How to Modify, Create and Delete Table Entries from SE16

Applies to

This article applies to all SAP ABAP based products; however the examples and screen shots are derived from ECC 6.0 system. For more information, visit the ABAP homepage.

Summary

This article explains how to modify, create and delete table entries when a table is not maintained through transaction SM30. In this case you can use transaction SE16 and the Debugger tool.

Author: David García Ortega

Company: Accenture

Created on: 21st January 2011

Author Bio

David is an SAP consultant with 3+ years experience working for Accenture. He has developing experience in ABAP and Business Objects.

David also has functional knowledge of FI, MM, SD, PM and IS-U modules with expertise in BPEM (Business Process Exception Management) configuration and SAP Data Archiving full life cycle implementation.

Table of Contents

Introduction 3

How to Modify Table Entries using Transaction SE16 4

- Scenario 4
- Approach 4
- Steps in SE16 4

How to Create Table Entries using Transaction SE16 7

- Scenario 7
- Approach 7
- Steps in SE16 7

How to Delete Table Entries using Transaction SE16 8

- Scenario 8
- Approach 8
- Steps in SE16 8

Related Content 11

Disclaimer and Liability Notice 12

Introduction

Have you ever been in a situation where you need a configuration flag to be set (or unset) but you don't know the transaction to configure it? Or in the middle of a test and you need some more data with particular conditions and the process to create this data is very long and tedious?

If you answer yes to any (or both) of these questions then this How-to article will probably turn out to be useful for you. If the tables where this data is stored are not maintained through SM30, this article will show you a series of techniques to modify, create and delete table entries using transaction SE16 (SE11 can also be used).

In order to use these techniques, assumption is made that debugging is enabled on your system and your SAP user has the right authorizations to change variables during debugging.

Note: Please bear in mind that these techniques are quite powerful in the sense that they allow you to change pretty much any configuration or data on your SAP system. You should always make sure that your data integrity is not affected when modifying, creating or deleting entries from tables and you should never use these methods in a Production system. These techniques can save you a lot of time but use them carefully.

How to Modify Table Entries using Transaction SE16

This section describes the steps to modify an entry from a table. An example scenario and approach are also provided in order to illustrate a situation where you might need to do this.

Scenario

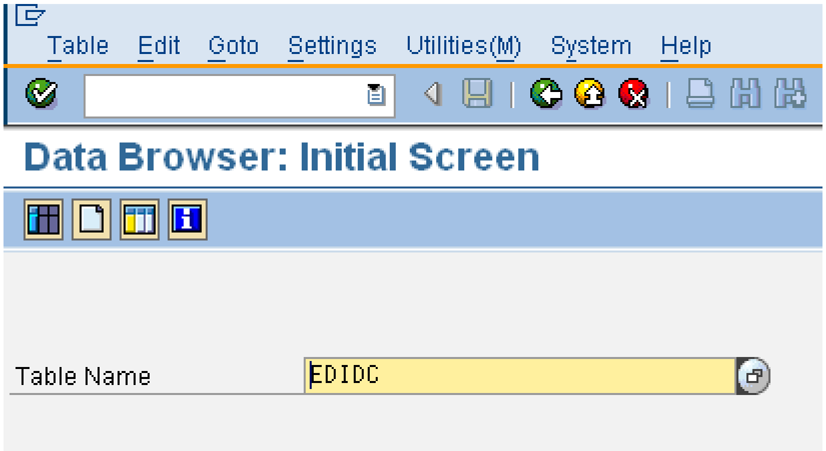
You're currently testing the archiving functionality for IDocs but there is no IDoc that complies with the residence period criteria (IDoc has been created X years ago).

Approach

In this situation, an option is to edit the creation date for an IDoc in order to make it X years old and hence eligible to be archived (as it will be "old enough").

Steps in SE16

1. Go to transaction SE16 (SE11 is also valid).
2. In the table name field enter the table (in this example is EDIDC).



Data Browser: Initial Screen

Table Name EDIDC

3. Select an IDoc that you'd like to modify and double-click on it to access its Display screen. Below you can see an image where the field that we want to change is highlighted.

