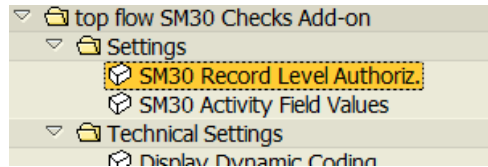
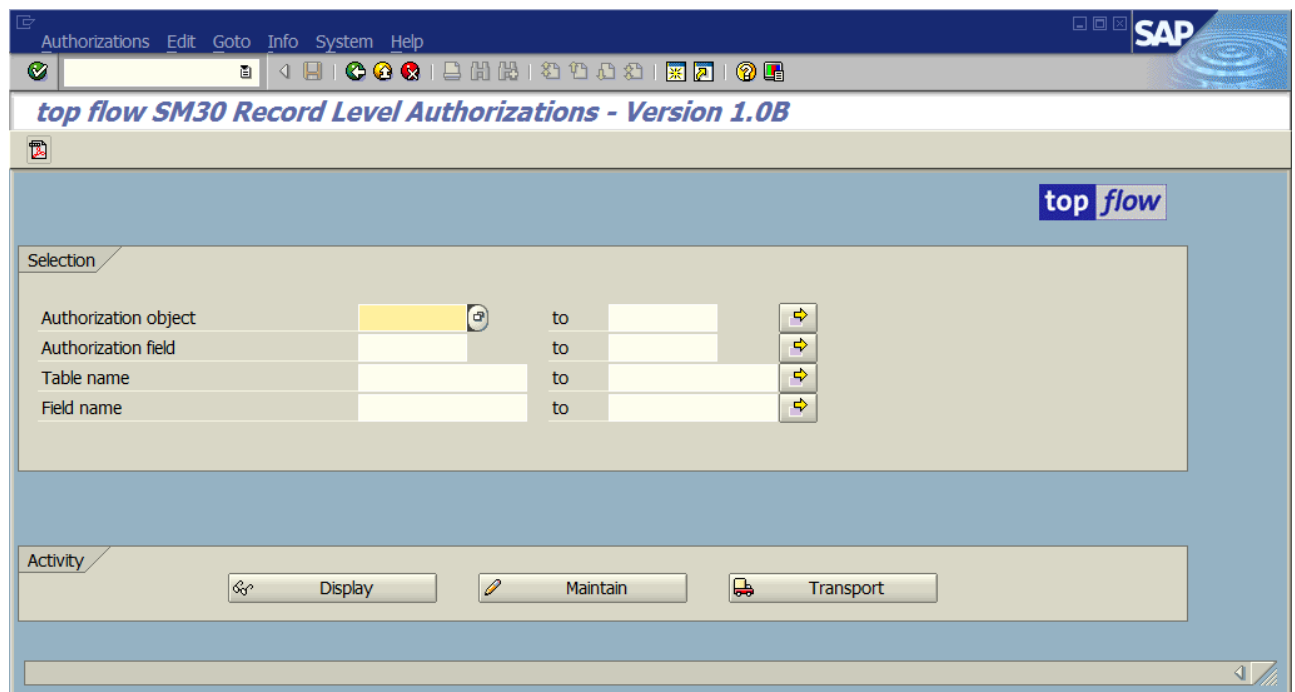


SM30 Record Level Authorizations

Please call transaction **/TFTO/SM30_SETTINGS**:



Double click on **SM30 Record Level Authoriz.** and the following selection screen shows up:

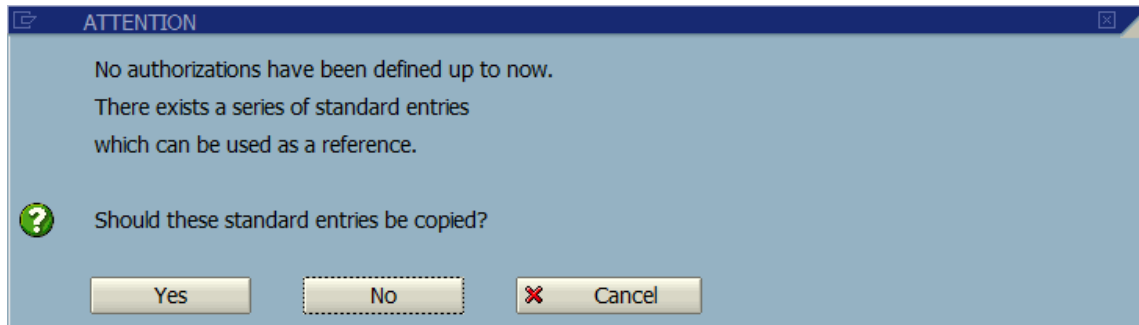


Authorized are the users which have one of the following roles assigned to them:

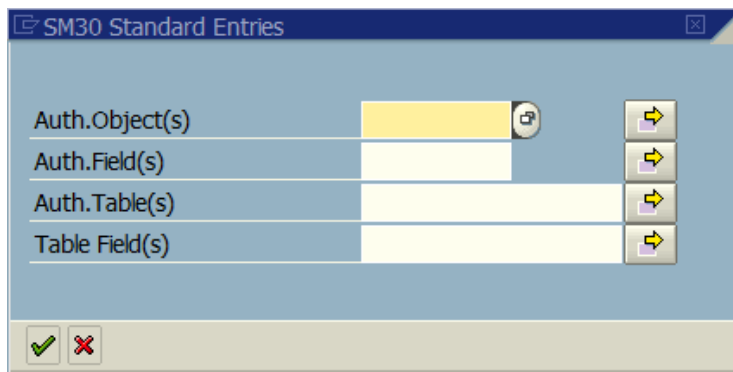
/TFTO/SM30_GLOB_MAINT	SM30 Global Settings	Maintenance
/TFTO/SM30_GLOB_DISPL	SM30 Global Settings	Display
/TFTO/SM30_AUTH_MAINT	SM30 Record Level Authorizations	Maintenance
/TFTO/SM30_AUTH_DISPL	SM30 Record Level Authorizations	Display

Instead of the roles, authorization objects **/TFTO/S3GL** or **/TFTO/S3AU** may be assigned (please refer to [SM30 Roles and Authorization Objects](#)).

After pressing the  button for the first time the following dialog window pops up :

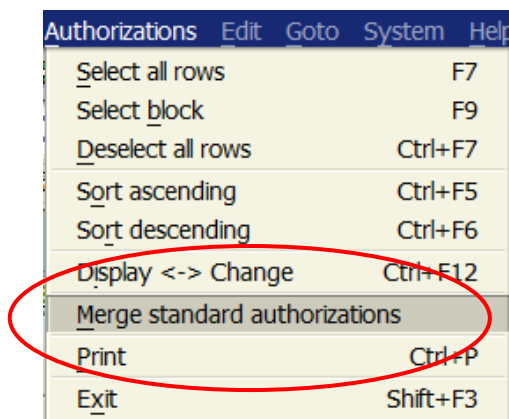


If desired it is possible to restrict the entries to be copied:

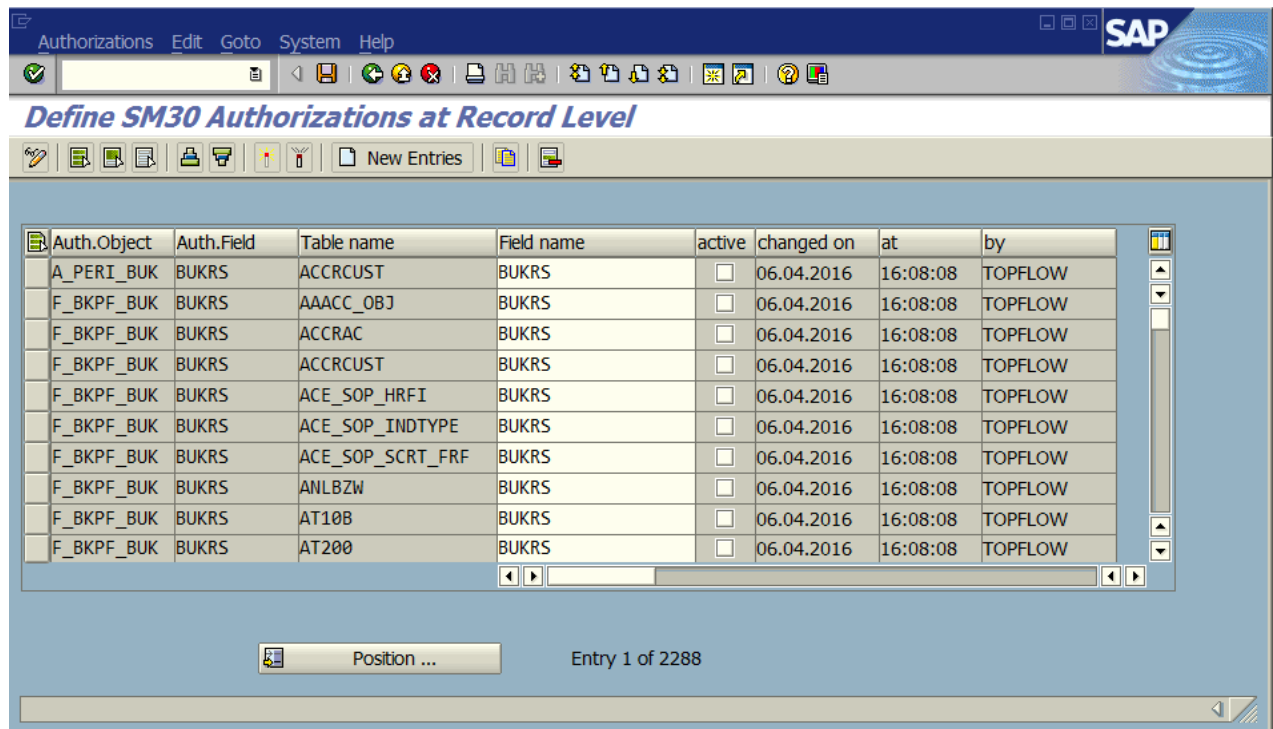


It is recommended to copy the reference entries in order to have a starting point for the ensuing definitions. The entries are copied with state „inactive“ – in this way they do not interfere with any possible authorization checks.

Should you ignore this offer for the moment, you can still copy the reference entries at a later time. The corresponding function can be found in the menu **Authorizations**:



In case the reference entries have been copied, the maintenance screen has the following appearance:



The screen is similar to a standard maintenance dialog – taking a closer look you detect functions (for example sort) which are not available in the standard.

To begin with a quick look at the involved columns:









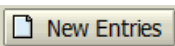




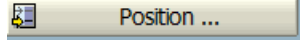
Column	Description
Auth.Object	Authorization object – for example <i>V_VBAK_VKO</i>
Auth.Field	Authorization field – for example <i>VKORG</i>
Table name	Database table – transparent / Pool / Cluster Views are not accepted. The authorization checks for views are based on the underlying database tables.
Field name	Field of the database table
active	Flag – only active entries are taken into account for the checks
changed on	Date of the last change
at	Time of the last change
by	Logon name of the user that made the last change

Available Functions

The application toolbar in maintain mode is as follows:

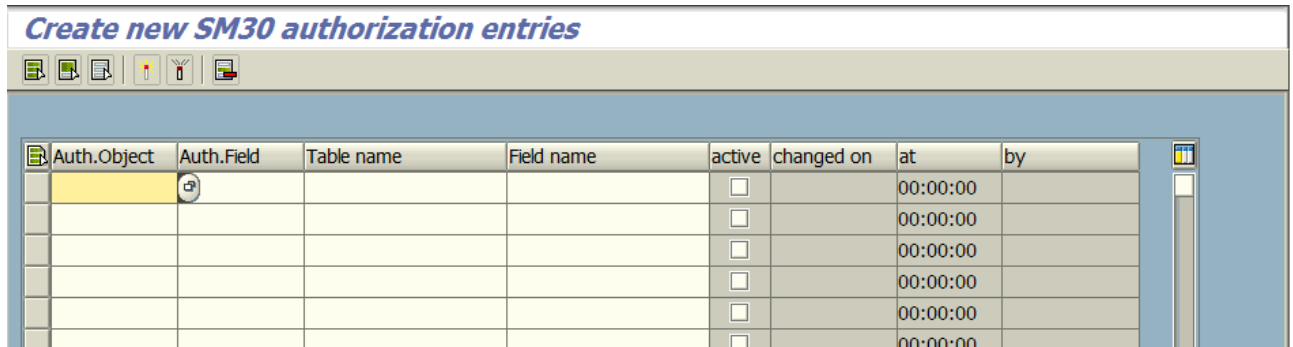


Jede Funktion wird nun kurz beschrieben.

