

ABAP: Table Maintenance Events



Applies to:

This document applies to SAP ECC 6.0, SAP Netweaver 2004s. For more information, visit the [ABAP homepage](#).

Summary

This article contains information on the various Table Maintenance Generator events and the step by step procedure to implement these events in SAP ECC 6.0 version.

Author: Renjith R Thampi

Company: Appexus Software Solutions (P) Ltd

Created on: 20 September 2010

Author Bio

Renjith R Thampi is working as SAP Technology Consultant with Appexus Software Solutions (P) Ltd.

Table of Contents

Introduction	3
Table Maintenance Events	4
Event 01: Before saving the data in the database	4
Event 02: After saving the data in the database	4
Event 03: Before deleting the display data	4
Event 04: After deleting the display data	4
Event 05: When creating a new entry	4
Event 06: After the 'Get original' function.....	4
Event 07: Before correcting the contents of a selected field.....	4
Event 08: After correcting the contents of a selected field.....	4
Event 09: After 'Get original' of an entry	4
Event 10: After creating the change request header entries (E071)	4
Event 11: After changing a key entry in the change request (E071K).....	4
Event 12: After changing a key entry in the change request (E071K).....	5
Event 13: End of processing (leaving the main function module).....	5
Event 14: After locking/unlocking in the main function module	5
Event 15: Before retrieving deleted entries.....	5
Event 16: After retrieving deleted entries.....	5
Event 17: Before printing entries.....	5
Event 18: After the data change check	5
Event 19: After initialization of global variables, field symbols, etc.....	5
Event 20: After input in the date sub screen (time-dep. tables/views).....	5
Event 21: Fill hidden fields	5
Event 22: Go to long text maintenance for other languages.....	5
Event 23: Before the address maintenance screen call	6
Event 25: At the start of the maintenance dialog.	6
Event 26: Before displayed data is output in a list	6
Event AA: Instead of the standard data read routine	6
Event AB: Instead of the standard database changes routine	6
Event AC: Instead of the standard 'Get original' routine	6
Event AD: Instead of the standard RO field read routine.....	6
Event AE: Instead of standard positioning coding	6
Event AF: When reading texts in other languages	6
Event AG: Instead of 'Get original' for texts in other languages	6
Event AH: Instead of DB change for texts in other languages.....	6
Event ST: GUI menu main program name.....	6
Step by Step Process to Implement Events	7
Related Content.....	14
Disclaimer and Liability Notice.....	15

Introduction

Table Maintenance Generator is a tool to create and interface through which data maintained in the custom tables can be created viewed or manipulated according to the users requirement. Maintenance dialogs are used mainly in customizing. They can also be used to maintain customer table or view contents TMGs can be created for custom tables for which Table Maintenance is allowed. It also provides options to check or modify the data which is entered by the help of events. The table TOTAL contains all data which are read, changed and created in the extended table maintenance stored in ascending order. The table EXTRACT is the data display work table. At runtime it contains all the data which was selected from the table TOTAL in the extended table maintenance. The selection is made by the user by calling functions such as *All entries read*, *All selected entries*, or *All changed entries*. It's sorted in ascending order. This document throws light on the events which are provided to implement additional logic to the existing one in the standard and their implementation.

Table Maintenance Events

As mentioned before TMG provide various Table maintenance events which can be used to process and update data provided. They are:

Event 01: Before saving the data in the database

This event is triggered before the data is saved via the table maintenance. This event can prove to be useful for updating hidden entries .Updating indicator flags to hidden tables on database modification.

Event 02: After saving the data in the database

This event is triggered as soon as data is saved into the database. It can be used to maintain hidden tables and help with application logging.

Event 03: Before deleting the display data

The event occurs when data is deleted using the delete functionality in the table maintenance. It is possible for the user to check whether a particular record can be deleted based on a particular check.

Event 04: After deleting the display data

After deletion of selected documents from the table this event is executed. The user can make use of this event to fill hidden fields, update internal data and to generate any messages.

Event 05: When creating a new entry

When a new entry is created or copied this event provides the user an option to perform checks, update hidden fields and update user internal tables.