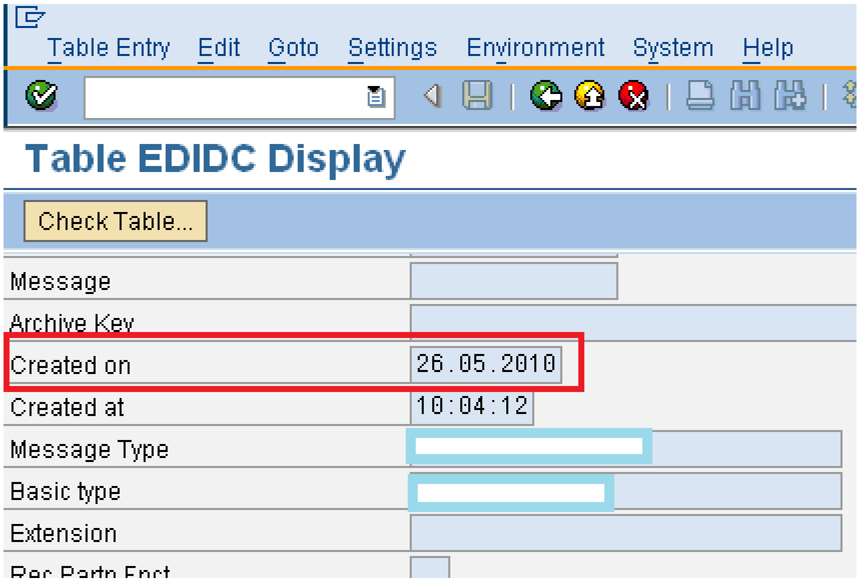
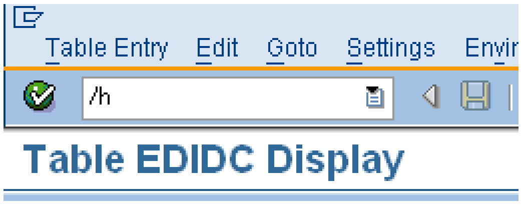


Table EDIDC Display

Message
Archive Key
Created on 26.05.2010
Created at 10:04:12
Message Type
Basic type
Extension
Rec Partn Fnct

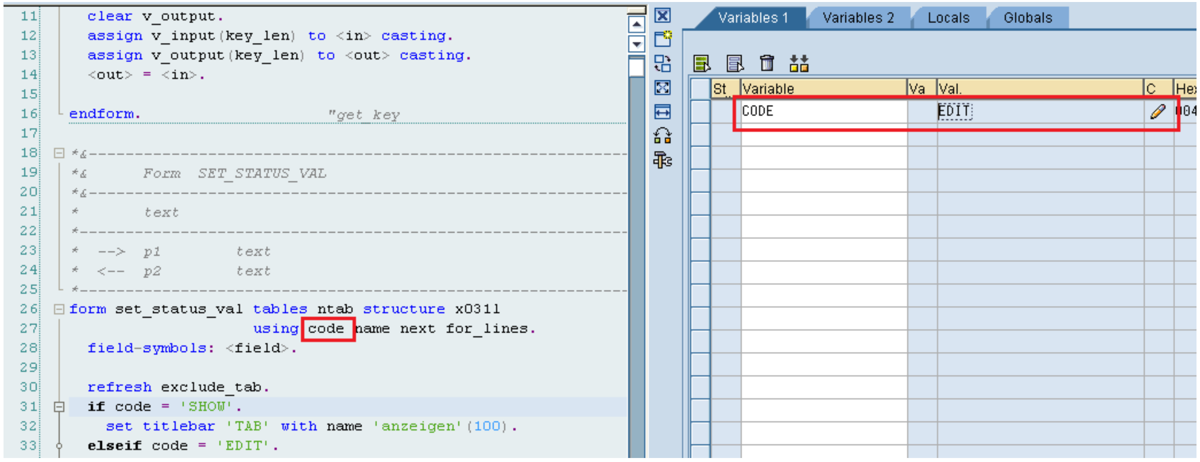
4. Write /h in the Command field and press return on your keyboard.



