**selective removal from adso**

[Содержание](#Содержание)

# [Selective deletion in process chains](#Selective_deletion_in_process_chains)

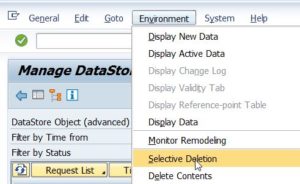
selective deletion of records from adso in abap

### ***How to read/write/delete from/to aDSO objects***

<https://blog.maruskin.eu/2022/12/how-to-readwritedelete-fromto-adso.html>

With the old InfoCubes and classic DSOs there was one very nice function in the menu Manage>>Content and this was “Selective deletion”.  Selective deletion means to delete specific values and leave the other values unaffected. These values are deleted based on specific selection of the characteristics. For example delete all values for fiscal year 2017 and period 1.

In the menu of advanced DSO there is actually a button Selective deletion. You can find it in the main menu when you already pressed “manage” for the ADSO:



This option is new and gives often errors, so there is an alternative way to perform it.

***Selective Deletion from Advanced DSO (Advanced Data Store Object)***

<https://www.dahlbeer.com/post/selective-deletion-from-advanced-dso-advanced-data-store-object>

Selective deletion from ADSO can be also done with a standard ABAP program - in transaction SE38, you need to execute program ***RSDRD\_DELETE\_FACTS***, select the Advanced DSO, enter the selection criteria and execute.

Before executing the program, there are 3 program properties available. If You choose *Generate deletion program*, you can set a name and use the program later. The *deletion program* can be used in process chains too.

Another transaction ***DELETE\_FACTS***, uses *the same ABAP program* for selective deletion form an Advanced DSO.

In the ADSO there is no request-based deletion available, so selective deletion remains the only option for removing data.

# *How to eliminate zero values from ADSO*

<https://www.nextlytics.com/blog/how-to-eliminate-zero-values-from-adso>

# *How to delete records from a Database table in SAP ABAP*

<https://abap-python.com/how-to-delete-records-from-a-database-table-in-sap-abap/#google_vignette>

Through *WHERE* clause

If at least one record is deleted from the Database table, then the system field is set to *sy-subrc* = 0. If the deletion is unsuccessful, then system field *sy-subrc* is set to *4*. The system field *sy-dbcnt* will give out the *count* of the records deleted.

DELETE FROM ztt\_db\_table2 *WHERE* contact\_id = 102.

Through Structure /Work area/

If the *single line* of record is found and deleted, then system field sy-subrc will be set to 0. If the record is not found and therefore not deleted, then the system field sy-subrc will be set to 4.

DATA ls\_contact TYPE ztt\_db\_table2.

ls\_contact – contact\_id = 101.

ls\_contact – contact\_name = ‘Thiru’.

ls\_contact – contact\_name = ‘Street 10’.

DELETE ztt\_db\_table2 FROM *ls\_contact*.

Through Internal table

DATA *lt\_int\_table* TYPE TABLE OF ztt\_db\_table2.

DATA ls\_data\_record TYPE *ztt\_db\_table2*.

ls\_data\_record – contact\_id = 103.

ls\_data\_record – contact\_name = ‘Mahesh’.

ls\_data\_record – contact\_address = ‘Street 10’.

APPEND ls\_data\_record TO lt\_int\_table.

ls\_data\_record – contact\_id = 102.

ls\_data\_record – contact\_name = ‘Mani’.

ls\_data\_record – contact\_address = ‘Street 11’.

APPEND ls\_data\_record TO lt\_int\_table.