Funktion	Beschreibung
	Toggle between maintain and display mode
	Select all entries
	Select block of entries
	Deselect all entries
	Sort in ascending order
	Sort in descending order
	Activate selected entries
	Deactivate selected entries (better than delete)
	Define new entries
	Copy selected entries as templates for new ones
	Delete selected entries
	Save changes to the database
	Print. A list of the entries is output by means of ALV. This list can then be printed.
	Position on a given entry

Definition of new entries

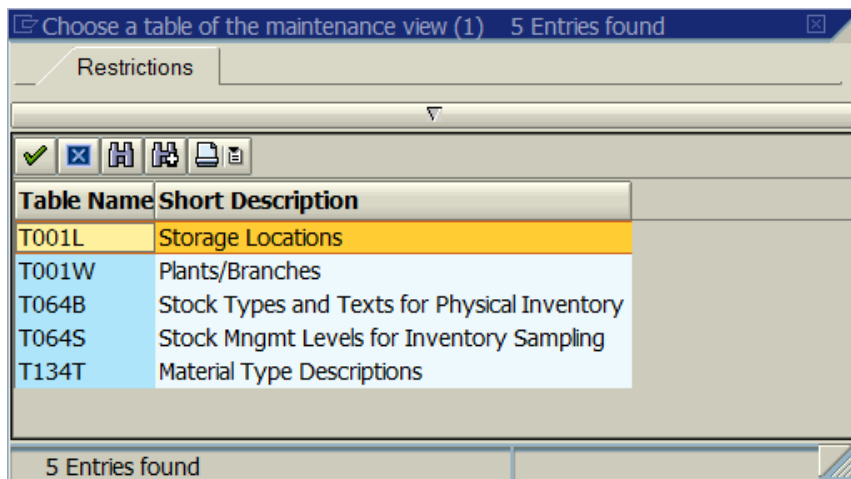
After pressing an empty screen appears for creating new entries:



Transaction SM30 makes mostly use of **maintenance views**. The present definition dialog, however, requires a **real table**. Question: Is it necessary for the user to find out each time by means of **SE11** on which database tables a given maintenance view is based? **The answer is no**. When the name of a maintenance view is entered into column "Table name", a popup window for choosing one of the involved tables shows up as soon as ENTER is pressed, like in the following example for view **V_064S_1**:

Auth.Field	Table name	Field name
	V_064S_1	

...



If only **one** table is involved in the maintenance view, the program replaces the name of the view with the name of the basis table **automatically**:

Example: Entering **V_001_D** → results in **T001**

In this way the first step of the definition process has been carried out.

This is usually followed by the choice of a field of the specified table. It is the content of this field that is examined during the authorization check.

The F4 help is ideal for choosing the table field:

Object	Auth.Field	Table name	Field name	active	change
		T001		<input type="checkbox"/>	<input type="checkbox"/>

...

Please choose a table field (1) 76 Entries found

Restrictions

Field Name	DTyp	Length	Data element	Short Description
MANDT	CLNT	000003	MANDT	Client
BUKRS	CHAR	000004	BUKRS	Company Code
BUTXT	CHAR	000025	BUTXT	Name of Company Code or Company
ORT01	CHAR	000025	ORT01	City
LAND1	CHAR	000003	LAND1	Country Key
WAERS	CUKY	000005	WAERS	Currency Key
SPRAS	LANG	000001	SPRAS	Language Key

76 Entries found

In our example we choose BUKRS:

Object	Auth.Field	Table name	Field name	active	change
		T001	BUKRS	<input type="checkbox"/>	<input type="checkbox"/>

Now a suitable authorization object can be looked for. Also using the F4 help:

Auth.Object	Auth.	Table name	Field name
		T001	BUKRS

...