/h

Table EDIDC Display

5. Press return again to go into debug mode, select the variable with id "code" and modify its value to be "EDIT".



```
  clear v_output.
  assign v_input(key_len) to <in> casting.
  assign v_output(key_len) to <out> casting.
  <out> = <in>.

endform.                    "get_key

*&---------------------------------------------------------------------
*&      Form  SET_STATUS_VAL
*&---------------------------------------------------------------------
*       text
*----------------------------------------------------------------------
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------
form set_status_val tables ntab structure x031l
                    using code name next for_lines.
  field-symbols: <field>.

  refresh exclude_tab.
  if code = 'SHOW'.
    set titlebar 'TAB' with name 'anzeigen'(100).
  elseif code = 'EDIT'.
```

St	Variable	Va	Val.	C	Hex
	CODE		EDIT		004

6. Press F8 to complete execution and modify the contents of the selected entry.

7. Edit value for the fields that you want to modify, in this case just "Created on" and click on Save.

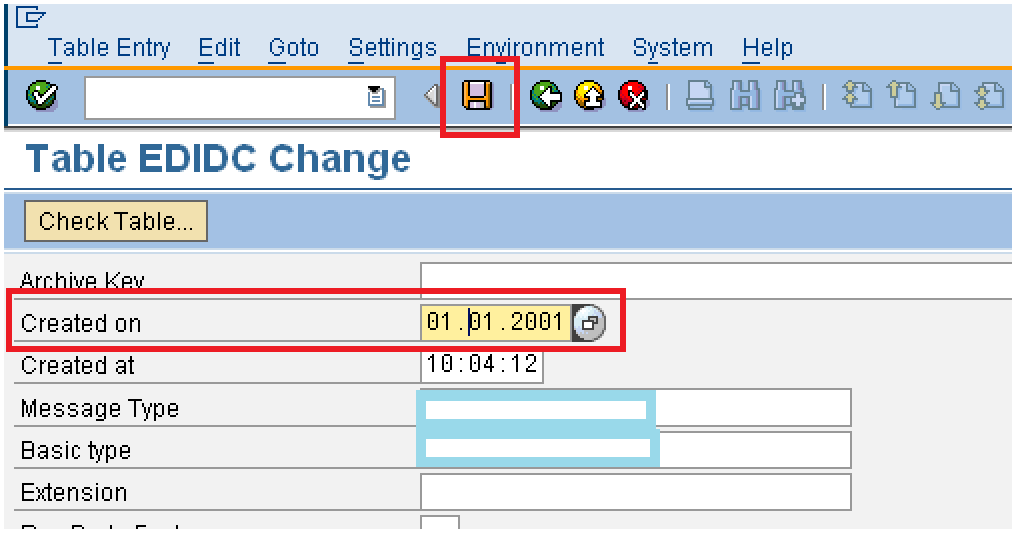


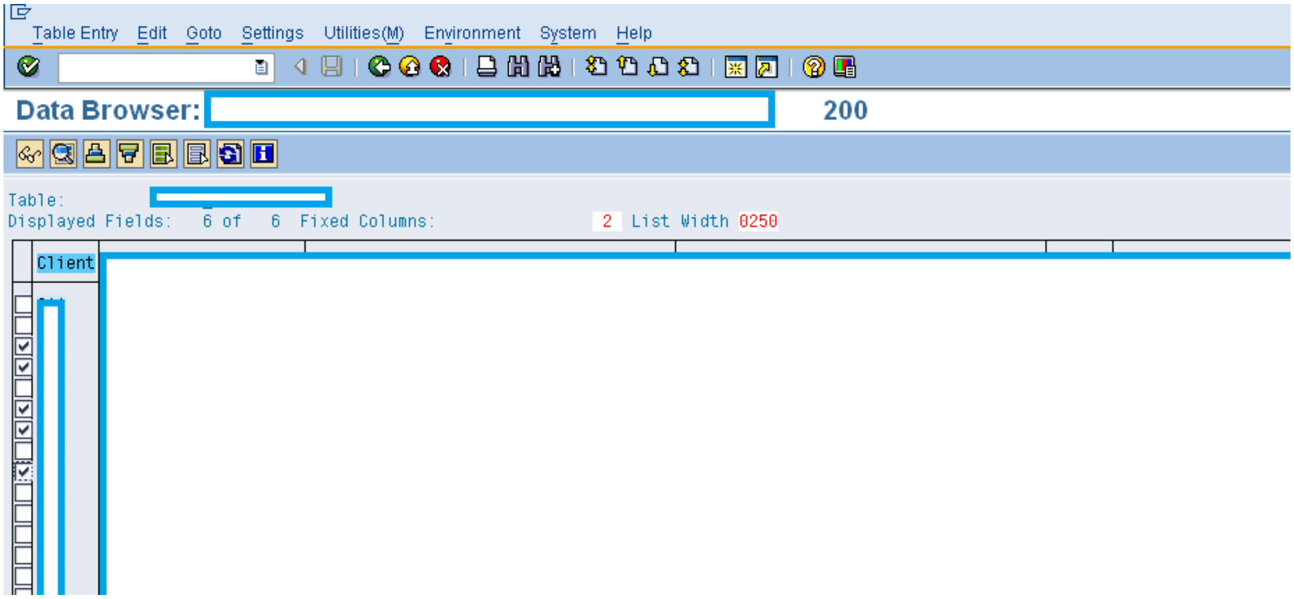
Table EDIDC Change