Event 06: After the 'Get original' function.

When 'Get original' function is executed this event can't be triggered to update user internal tables also initialize or fill hidden fields.

Event 07: Before correcting the contents of a selected field

In order to check whether selected documents can be changed when change fields function is called this event can be used. It can be also used to fill hidden table and update user internal tables.

Event 08: After correcting the contents of a selected field

This event can be used to fill hidden fields and update user internal tables .it has to be noted that one cannot tell which entries were handled. The selected entries must be captured into an internal table at Event07.

Event 09: After 'Get original' of an entry

This event can be used to fill, initialize hidden fields and fill user internal tables once "Get Original" function is triggered for a selected entry. Only one entry is handled at a time and is found in the header line of TOTAL and EXTRACT.

Event 10: After creating the change request header entries (E071)

The event is triggered when standard change request header entry has been created .It can also be used t o enter the header entries of other tables which are not in the view structure definition and are not assigned text tables, in the change request. These tables do not figure in the maintenance object description object list

Event 11: After changing a key entry in the change request (E071K)

This event is activated when an entry is made in the change request key list by choosing 'Save'. It can also be used to enter tables which were maintained in the background, and are not in the view structure definition and are not assigned text tables, in the key list.

Event 12: After changing a key entry in the change request (E071K)

Once changes have been made to the change request and save action is performed this event is triggered.

Event 13: End of processing (leaving the main function module)

This event can be used to reset user flags, initialize user fields, internal tables and the function group local memory ,Perform activities determined by user settings .Its triggered on leaving the maintenance dialog.

Event 14: After locking/unlocking in the main function module

This event can be activated when the processing mode changes from Display to Change or vice-versa and after the maintenance dialog tables have been locked according to Data Dictionary information. It can be used for locking/unlocking other hidden tables.

Event 15: Before retrieving deleted entries

Deleted entries can be retrieved using the 'Retrieve' function. The event can be used to perform checks like whether retrieval is restricted by any other dependency.

Event 16: After retrieving deleted entries

Once the 'Retrieve' function has been triggered it can be used to fill hidden fields, update user specific internal tables and message generation.

Event 17: Before printing entries

The use of this event is not recommended.

This event occurs before the 'Print' function. It can be used to change the format of table/view fields for printing, include hidden fields for printing; Prevent printing fields which are not flagged as hidden in the Data Dictionary.

Event 18: After the data change check

This event can be used to perform security relevant checks in user internal tables and TOTAL internal table etc.

Event 19: After initialization of global variables, field symbols, etc.

This event can be triggered when a table maintenance dialog is called. The function group local memory is initialized at each entry. It can also be used to perform global data n internal table initialization, change title bar in maintenance screen, deactivate interface functions in the internal table EXCL_CUA_FUNCT

Event 20: After input in the date sub screen (time-dep. tables/views)

It is associated with time dependant tables/views .Its triggered when input is made. It can be used to perform actions like filling hidden fields or update user internal tables. The routines here must have two parameters specifying the start and end date respectively.

Event 21: Fill hidden fields

This event can be used to fill the fields which are deemed as hidden in the data dictionary.

Event 22: Go to long text maintenance for other languages

The user can perform long text maintenance from the translation. Changes made here affect the maintenance screen. When the event is assigned to the maintenance dialog, a pushbutton with the 'Display/change others' icon appears after each entry in the "Texts in other languages" maintenance screen. The event occurs when the user activates the pushbutton to go to the long text of the current entry. The long text is processed (display, database accesses, etc.) as per the user routine.

Event 23: Before the address maintenance screen call

Standard screen maintenance can be modified to meet user requirements but using this event. It's triggered before the address maintenance screen is displayed, i.e. before the ADDR_DIALOG_PREPARE function module call.

Event 25: At the start of the maintenance dialog.

Additional authorization checks can be triggered by this event.

Event 26: Before displayed data is output in a list

The function module REUSE_ALV_LIST_DISPLAY outputs the list. It can also be used to Change table/view field display format, Print hidden fields, prevent non hidden fields from being printed etc.