Please choose an authorization object (1) 299 Entries found

Restrictions

Object	Text
F_KK_ODBUK	FI-CA Request: Company Code Authorization
F_KNA1_BUK	Customer: Authorization for Company Codes
F_KNB1_ANA	Customer: Authorization for Account Analysis
F_LFA1_BUK	Vendor: Authorization for Company Codes

299 Entries found

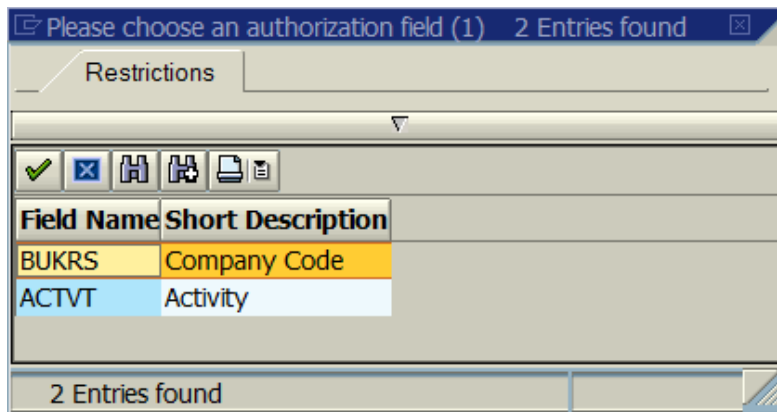
We choose for example F_KNA1_BUK:

Auth.Object	Auth.Field	Table name	Field name
F_KNA1_BUK		T001	BUKRS

Only the matching authorization field is missing for the definition to be complete. We make use once more of the F4 help:

Auth.Object	Auth.Field	Table name	Field name
F_KNA1_BUK		T001	BUKRS

...



BUKRS is the only suitable authorization field:

Auth.Object	Auth.Field	Table name	Field name
F_KNA1_BUK	BUKRS	T001	BUKRS

Now the new entry is complete.

NOTE:

The above described sequence is not mandatory. You can also begin by entering an authorization object, followed by an authorization field, and finally a database table with one of its fields. Or you specify first an authorization field like for example VKORG, and look after that for a matching authorization object.


The F4 help takes the already entered values into account – it is thus much easier to find matching entries.

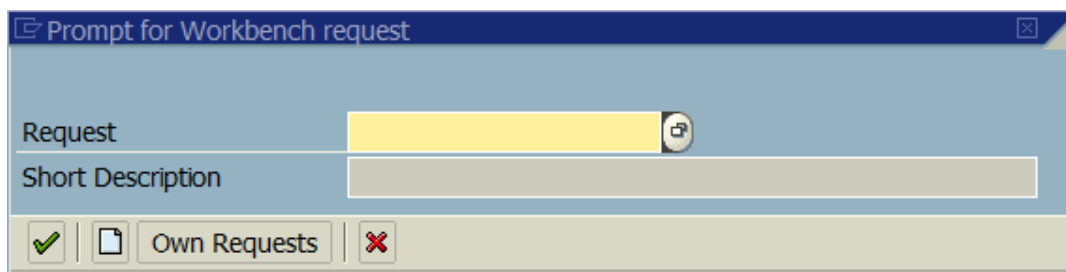
Transport of Entries

It is also possible to transport the defined entries.