Archive Key	
Created on	01.01.2001
Created at	10:04:12
Message Type	
Basic type	
Extension	

8. You have successfully modified a field in a table (in this example EDIDC).

Tip: You can select several entries on the Data Browser screen and modify them with only one execution of the debugger tool to modify the value of "code" variable.

Selecting several entries from the Data Browser screen:



Data Browser: 200

Table:
Displayed Fields: 6 of 6 Fixed Columns: 2 List Width 0250

Client

After modifying the values of a single entry, click on “Save” and then “Next entry”:

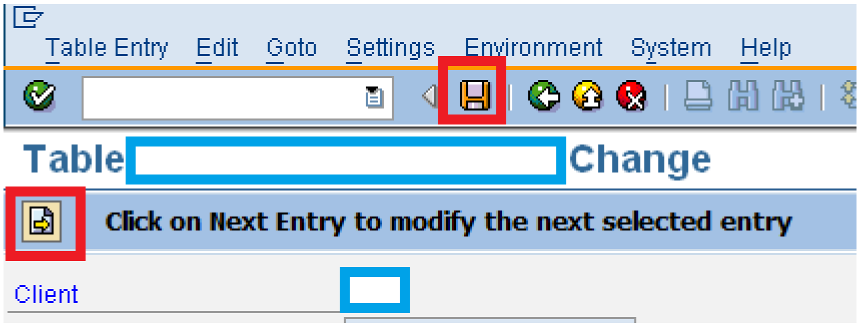


Table Change

Click on Next Entry to modify the next selected entry

Client

How to Create Table Entries using Transaction SE16

This section describes the steps to create a new entry in a table. An example scenario and approach are also provided in order to illustrate a situation where you might need to do this.

Scenario

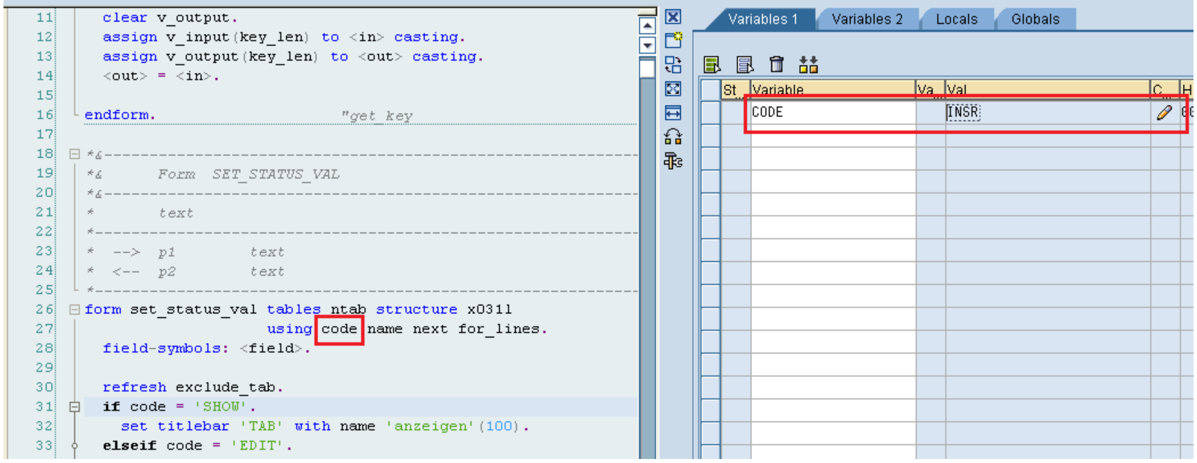
You are testing a process that requires meter reads with actual consumption over 500. The actions to create new meter reads require a lot of time but you know the custom table where meter reads are stored.