Event AA: Instead of the standard data read routine

This event can be used to trigger an alternate logic to the standard one when data is read in. It can be used to read additional data, manipulate read data, and provide additional checks on top of the existing standard logic.

Event AB: Instead of the standard database changes routine

This can be used to run an alternate logic when 'Save' function is executed. E.g.: update hidden fields.

Event AC: Instead of the standard 'Get original' routine

An alternate logic can be triggered when 'Get Original' function is activated. It can be used to perform tasks like processing fields which are not in the view/table, but are displayed in the overview or detail screens.

Event AD: Instead of the standard RO field read routine

It's relevant in the case of views which contain fields from referred tables. It can be used when the contents of these fields ought to be changed.

Event AE: Instead of standard positioning coding

This event Executed when 'Position' is triggered. Can be used to implement alternate logic for scenarios like when EXTRACT table is not sorted.

Event AF: When reading texts in other languages

When 'Translation' functions is executed. A different logic from the standard method where texts are made available for translation to selected languages can be implemented, Can be applied to change read date, to follow a different read procedure. This event is also triggered if time-dependent entries are restricted by time-dependent texts. In this case, the language-dependent texts must be read in all languages.

Event AG: Instead of 'Get original' for texts in other languages

Get Original gets all language dependent texts from database. With this event an alternate logic can be implemented in the standard .like processing texts which aren't part of view but are displayed on the screen.

Event AH: Instead of DB change for texts in other languages

This event occurs at the 'Save' function. A different logic from the standard can run, for example to also change hidden tables in the database.

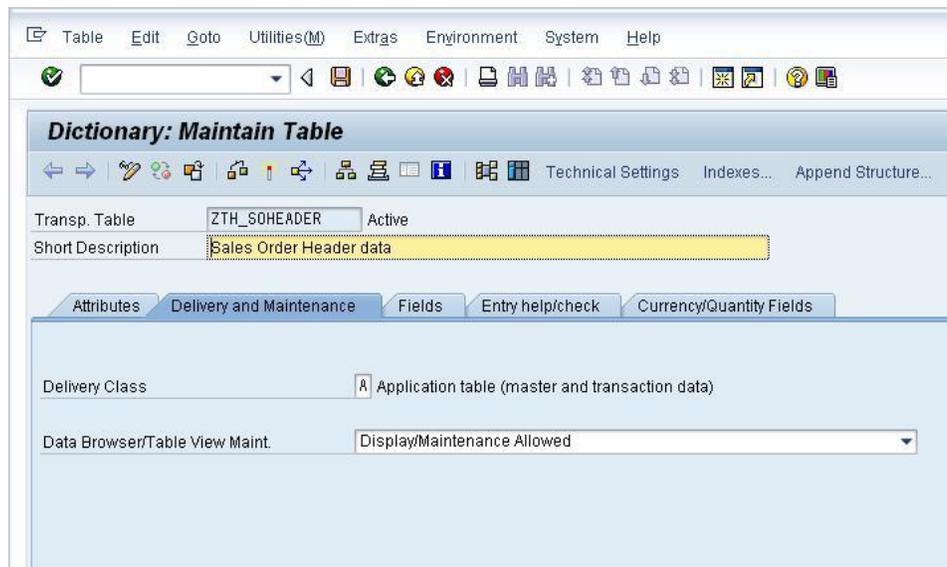
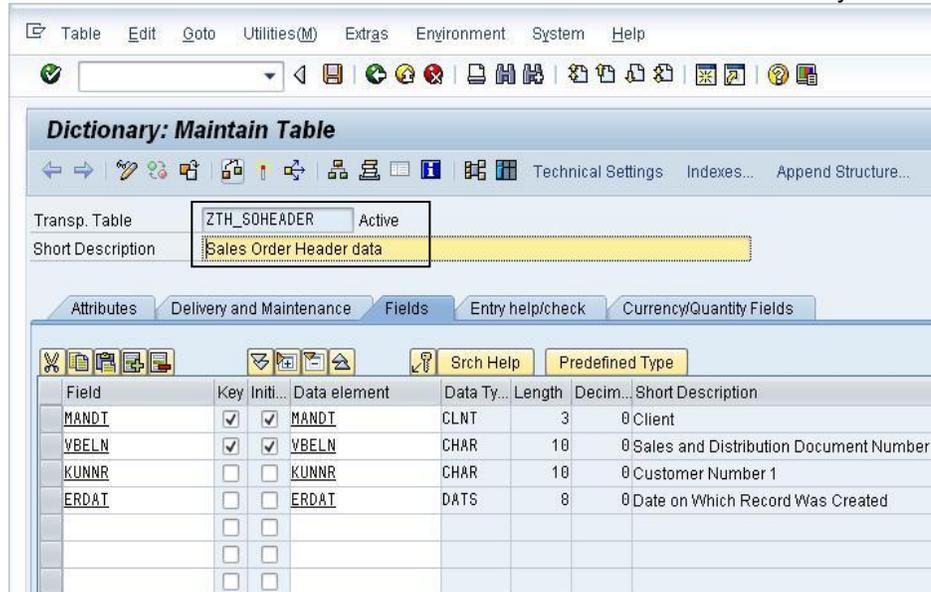
Event ST: GUI menu main program name