DELETE *ztt\_db\_table2* FROM TABLE *lt\_int\_table*.

select \*

from

(SELECT current\_date, HOST, TABLE\_NAME,

sum(MEMORY\_SIZE\_IN\_TOTAL) as "MEMORY\_SIZE\_IN\_TOTAL",

sum(ESTIMATED\_MAX\_MEMORY\_SIZE\_IN\_TOTAL) as "ESTIMATED\_MAX\_MEMORY\_SIZE\_IN\_TOTAL"

FROM "SYS"."M\_CS\_TABLES"

where TABLE\_NAME in ( '/BIC/ABDNPSLPM2', '/BIC/ABDSLSDOC12',

'/BIC/ABDSLSGDS2', '/BIC/ABDSLSDOC2',

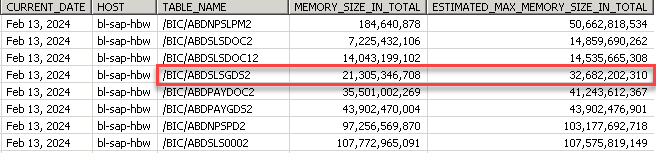
'/BIC/ABDPAYDOC2', '/BIC/ABDPAYGDS2',

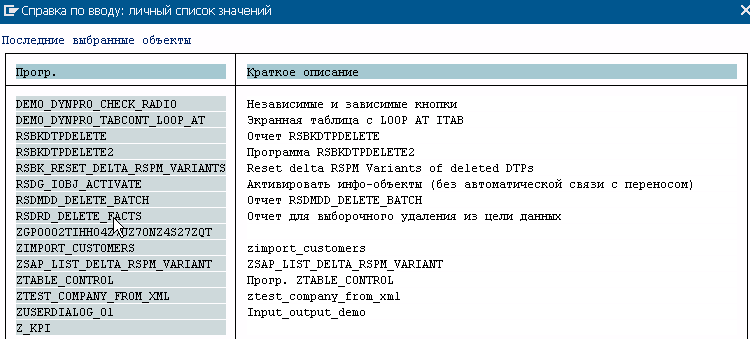
'/BIC/ABDSLS0002', '/BIC/ABDNPSPD2')

group by rollup(HOST, TABLE\_NAME)) as zx

where HOST in ('bl-sap-hbw') and TABLE\_NAME != '?'

order by "MEMORY\_SIZE\_IN\_TOTAL"





# 

# Selective deletion in process chains

[Содержание](#Содержание)

*from*

<https://blog.maruskin.eu/2022/12/how-to-readwritedelete-fromto-adso.html>

There is a set of SAP standard function modules for reading, writing and deleting data from/to the ADSO. They can easily be used in [custom ABAP code](https://blog.maruskin.eu/2018/12/bw4hana-custom-abap-code-in.html) in [BW’s transformations](https://blog.maruskin.eu/search/label/Transformations).

There are function modules representing APIs for DSO objects under name space RSDSO\_\* [or RSDRD\_\*]. In case of aDSO objects the API are there under name space *RSDSO\_\**.

|  |  |  |
| --- | --- | --- |
| **aDSO type** | **Operation** | **Method / API** |
| **Standard** | READ | Open SQL *SELECT; RSDRI\_INFOPROV\_READ* |
|  | WRITE | *RSDSO\_WRITE\_API; RSDSO\_WRITE\_API\_RFC* |
|  | ACTIVE | *RSDSO\_ACTIVATE\_REQ\_API\_RFC* - Activates requests in aDSO. Multiple requests can be activated separately or whether or system can activate as many requests as possible at one shot. |
|  | DETELE | N/A |
| **Direct Update** | READ | Open SQL *SELECT* |
|  | WRITE | *RSDSO\_DU\_WRITE\_API; RSDSO\_DU\_WRITE\_API\_RFC*– Writes data from an itab into active data table of the aDSO. Each API call results in a new request. Database transaction is committed automatically. |
|  | DELETE | *RSDSO\_DU\_DELETE\_API\_RFC* - Deletes data from an aDSO. Whole content can be deleted or data can be deleted based on a selective deletion. |
|  | CLEANUP | *RSDSO\_DU\_CLEANUP\_API\_RFC* - Changes status of red requests in aDSO to green. Red requests are blocking further data loads thus have to be corrected. Only requests that were loaded via API are considered. |

FMs ending with *\*RFC* are supposed to be used for remote scenarios whereas other FMs are locally to be used where the aDSO and calling code resides within the same system.