Approach

In this situation, an option is to create a new entry in this custom (Z) table with value for actual consumption over 500.

Steps in SE16

1. Go to transaction SE16 (SE11 is also valid).
2. In the table name field enter the table (in this example your Z custom table).
3. Select any entry (ideally you'd like to select an entry whose content you can reuse for the new one) and double click on it to access its Display screen.
4. Write /h in the Command field and press return on your keyboard.
5. Press return again to go into debug mode, select the variable with id "code" and modify its value to be "INSR".



```
    clear v_output.
    assign v_input(key_len) to <in> casting.
    assign v_output(key_len) to <out> casting.
    <out> = <in>.

  endform.                              "get_key

*&---------------------------------------------------------------------
*&      Form  SET_STATUS_VAL
*&---------------------------------------------------------------------
*       text
*----------------------------------------------------------------------
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------
form set_status_val tables ntab structure x031l
                    using code name next for_lines.
  field-symbols: <field>.

  refresh exclude_tab.
  if code = 'SHOW'.
    set titlebar 'TAB' with name 'anzeigen'(100).
  elseif code = 'EDIT'.
```

St	Variable	Va	Val	C	H
	CODE		INSR	✎	

6. Press F8 to complete execution and create a new entry based on the values of the entry that you were displaying.
7. You now need to be careful and make sure that you're respecting integrity rules for the table (i.e. not duplication of primary or unique keys, in this example "Int. MR doc. ID"). Provide a new value for these fields and also make actual consumption bigger than 500 (which is the purpose of this example). Once you have completed your changes click on save.

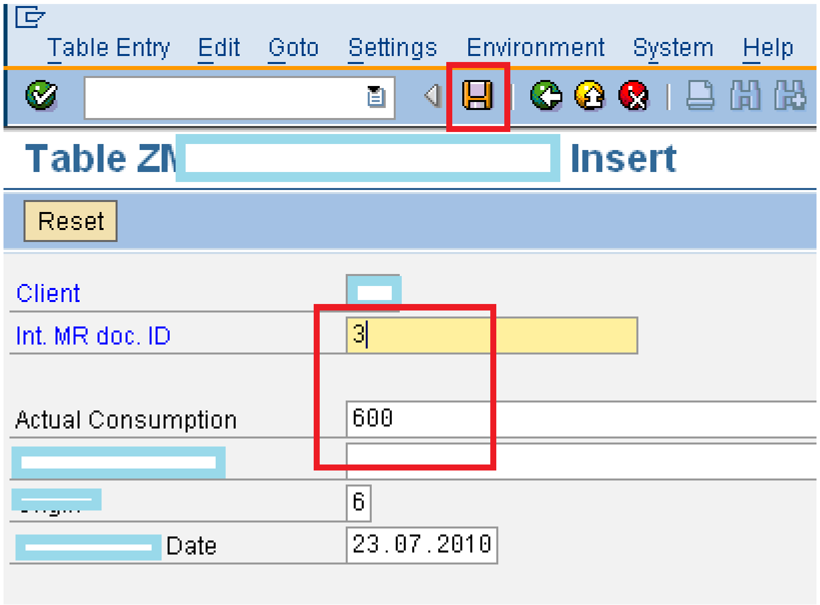


Table ZM[redacted] Insert

Client [redacted]

Int. MR doc. ID 3

Actual Consumption 600

[redacted]

[redacted] 6

[redacted] Date 23.07.2010

8. You have successfully created a new entry on this custom table.

Note: Using these steps you can only create one entry each time.

How to Delete Table Entries using Transaction SE16

This section describes the steps to delete an entry from a table. An example scenario and approach are also provided in order to illustrate a situation where you might need to do this.