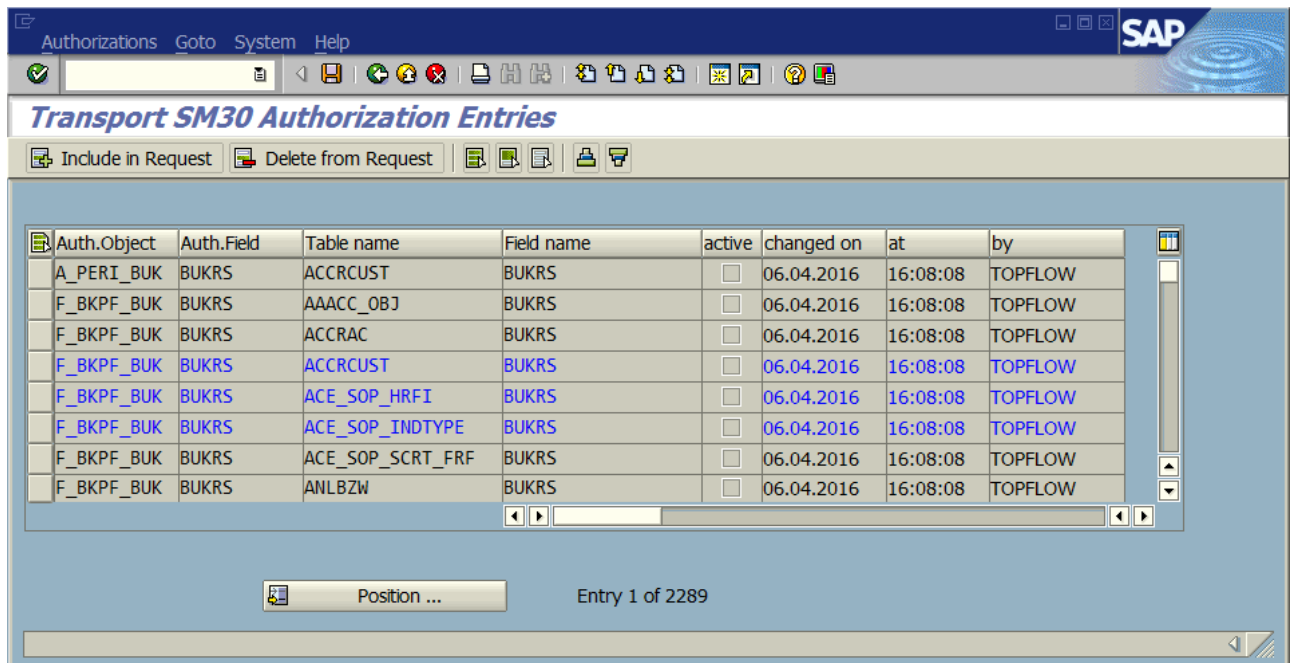
The transport functionality has to be chosen on the initial screen:



As soon as you press the  **Transport** button, the program will prompt you for a workbench request:



Now a list of the selected entries is displayed. In this list you can choose the entries to be transported. The ones already in the transport request are highlighted:

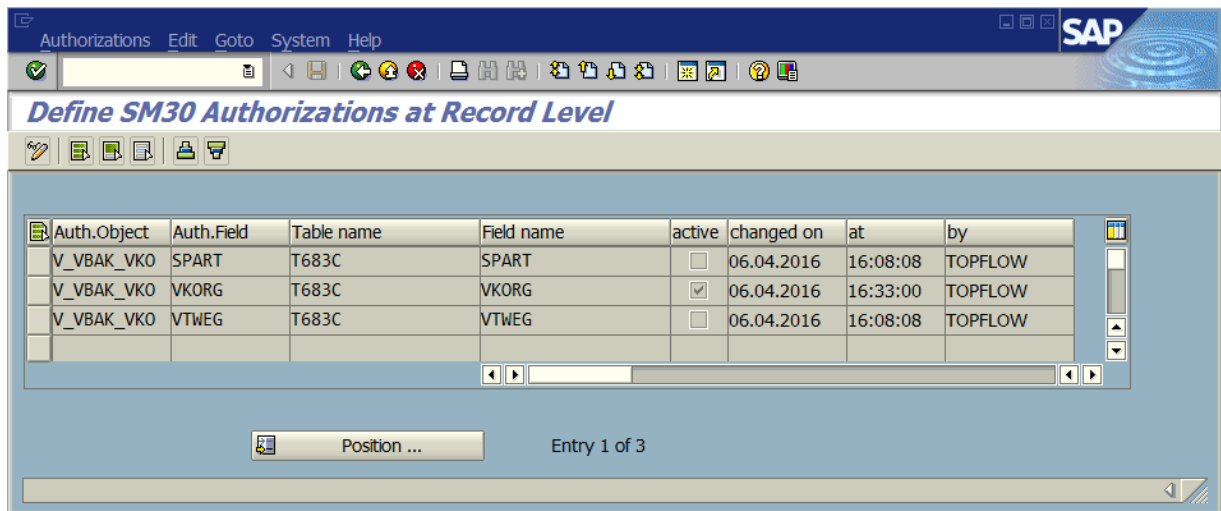


The transport request is updated when you press .

