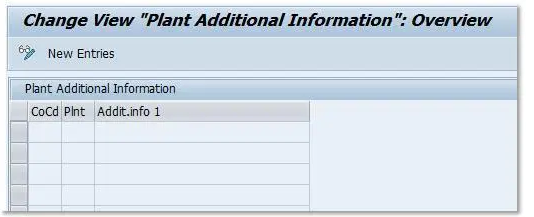
 If it comes back saying you are already editing main program then close everything, open table in display mode and navigate to PBO in display mode and try again with double click.

In module modify\_element copy following code. This code is going through the table control structure to hide columns.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | \*----------------------------------------------------------------------\*  \*\*\*INCLUDE LZSD\_PLANT\_CODEO01 .  \*----------------------------------------------------------------------\*  \*&---------------------------------------------------------------------\*  \*&      Module  MODIFY\_ELEMENT  OUTPUT  \*&---------------------------------------------------------------------\*  \*       text  \*----------------------------------------------------------------------\*  MODULE modify\_element OUTPUT.      LOOP AT <vim\_tctrl>-cols INTO vim\_tc\_cols.      IF vim\_tc\_cols-screen-name = 'ZSD\_PLANT\_CODE-ERNAM' OR         vim\_tc\_cols-screen-name = 'ZSD\_PLANT\_CODE-ERDAT' OR         vim\_tc\_cols-screen-name = 'ZSD\_PLANT\_CODE-ERZET' OR         vim\_tc\_cols-screen-name = 'ZSD\_PLANT\_CODE-AENAM' OR         vim\_tc\_cols-screen-name = 'ZSD\_PLANT\_CODE-AEDAT' OR         vim\_tc\_cols-screen-name = 'ZSD\_PLANT\_CODE-AEZET' .          vim\_tc\_cols-invisible = 1 .        MODIFY <vim\_tctrl>-cols FROM vim\_tc\_cols .      ENDIF.    ENDLOOP.    ENDMODULE.                 " MODIFY\_ELEMENT  OUTPUT |

Save and activate all objects. Make sure the function group where table maintenance generator code is active including screen and our new include.

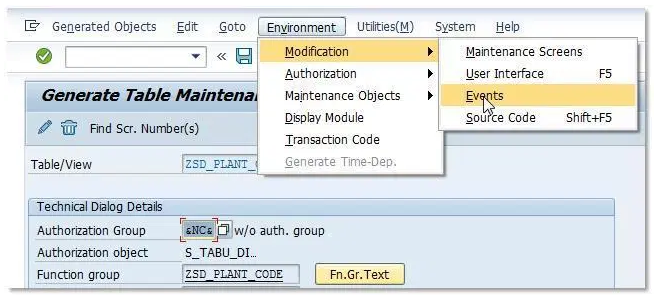
If you run table maintenance now you should not see these field.



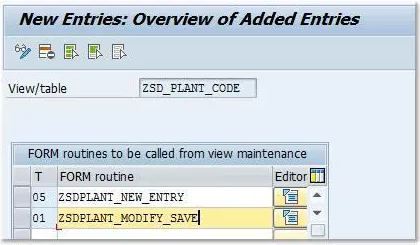
Now it’s time to add code which will fill user name, date and time when records are created or changed.

Table Maintenance Generator offers events that during program processing and we programmer can add our own code to execute at this point. Very similar to user-exits. Events in which we are interested in are 05 which executes when new entry is create and event 01 which triggers before saving the data in database. We are going to add code at these event which will populate our fields.

Open Table Maintenance Generator in change mode and choose menu option Environment->Modification->Events.



Click ok on dialog box which say do not change SAP data. On next screen press ‘New Entries’ and add modules at 05 and 01 events.



Click on the editor button next to form routine, create include if required, and put following code in each of these form routines.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | \*&---------------------------------------------------------------------\*  \*&      Form  ZSDPLANT\_NEW\_ENTRY  \*&---------------------------------------------------------------------\*  FORM zsdplant\_new\_entry .    zsd\_plant\_code-ernam = sy-uname .    zsd\_plant\_code-erdat = sy-datum .    zsd\_plant\_code-erzet = sy-uzeit .  ENDFORM.                    "zsdplant\_new\_entry |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33 | \*&---------------------------------------------------------------------\*  \*&      Form  ZSDPLANT\_MODIFY\_SAVE  \*&---------------------------------------------------------------------\*  FORM zsdplant\_modify\_save .      FIELD-SYMBOLS : <fs\_field> TYPE ANY .      LOOP AT total .      CHECK <action> EQ aendern.      ASSIGN COMPONENT 'AENAM' OF STRUCTURE <vim\_total\_struc> TO <fs\_field> .      IF sy-subrc = 0 .        <fs\_field> = sy-uname .      ENDIF.        ASSIGN COMPONENT 'AEDAT' OF STRUCTURE <vim\_total\_struc> TO <fs\_field> .      IF sy-subrc = 0 .        <fs\_field> = sy-datum .      ENDIF.        ASSIGN COMPONENT 'AEZET' OF STRUCTURE <vim\_total\_struc> TO <fs\_field> .      IF sy-subrc = 0 .        <fs\_field> = sy-uzeit .      ENDIF.        READ TABLE extract WITH KEY <vim\_xtotal\_key>.      IF sy-subrc = 0.        extract = total .        MODIFY extract INDEX sy-tabix.      ENDIF.        MODIFY total.    ENDLOOP.  ENDFORM .                    "ZSDPLANT\_MODIFY\_SAVE |

Save and Activate all objects. To activate function group you can go to SE80 and activate from there.

I have created two entries in table maintenance and then changed one. This is how table contents look like.