Scenario

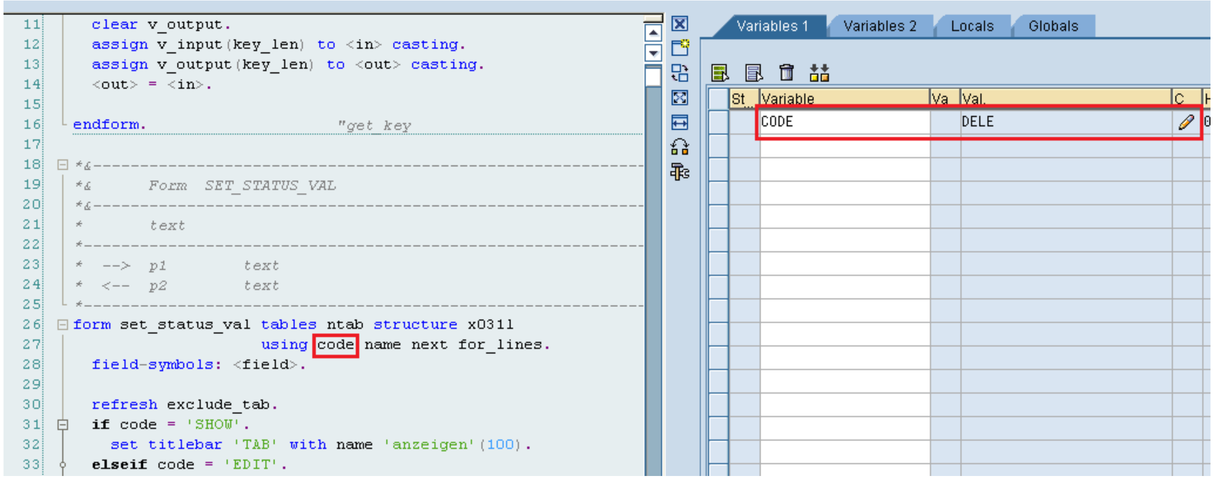
A process incorrectly configured has created some wrong data and you'd like to delete it as it'll probably affect your tests.

Approach

In this situation, an option is to delete this data directly from the table where it's stored (in this example a custom table)

Steps in SE16

1. Go to transaction SE16 (SE11 is also valid).
2. In the table name field enter the table (in this example your Z custom table).
3. Select the entry containing the wrong data and double click on it to access its Display screen.
4. Write /h in the Command field and press return on your keyboard.
5. Press return again to go into debug mode, select the variable with id "code" and modify its value to be "DELE".



```
    clear v_output.
    assign v_input(key_len) to <in> casting.
    assign v_output(key_len) to <out> casting.
    <out> = <in>.

  endform.                    "get_key

*&---------------------------------------------------------------------
*&      Form  SET_STATUS_VAL
*&---------------------------------------------------------------------
*       text
*----------------------------------------------------------------------
*  -->  p1        text
*  <--  p2        text
*----------------------------------------------------------------------
form set_status_val tables ntab structure x0311
                    using code name next for_lines.
  field-symbols: <field>.

  refresh exclude_tab.
  if code = 'SHOW'.
    set titlebar 'TAB' with name 'anzeigen'(100).
  elseif code = 'EDIT'.
```

St	Variable	Va	Val.	C
	CODE		DELE	✎

6. Press F8 to complete execution and proceed to delete the entry that you selected.
7. Confirm that you want to delete this entry by clicking on “Delete Entry”.

