Authorizations In An SAP BW Project

BUSINESS INFORMATION WAREHOUSE

Document Version 1.02

SAP AG assumes no responsibility for errors or omissions in these materials.

These materials are provided “as is” without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall not be liable for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials.

SAP does not warrant the accuracy or completeness of the information, text, graphics, links or other items contained within these materials. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third party web pages nor provide any warranty whatsoever relating to third party web pages.

1 Introduction

In a software project the authorization concept usually requires a sub-project with different phases similar to the functional implementation itself. Even more so in an SAP BW project. In comparison with an SAP ERP implementation there is another level of complexity involved in the authorization concept. In many projects this topic is taken care of at a late stage of the project if not too late.

2 Project Phases

Whichever project methodology you are following, there are always different phases like e.g. blueprint, design, implementation, testing. As a general guideline, all phases should contain steps taking care of authorizations similar to steps focussing on the core functionality. Authorization requirements need to be collected, a concept must be developed, objects need to be implemented and testing of authorizations is necessary.

If you have very simple authorization requirements you may find that some steps are not required, nevertheless you should have this fact documented and signed off. On the other hand, complex authorization requirements may have an impact on the design and implementation, i.e. you may to have to adjust data models according to authorization needs. That’s why the authorization considerations must be in sync with the project life cycle.

3 SAP BW Authorization Specifics

In an SAP BW system there are two different types of authorization objects.

- Standard authorization objects: This type of authorization objects is provided by SAP and covers all checks for e.g. system administration tasks, data modelling tasks, and for granting access to InfoProviders for reporting. For this type of authorizations the same concept and technique is used as in an SAP R/3 system.
- Reporting authorization objects: For more granular authorization checks on an InfoProvider’s data you need another type of authorization objects defined by the customer. With these objects you can specify which part of the data within an InfoProvider a user is allowed to see.

Both types of authorization objects use the same authorization framework. Technically they are treated in the same way. However, the design of reporting authorizations is more complex because you need to design the reporting authorization objects first. This is an additional step that needs to be treated with care because the structure of the authorization objects determines the possible use in regards to selections, combinations and granularity. In your project you need expertise in the area of reporting authorizations; knowledge of the basis authorization framework is not sufficient.

4 Knowledge of Security Strategy of SAP & BW Reporting Authorizations

Most general, the possible guiding strategies of authorizations can be

1. Permission of everything which is not prohibited explicitly.

2. Permit only in case that everything you are going to see is cover by authorizations.

The second strategy is chosen in R/3 and BW for various reasons. Therefore you must have all authorizations for data you are going to see. Otherwise you receive a message stating “You have no authorization”.

The first reason for this strategy is a strong security requirement in order to not permit display of data due to missing or forgotten prohibition.

The second reason is the end user expectation that identical reports shall be displayed identically, no respect which user with possibly different authorizations is watching it, and in lack of a complete knowledge of his authorizations and missing authorizations assuming wrong data.

Especially one has to emphasize the main property of reporting authorizations which is a result of the second reason: from the beginning, reporting authorizations do not work as a filter on a larger set of data! In contrast, the selections have to filter out first all data which are not allowed *before* the query is actually executed. This can be done with fixed filters or, much better and flexible, with variables which are of the type “filled by authorizations”. This pre-filtering is the most critical point in the modelling of reports and reporting authorizations.

With filtering and authorizations in one dimension (=characteristic/activities/key figure etc.) this process is not very difficult but using more dimensional authorization combinations may lead through various pit falls and complications of the modelling process. Typical situations are described in consulting OSS notes: 557924, 653383 and 654947. We will show some detailed examples here.

5 Technical Details and Examples

5.1 Options

By means of a reporting authorization object you can determine what a user is allowed to work with:

- which characteristic values
- which hierarchy nodes and subtrees of hierarchies
- which key figures
- which activity (for BW-BPS: display or change)
- any combination of the above
- which display attributes

5.2 Examples

Assume that a sales manager is assigned to sales organization (InfoObject 0SALESORG) “Germany” (technical key 1100) and is responsible for distribution channel (0DISTR_CHAN) “Direct Sales” (DS). For Germany and Direct Sales the manager is allowed to see all key figures.

For the region EMEA (a node of a hierarchy defined on 0SALESORG) the sales manager is allowed to see the revenue (0REVENUE) for “Direct Sales” and summarized over all distribution channels.

In order to accomplish this you have to define a reporting authorization object combining the different objects. The structure must be as follows (in arbitrary order):

Field	Explanation
1KYFNM	Required to define authorizations for key figures
0SALESORG	Sales Organization characteristic
0DISTR_CHAN	Distribution Channel characteristic
0TCTAUTHH	Required to define authorizations for hierarchies / hierarchy nodes

In order to grant authorizations to the sales manager as described you have to define them as follows:

Example: Authorization 1:

Field	Value	Explanation
1KYFNM	*	All key figures
0SALESORG	1100	Only Sales Organization "Germany"
0DISTR_CHAN	DS	Only Distribution Channel "Direct Sales"
0TCTAUTHH	DUMMY (SPACE)	Not required

Example: Authorization 2:

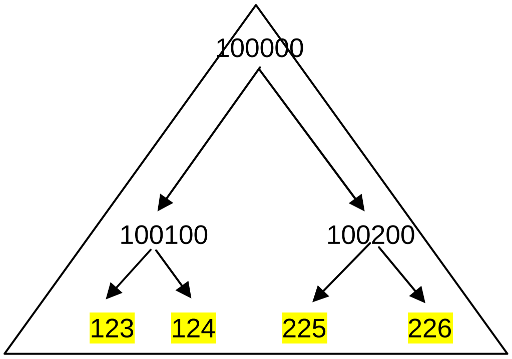
Field	Value	Explanation
1KYFNM	0REVENUE	Only key figure "Revenue"
0SALESORG	DUMMY (SPACE)	Defined by hierarchy node, so no meaningful entry required
0DISTR_CHAN	DS, :	"Direct Sales" and ":" for summarized view
0TCTAUTHH	NODE_EMEA	This value must refer to a hierarchy node definition for authorizations specifying the node EMEA. The value "NODE_EMEA" is arbitrarily chosen. One can also define several nodes and subtrees below.

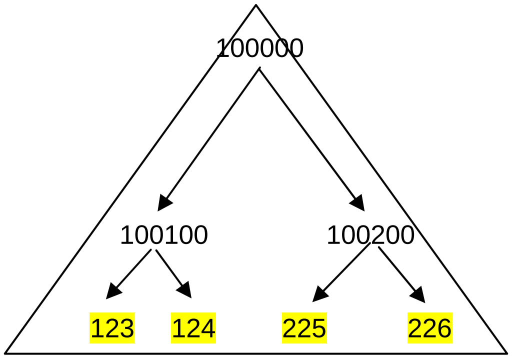
In principle there are all combinations of all types of authorizations possible. In addition to characteristic values it is possible to protect hierarchy structures and to authorize them with hierarchy node authorizations as sketched above. However, this implies a strict division between characteristic values and hierarchy structure elements (nodes and parts of hierarchies which can be interpreted as "triangles", see example sketch below).

With this strict division it is for example possible to protect the display of hierarchical structures of employees or cost enters of a company. If an end user does not have authorizations for the full structure, which means *all nodes*, he won't be allowed to see it's data. This also implies that is never possible to authorize hierarchy nodes by just authorizing the leaves below them because in this example the node structure wouldn't be included.

Vice versa it *is* possible to authorize characteristic values by authorizing a node and everything below such that the included leaves cover the selected (=requested for display) characteristic values.

Example:

Authorization: Single characteristic values	Selection of node
Cost Center 123 Cost Center 124 Cost Center 225 Cost Center 226	Cost Center hierarchy 
Authorization *is not* sufficient !	

Authorization: Node plus subtree (=triangle)	Selection: Single characteristic values
Cost Center hierarchy  	Cost Center 123 Cost Center 124 Cost Center 225 Cost Center 226
Authorization *is* sufficient ! (maybe slow)	

For the sake of performance one should have in mind which kind of authorizations can cover which kind of selections and which combination makes sense. Quite generally we encounter four areas of check situations, illustrated in the following table:

Table 1: Performance and Combinations of Selections and Authorizations

Combinations: Selection/Authorization	Value selection (values, intervals)	Hierarchy selection
Value authorization (single values, intervals)	Usually okay, but depends on number of values in selection and authorization(s) -> maybe hierarchies authorization(s)	Not possible![1]
Hierarchy authorizations	Possible, but generically performance critical, depends on number of selected values and hierarchy authorization size (number of nodes etc.).	Okay (preferred!)

5.3 Maintenance vs. Functionality

As you can see in the examples above, authorizations can be granted on a very detailed level and may have several pitfalls which, however, origin in the powerful concepts. Maintenance of authorizations usually gets more expensive the more detailed the authorization concept is. There are several concepts that help to mitigate the maintenance effort:

- Good knowledge of security strategies and concepts and their interplay.
- Usage of variables in authorization definition
- Loading authorization information and generating the authorizations
- Manual maintenance tool for faster maintenance

Don't forget to set up a concept for maintenance and estimate the effort involved at an early stage in the project.

5.4 A few Important Tips & Hints

5.4.1 Support Pack Release Strategy

SAP recommends minimally a “Quarterly Support Pack Release Strategy” or sooner. To keep the software release as current as possible is generally a good policy to apply software corrections. This is of particular importance in the area of security to insure no loop hold or exposures of security error be permitted. Therefore, it is recommended for customers to apply most recent support pack which may include security patches for reporting authorization issues at the earliest possible time.

5.4.2 Huge number of Roles can impact performance

It is important to understand the security requirements from a business point of view and organized using a security matrix to avoid defining a huge number of roles that could create a maintenance nightmare and also resulted in a performance issue.

[1] only exceptions: hierarchy not displayed and leaf or chargeable node as filter

5.4.3 Reporting - a BW authorization problem

To expedite the problem resolution process when you report a possible BW authorizations problem, it is important o create an effective OSS message with following elements in mind:

- Prepare a query which is as simple as possible and still reproduces the error
- Prepare a SAP_ALL user and a restricted user.
- If you use variables (customer exits) replace their content into profile of the restricted user

  Please be advised that we do not support customer code.
- Explain clearly what you expect to see and what the error is.
- Don't forget to give all the necessary information: usernames, passwords, System, names
- Open the system.