This event enables changing function group interface for the table/view maintenance dialog at runtime and change the standard interface.

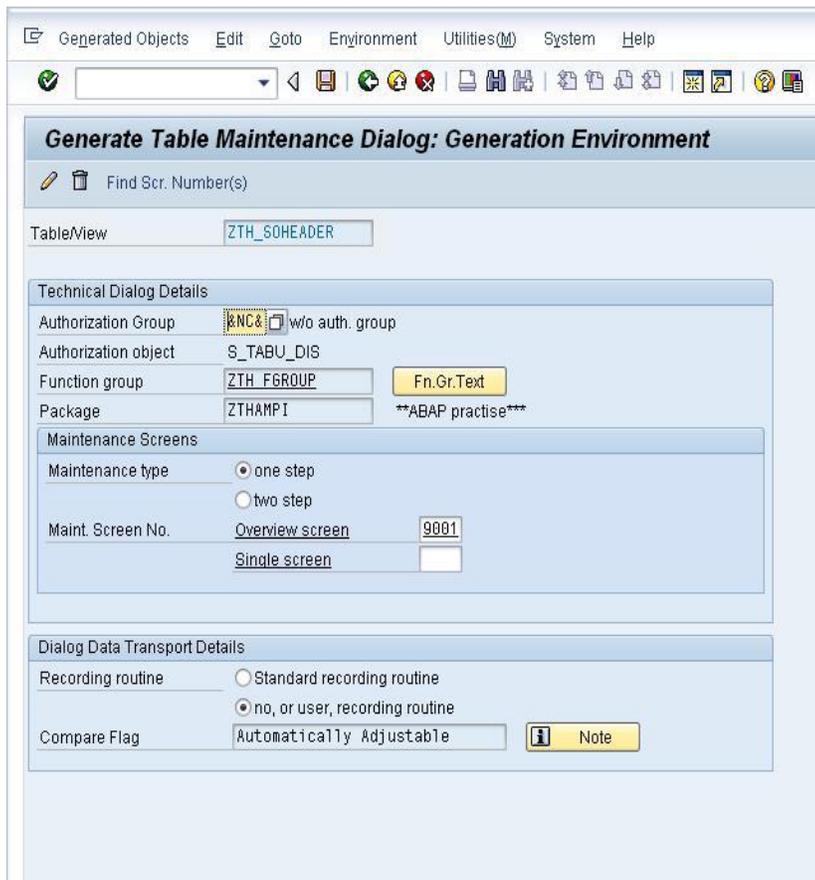
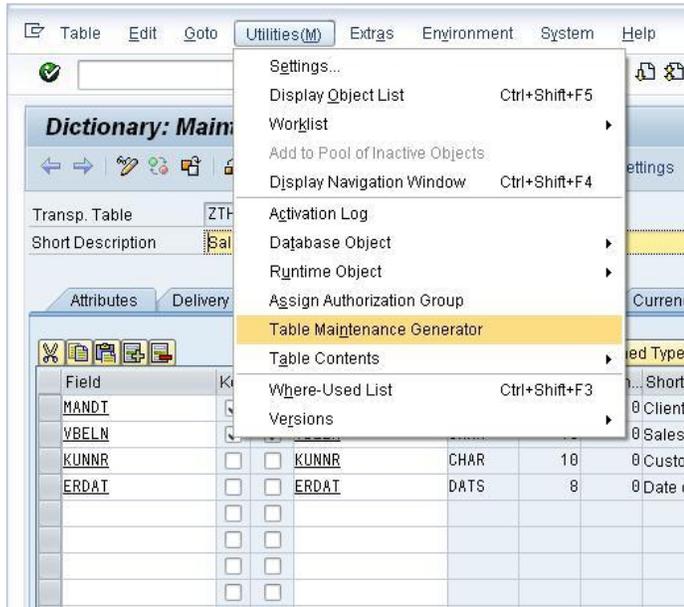
Here the function group SAPL<...> is provided instead of the custom routine as in the case of other events.