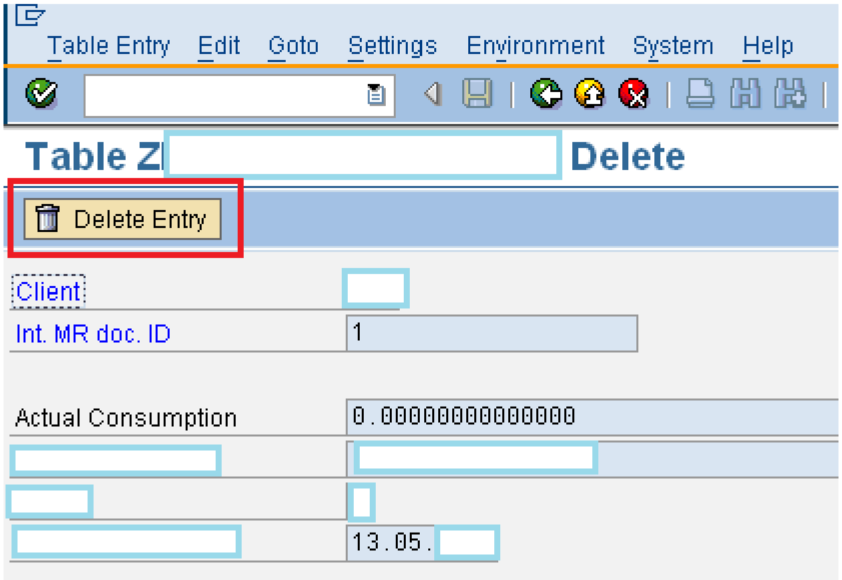


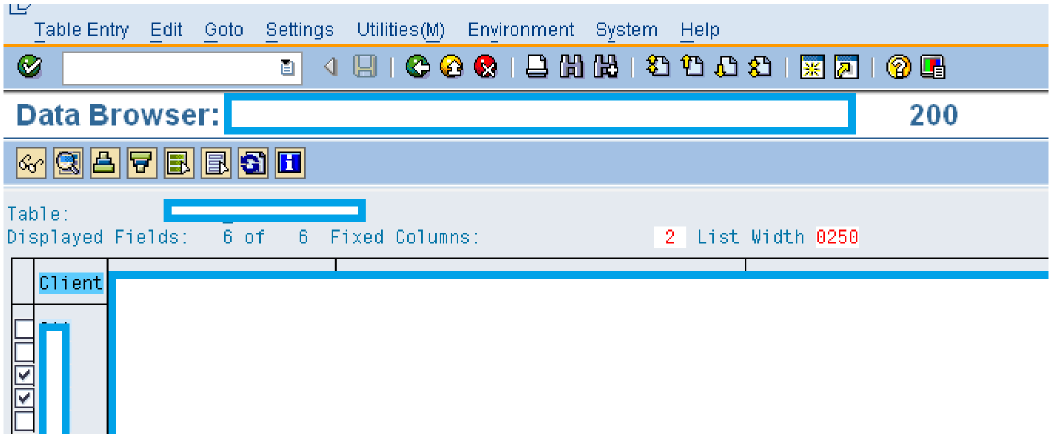
Table Z[redacted] Delete

Client	[redacted]
Int. MR doc. ID	1
Actual Consumption	0.000000000000000
[redacted]	[redacted]
[redacted]	[redacted]
[redacted]	13.05.[redacted]

8. You have now successfully deleted this entry from the custom table.

Note: You can select several entries on the Data Browser screen and delete them with only one execution of the debugger tool to modify the value of “code” variable.

Selecting several entries from the Data Browser screen:

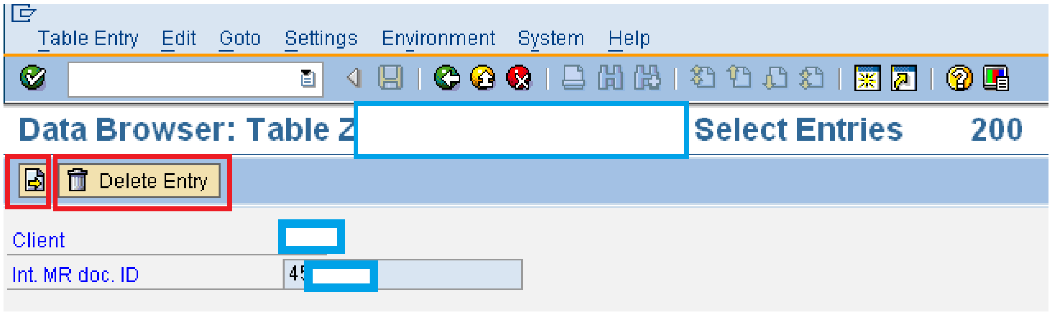


Data Browser: [redacted] 200

Table: [redacted]
Displayed Fields: 6 of 6 Fixed Columns: 2 List Width 0250

	Client	
□	[redacted]	[redacted]
□		
☑		
☑		
□		

Click on “Delete Entry” and then “Next entry”:



Data Browser: Table Z[redacted] Select Entries 200

Client	[redacted]
Int. MR doc. ID	45[redacted]

Related Content

ABAP Debugger

SAP Tables for Technical Consultants

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.