Supporting FMs

*RSDSO\_DEBUG\_API* – It enables a user to debug above listed RFC enabled APIs for aDSO.

<https://community.sap.com/t5/technology-blogs-by-members/selective-deletion-in-process-chain/ba-p/13005016>

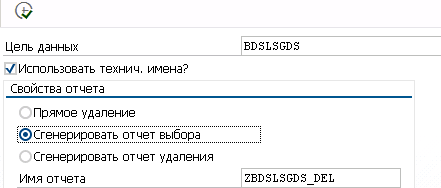
Scenario

Suppose we have a infocube where we need to recalculate some values everyday considering *last three weeks* ⇒ duirng everyday's load there will be some data already present in the cube which need to be reloaded to recalculate the values. In such case we need to *delete that particular data range* else it will create data duplication in Infocube. In this case we need to *delete last three weeks data* before loading to Infocube.

How to Do

We can achieve this by using *DELETE\_FACTS*Transaction code.

Go to T-code *DELETE\_FACTS*and give InfoCube name. Select *Generate selection program* option and then execute.



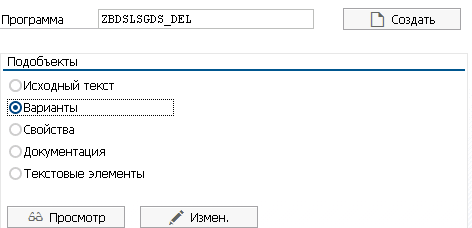
ZBDSLSGDS\_DEL

Our target is to delete data for last three weeks. We can achieve this in two ways.

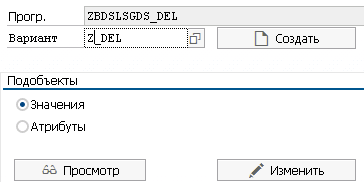
1. Using *variant*
2. Copying auto generated program into *Z program* and change inside the code

***First Option - Using Variant***

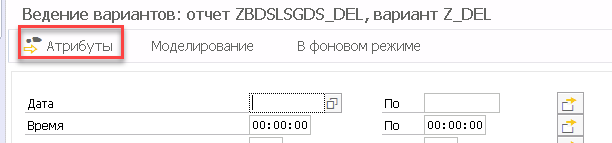
*Step 1* - Take this Z program and then go to SE38. Give program name and click on *Variants* option. Click on *Display*.



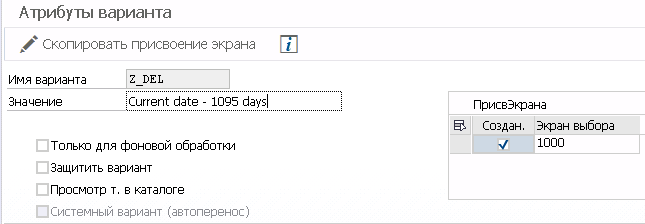
*Step 2* - Give a *Variant Techenical name* and click on *Create*.



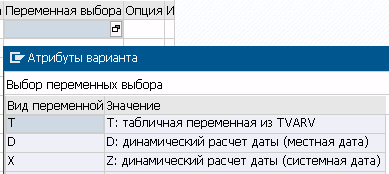
*Step 3* - It will go to Values Screen of the Variant. Click on *Attributes*.



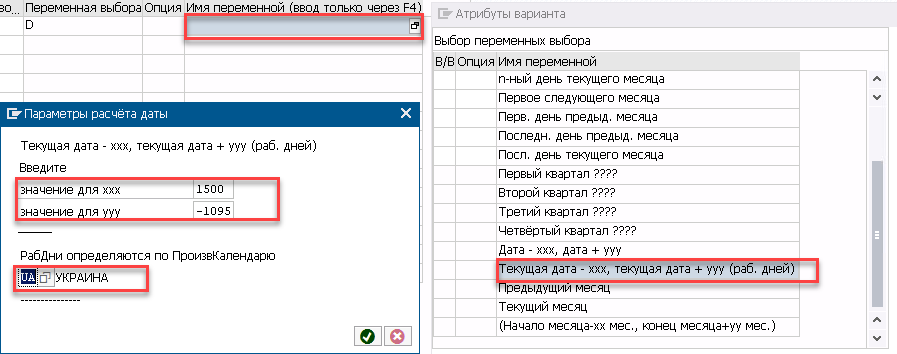
*Step 4* - Give *description* of the Variant.