Step by Step Process to Implement Events

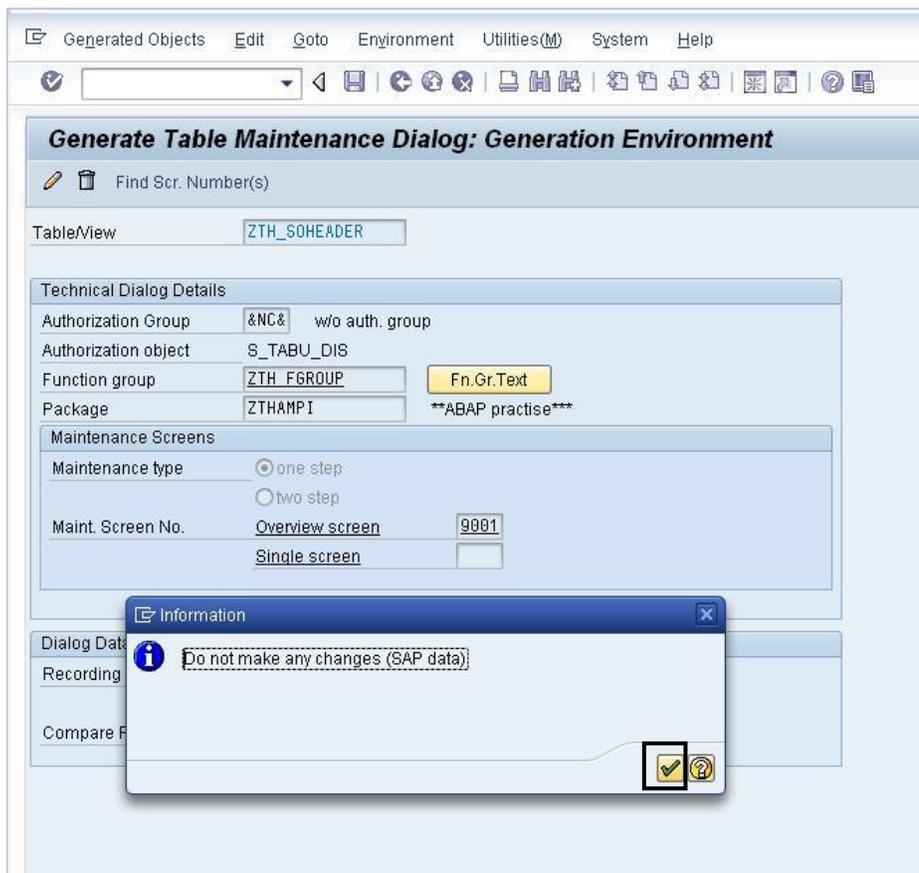
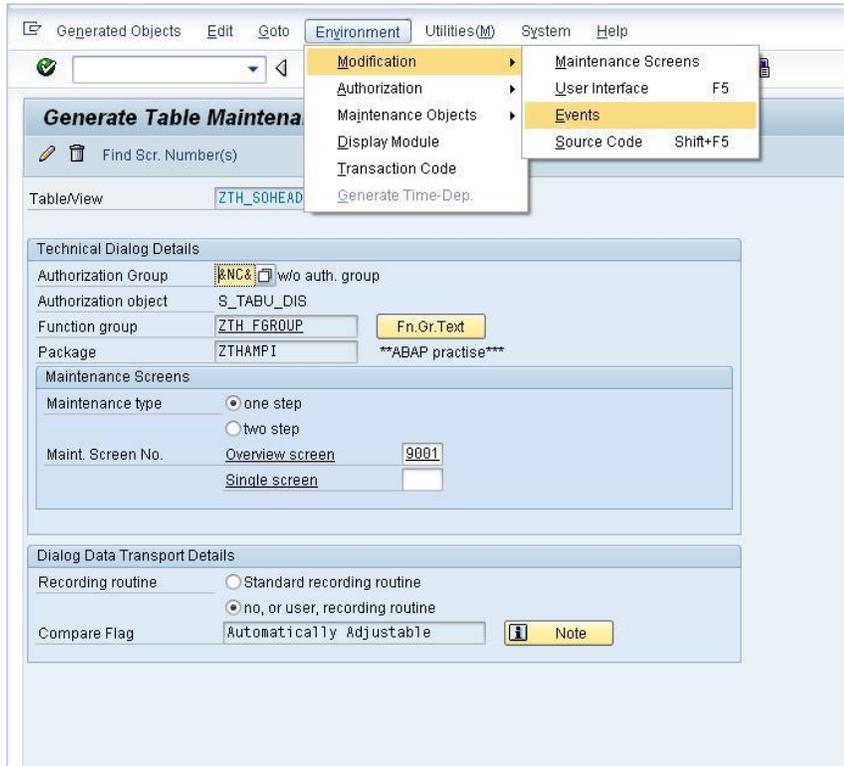
From SE11 transaction create the table. Enable table maintenance from the 'Delivery and Maintenance' tab.