How the Authorization Checks are carried out

In order to show how the SM30 Add-on works, an example will be made using table **T683C** in combination with authorization object **V_VBAK_VKO**.

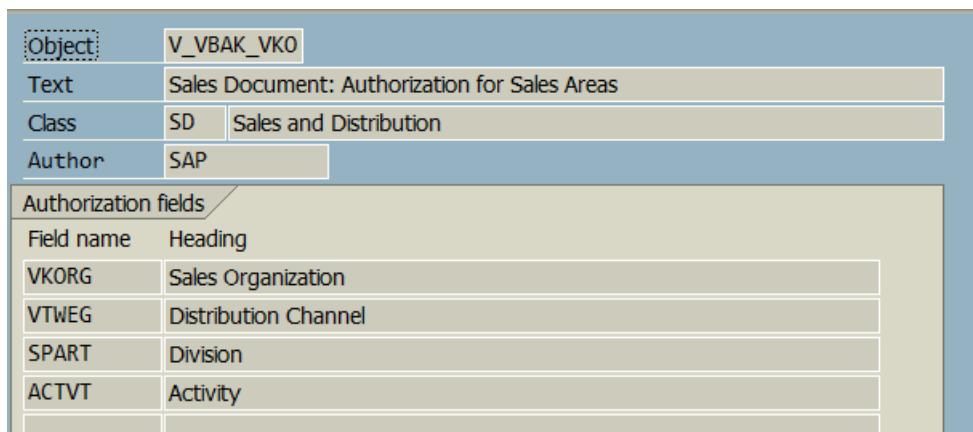
The corresponding entries could be as follows:



Auth.Object	Auth.Field	Table name	Field name	active	changed on	at	by
V_VBAK_VKO	SPART	T683C	SPART	<input type="checkbox"/>	06.04.2016	16:08:08	TOPFLOW
V_VBAK_VKO	VKORG	T683C	VKORG	<input checked="" type="checkbox"/>	06.04.2016	16:33:00	TOPFLOW
V_VBAK_VKO	VTWEG	T683C	VTWEG	<input type="checkbox"/>	06.04.2016	16:08:08	TOPFLOW

Position ... Entry 1 of 3

A double click on the authorization object shows its definition:



Object	V_VBAK_VKO
Text	Sales Document: Authorization for Sales Areas
Class	SD Sales and Distribution
Author	SAP
Authorization fields	
Field name	Heading
VKORG	Sales Organization
VTWEG	Distribution Channel
SPART	Division
ACTVT	Activity

As may be seen, the object features three normal authorization field.

Field **ACTVT** represents the activity – its values are defined in the dialog "SM30 Activity Field Values". For our example we will make use of value '02' (change).

Since only the entry for field **VKORG** has been activated, the SM30 add-on performs the authorization check with only this field. The remaining fields appear in the **AUTHORITY-CHECK** statement in combination with **DUMMY**.

In our example we use maintenance view **V_T683C**, which is based on **T683C**.

In the generated dynamic coding we find the following authorization check:

```
AUTHORITY-CHECK OBJECT 'V_VBAK_VKO'  
  ID 'VKORG' FIELD LS_A001-VKORG  
  ID 'VTWEG' DUMMY  
  ID 'SPART' DUMMY  
  ID 'ACTVT' FIELD '02'.
```

Each selected V_T683C entry is subjected to this check.

The authorization check turns out differently, if all three fields (VKORG, VTWEG and SPART) are activated:

```
AUTHORITY-CHECK OBJECT 'V_VBAK_VKO'  
  ID 'VKORG' FIELD LS_A001-VKORG  
  ID 'VTWEG' FIELD LS_A001-VTWEG  
  ID 'SPART' FIELD LS_A001-SPART  
  ID 'ACTVT' FIELD '02'.
```

NOTE: Fields that are initial are not checked. If the authorization check involves three fields, all three fields must be filled. Otherwise the authorization check is not performed.

If for a given maintenance view many authorization objects are defined, each record is subjected to an authorization check for each object. A record has to pass all AUTHORITY-CHECKs in order to make it to the result list.

This is not the case for more than one activity field value. If for example the values '01' and '02' have been defined for activity field ACTVT, a given record passes the authorization check if at least one of the values is successful.