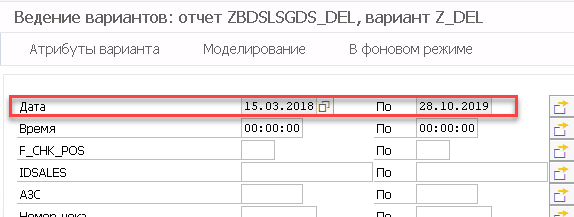
*Step 5 -* Now find Calendar Day and select on Selection Variable field. Select Type of variable as "D"



*Step 6 - Click on name of variable field* and chose Variable which will give option of Current date - XXX, Current Date + YYY option. Enter 21 as XXX and 0 as YYY as we need to delete three weeks data (21 days)



*Step 7 -* Then *save the variant* and you will be able to see the variants with the caledar day values. It is reflecting last 21 days from current day.



SELECT COUNT( \* ) AS "CNT"

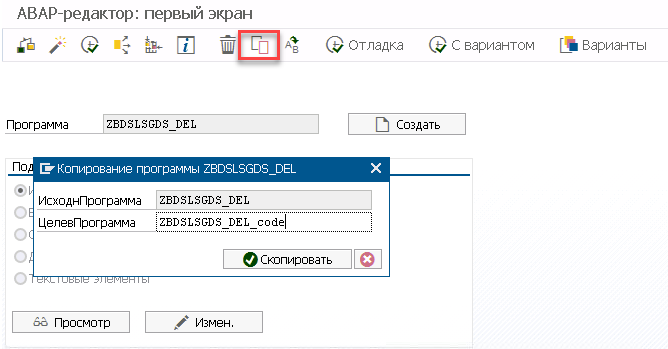
FROM "/BIC/ABDSLSGDS2" "A"

WHERE ("A"."DATE0" BETWEEN '20180315' AND '20191028' )

*Step 8 -* Now *add this Z program* *with the created variant* in the *process chain* to delete last three weeks data.

**Second Option - Changing inside ABAP Program**

*Step 1 -* Go to SE38 and enter Z program in the Program field. Clcik on *Copy* and copied this SAP generated program into a Custom Z program



*Step 2 -* Find 0CALDAY field in the newly copied Z program and change the code so that it takes last 21 days values from current date. Check if this field is Initial (not having any value)  and then Pass sy-datum in *l\_s\_range-high* and sy-datum - 21 in *l\_s\_range-low field*.

…

END-OF-SELECTION.

IF NOT *C001*[] IS INITIAL.

CLEAR L\_SX\_SEL.

L\_SX\_SEL-IOBJNM = '*0DATE*'.

LOOP AT C001 .

CLEAR L\_S\_RANGE.

MOVE C001-SIGN TO L\_S\_RANGE-SIGN.

*" MOVE C001-OPTION TO L\_S\_RANGE-OPTION.*

MOVE 'BT' TO L\_S\_RANGE-OPTION.

MOVE *sy-datum - 1500* TO *L\_S\_RANGE-LOW*. *" 5 years ago*

MOVE *sy-datum - 1095* TO *L\_S\_RANGE-HIGH*. *" 3 years ago*

MOVE RS\_C\_TRUE TO L\_S\_RANGE-KEYFL.

APPEND L\_S\_RANGE TO L\_SX\_SEL-T\_RANGE.

ENDLOOP.

INSERT L\_SX\_SEL INTO TABLE L\_THX\_SEL.

ENDIF.

IF NOT *C002*[] IS INITIAL.

…

MOVE 'BT' TO L\_S\_RANGE-OPTION.