After activating the table goto 'Utilities' and navigate to Table Maintenance Generator and creater table maintenance.



Once Table maintenance is created navigate to Environment → Modifications → Events.
 On selecting an option a pop up screen will be displayed select OK.



From the screen which is displayed. Use the search help to select the event which ought to be triggered.

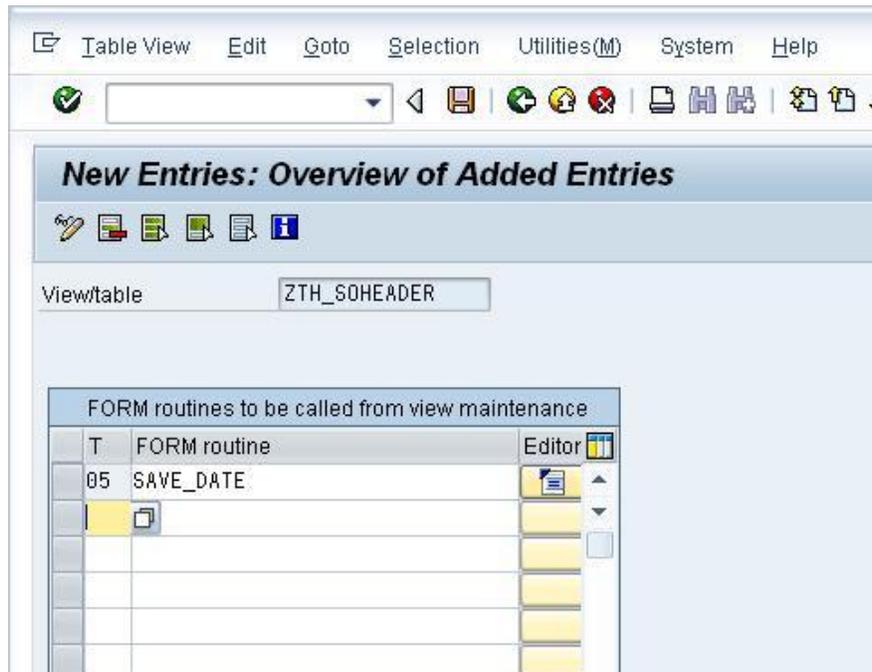
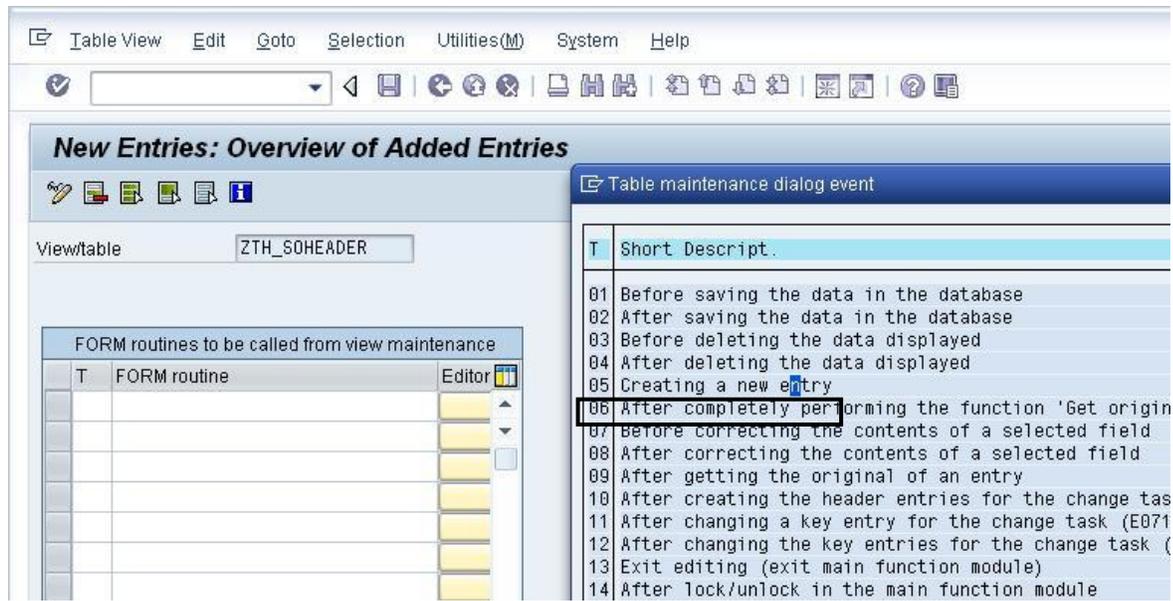
The screenshot shows the SAP Table Maintenance dialog event selection screen. The main window is titled "New Entries: Overview of Added Entries" and displays a table with columns "T", "FORM routine", and "Editor". The table is currently empty. A dialog box titled "Table maintenance dialog event" is open, showing a list of events. The events are listed as follows:

T	Short Descript.
01	Before saving the data in the database
02	After saving the data in the database
03	Before deleting the data displayed
04	After deleting the data displayed
05	Creating a new entry
06	After completely performing the function 'Get original'
07	Before correcting the contents of a selected field
08	After correcting the contents of a selected field
09	After getting the original of an entry
10	After creating the header entries for the change task (E071)
11	After changing a key entry for the change task (E071K)
12	After changing the key entries for the change task (E071K)
13	Exit editing (exit main function module)
14	After lock/unlock in the main function module
15	Before retrieving deleted entries
16	After retrieving deleted entries
17	Do not use. Before print: Event 26
18	After checking whether the data has changed
19	After initializing global variables, field symbols, etc.
20	after input in date subscreen (time-dep. tab./views)
21	Fill hidden fields
22	Go to long text maintenance for other languages
23	Before calling address maintenance screen
24	After restricting an entry (time-dep. tab./views)
25	Individual authorization checks
26	Before creating a list
27	After creation or copying a GUID (not a key field)
28	After entering a date restriction for time-dep. views
AA	Instead of the standard data read routine
AB	Instead of the standard database change routine
AC	Instead of the standard 'Get original' routine
AD	Instead of the standard R0 field read routine
AE	Instead of standard positioning coding

