

Applies to:

SAP ERP 6.0 – SAP NetWeaver 7.00.

Summary

Text maintenance is an important feature in many different transactions of SAP, which lacks one main feature compared to field related information. The audit log functionality, which provides a detailed history for all field related changes, if required. This feature clearly does not exist for any long text, even so the user would like to see this in some cases. Text maintenance in form of notes allows the user to create new text, which gets stored together with user name, and time stamp, but never allows deleting the old text. The result is a journal of all text from different users over time. The following paper shows an approach to solve this with a new created text editor interface inside the text maintenance possibilities of SAP text editing.

Author: Rene Turnheim

Company: Instrumentation Laboratory

Created on: 22 March 2008

Author Bio



My name is Rene Turnheim and I work as Manager of Software Development at [Instrumentation Laboratory](#), a medical device company. I joined the company in 1999, when I transferred from Austria to the US. My department is responsible for the SAP support and development for the US plants in Lexington, MA and Orangeburg, NY.

Table of Contents

Introduction	3
Text maintenance	3
Text Editor	3
Text Editing	4
Implementation	5
Text Format	5
Conclusion	6
Related Content	6
Disclaimer and Liability Notice	7

Introduction

At several occasions our SAP users at [Instrumentation Laboratory](#) (IL) have requested the possibility to record text changes in form of a journal. The standard implementation of SAP text editing, does not provide this feature. The extensive audit log or change document possibilities in SAP on a table field by table field basis, guided our user to the idea having also this kind of feature or a similar solution during text editing. As IL is a FDA ([Food and Drug Administration](#)) regulated company also this has put pressure to find an adequate solution in some cases.

The functionality as Microsoft Word is providing it with the track changing solution is clearly a high sophisticated solution and is in most cases far beyond the basic needs our users have expressed for SAP text maintenance. The user at IL has been asking for the following functionality as a minimum requirement:

- Record user, date and time stamp of each time new text is added to an existing text
- Disallow the change of already existing text
- Display the history of written text by different users in an easy accessible way during adding of new text.

Text maintenance

The solution for a notes-logic is very depended on the type of editor used for creating the text. The text editor solution has been changed by SAP over the years, therefore I want to make first a short review of the possibilities, before we look at the proposed solution.

Text Editor

A main factor for the text editor used is clearly the SAP environment, used by the user and the development environment used. IL's users have always accessed SAP with the SAP GUI. All PC's have Microsoft Office installed. All development for our users is based on ABAP. Historically at IL we have used the class CL_GUI_TEXT_EDIT as a solution to provide in our customized transaction the capability for text editing. Last year we have successfully upgraded to SAP ERP 6.0 with SAP NetWeaver 7.0 and converted to UNICODE. This change has forced us to improve our solution and standardize the interface.

Taking all this into account going forward the following text editor solutions are available to us as of today:

- Traditional line editor (SAP solution without MS Office)
- Microsoft Word (SAP Solution with MS Office)
- CL_GUI_TEXTEDIT object
- BTF Editor (enhanced text editing possibilities)

As the traditional line editor is no longer accepted by our users, we have implemented three different solutions for our notes-logic based on the following editor objects of SAP:

- CL_WORD_SAPSCRIPT_EDITOR for Microsoft Word
- CL_GUI_TEXTEDIT
- IF_BTF_EDITOR (in testing phase as an enhanced solution)

Text Editing

As an example how this solution works, I want to show you an implementation of the notes logic for the customer master. The Customer internal notes in the customer master (General data) have been changed to the notes logic with journal. The following changes in SE75 have been done:

Change ID

Text ID: 0018

Meaning: Customer internal text

Text Name

Display in Title:

Structure of key:

Text format: ZGTENO

INCLUDE ID:

Note: A detail explanation behind the idea of the Text format (ZGTENO) logic is explained later. This shows only the minimal changes needed, to get the new functionality working.

The result in XD02 (Customer maintenance, General data internal text) looks like this:

Change Customer internal text: 0000332029 Language EN

03/20/2008 22:43:39 John McCusker (JMCCUSKER)
PER RON RICE DO NOT SET UP THIS ONTARIO CUST W/7000000000 FOR A JURIS
CODE - THEY DID NOT SUPPLY TAX EXEMPT INFO - HE SAID HE WILL CHANGE THE
CODE LATER IF THEY DO.

Ln 1, Co 1 Ln 1 - Ln 1 of 1 lines

Note: This shows a reduced screen picture. The text is maintained in full screen modus: Upper half for new text and lower half for notes history.

As you can see, every change gets recorded with user name, date and time stamp. This logic applies to any of the three text editor solutions I have developed.

Implementation

My goal to implement the proposed solution was based on the desire to blend into the existing solution of SAP for text editing. Therefore the solution tries to avoid any modification to existing SAP code and is fully based on a simple customization and a set of functions and reports following the logic of SAP text editing.

Text Format

SAP provides the possibility with transaction SE75 to maintain either at *text object* level or at *text id* level a text format. The text format allows to influence, which editor should be used. This text format gets controlled with table TTXFORMAT. I have created the following entries in table TTXFORMAT:

Txt format	DTP format	SUPPORTED
ZBTFNO	ZXXXNX	X
ZBTFTX	ZXXXTX	X
ZGTENO	ZXXXNX	X
ZGTETX	ZXXXTX	X
ZWRDNO	ZXXXNX	X
ZWRDTX	ZXXXTX	X

Note: This shows the allowed values for transaction SE75. For each editor type (BTF, GTE, WRD) two entries allow normal text editing (TX) or notes logic (NO). The second column is the base for the function module which gets called by the function module EDIT_TEXT

As soon as either a *text object* or a *text id* is maintained with a text format from table TTXFORMAT, the standard text editing function module EDIT_TEXT will automatically call the following function module:

EDIT_TEXT_FORMAT_XXXXXX

The XXXXXX gets substituted by the entry of table TTXFORMAT of field DTP format. I have created two functions:

- EDIT_TEXT_FORMAT_ZXXXTX for regular text editing (to make use of TEXTEDIT GUI control, BTF editor, ...)
- EDIT_TEXT_FORMAT_ZXXXNO for notes-logic text editing

Each of the two function modules supports three different text editor types:

- GTE for the CL_GUI_TEXTEDIT GUI object
- WRD for the CL_WORD_SAPSCRIPT_EDITOR GUI object
- BTF for the IF_BTF_EDITOR interface

The result of this definitions is, now any text, (there are limitations, if the transactions allows the first one or two lines to be changed through screen fields) which gets maintain in SE75 with one of the six different text types, will force the call of one of the two functions mentioned above for any new text entered. The existing text will get maintained unchanged (standard logic), as the text format value gets copied into table STXH during text creation.

Conclusion

This is a short overview how I have implemented the notes logic into the text maintenance function of SAP. I am very interested to get feedback, how you have approached this problem (if) and what you think about this solution. Feel free to contact me for any further information or questions.

Related Content

[BTF editor in SAPGUI](#)

[BTF in the real world](#)

[The Standard Text Editor \(OO ABAP CFW class\)](#)

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.