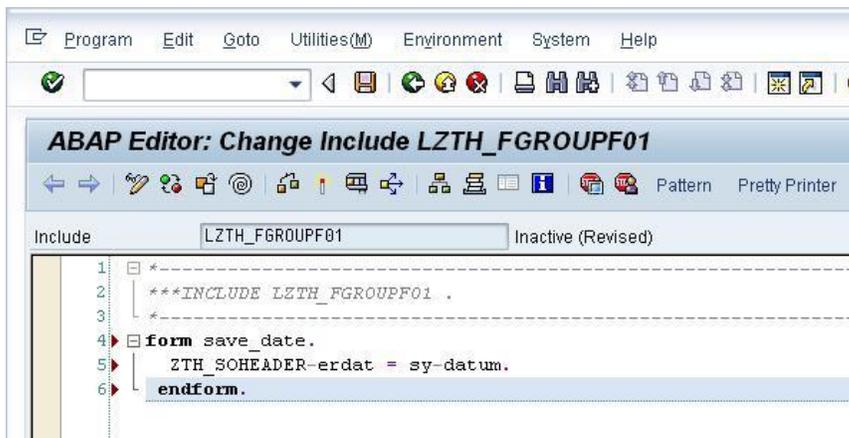
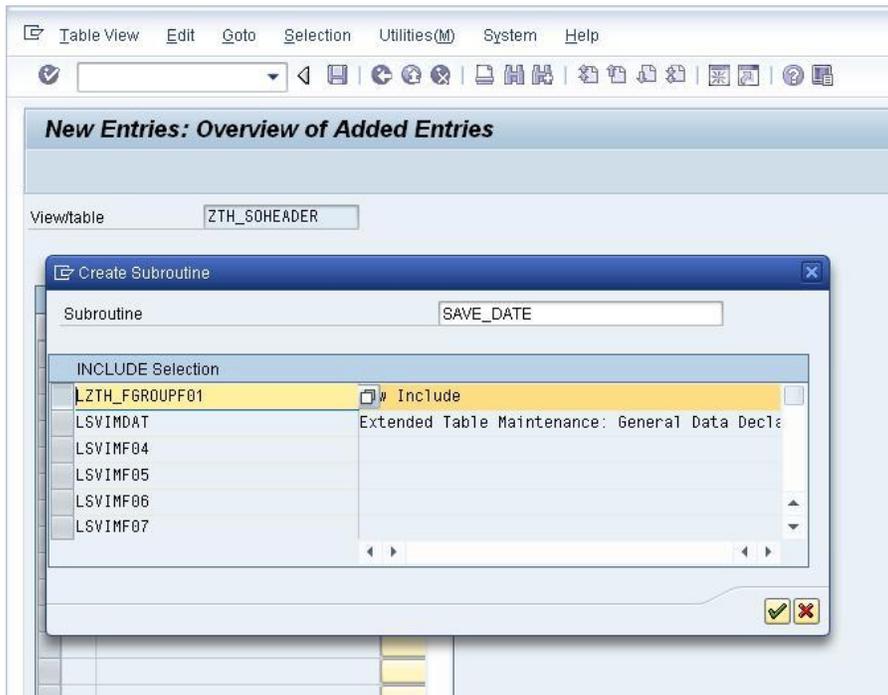
A search help window is also visible, showing a list of events:

AF	Instead of reading texts in other languages
AG	Instead of 'Get original' for texts in other languages
AH	Instead of DB change for texts in other languages
ST	GUI menu main program name
AI	Internal use only

Select the appropriate event and provide the subroutine name which has to be run when the event is triggered. (Here in the example its Event 5 and routine SAVE_DATE)



Once it's completed press the editor button. A prompt is generated indicating creation of a new include to house the routine. On selecting 'OK', an include is created where the source code can be typed in. (Here the code written is to auto populate the ERDAT field with current date every time data is saved.)



Related Content

[Extended Table Maintenance Events](#)

[Table Maintenance Generator](#)

[How to Implement Events in Table Maintenance](#)

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.