
Ensure and track security across your SAP system landscape using SAP NetWeaver 7.0 BI analysis security

by Joerg Boeke



Joerg Boeke
BI Director/BW Consultant,
syskoplan AG

Joerg Boeke is an authorized signatory BI director and a BW consultant for syskoplan AG. He has worked with SAP BW since 1998 focusing on designing, implementing, and enhancing automated scenarios. He created an SAP-certified BI data interface for non-SAP applications and a customizable BW add-on tool to automatically refresh and distribute BW workbooks internally in SAP BW or to external users in XLS or PDF format. You may contact him at joerg.boeke@syskoplan.com.

A potential customer recently asked me which new features in SAP NetWeaver 7.0 (formerly 2004s) he could use to secure his company's system and protect its analysis reporting. To answer this question correctly, I had to ask, "What kind of security and what features do you currently use to protect your SAP NetWeaver Business Intelligence (BI) environment?" The customer replied, "That's difficult to explain because we aren't using the basic administration authorization that our Basis team provides or the reporting security that our SAP Customer Relationship Management (SAP CRM), SAP Financial Accounting (FI), SAP Controlling (CO), and Sales and Distribution (SD) teams provide. We haven't implemented a common security model yet."

While this answer isn't surprising, it does give me a starting point to explain BI 7.0¹ analysis security. BI 7.0 analysis security is critical to every company that wants to secure and monitor general access to systems across its landscapes in terms of SAP transactions. It's also useful to companies wanting to secure analysis reporting by restricting, for example, the types of front-end programs, such as Microsoft (MS) Excel or MS Internet Explorer,² and the types of data, such as MS Excel spreadsheets and financial records formatted by SAP Report Designer, that users are allowed to see on screen or in print. Moreover, today's companies need to comply with Sarbanes-Oxley Act regulations and guidelines. No company can afford to have a hazy view or an undefined strategy of security anymore.

If someone asked you about the specific user authorization required for your company's FI and CO systems or for access to your customer-based data, could you explain with clarity and specifics what a particular user has

¹ In this article, I refer to SAP NetWeaver 7.0 Business Intelligence as BI 7.0.

² SAP refers to reporting in Microsoft Internet Explorer as WebReports or Web reporting, terms I use throughout this article.

permission to see or access? Your explanation should leave no doubt about the authorizations for any aspect of your systems. A comprehensive approach and strategy is not an option today; it's mandatory.

While the legal regulations are indeed mandatory, the business side of creating a reportable security landscape — with the help of BI 7.0 analysis security — is equally important. You must provide a consistent and secure landscape across SAP ERP and BI. For example, when you're determining what information a user can see about employee salaries, a user in ERP shouldn't be able to see more than the equivalent user on the BI side, and vice versa.

Users have common concerns regarding how to integrate a new authorization concept, which restricts access to the entire BI component, as well as to specific sensitive data that's found in both BI 7.0 and other existing SAP applications. This may be complicated because you can use a different security strategy to implement each application landscape, such as FI, CO, SAP CRM, and SD. No common security model applies to all of these applications.

In this article, I describe how you can use BI 7.0 analysis security to protect your security system landscape and track safety measures so that you proactively maintain that environment. I discuss why it is critical for a company to have a well thought-out security plan. I want you to understand security and its maintenance as fully as possible. I also show you the newly available security tools and features in BI 7.0 and explain the concept of BI 7.0 analysis security. You'll discover how to set up security in your BI environment with a real-life example and the tools

that you need to implement BI 7.0 analysis authorization, including testing your system. In addition, I show you methods for populating authorization profiles to handle redundant work automatically.

Security applies to the ability to access your BI system, but it's also a question of which front end you should use for reporting. It includes how to deal with security issues, such as resetting a forgotten password. You should include all security issues in a common model, and they should be similar for all SAP systems to ensure consistency system-wide.

Let's begin with a security-related project I worked on recently. The situation forced me to think about security in general and, perhaps more importantly, about methods for enforcing security while keeping it simple to implement and maintain. The client in this scenario is a large corporation with more than 10,000 employees, many of whom are reporting consumers of BI.

In this company, one BI administrator had permission to create new users in a BI system. However, he was in the process of transferring to another position. Just before he moved, he created a copy of his own BI user profile so that he could access BI later. When he moved, the company terminated his BI user profile. Having set up the reporting functionality using BI 7.0, which is secure, we were able to trace both his user profile and its copy and delete both.

What's new in security in BI 7.0?

BI 7.0 employs a new concept of analysis security. Although security in SAP Business Information Warehouse (BW) 3.5 was good, BI 7.0 is even better. The following capabilities underlie this new concept:

- Turning on analysis security authorization globally
- Specifying authorization on attributes without restricting the parent object
- Accessing security from a single point of entry via transaction RSECADMIN (Management of Analysis Authorizations)

Note!

This article is for administrators and those reporting users who need to implement or audit security. CFOs and CIOs can also benefit from this article if they are asked to plan an internal security implementation. To understand this article, you should have a general knowledge of BI.

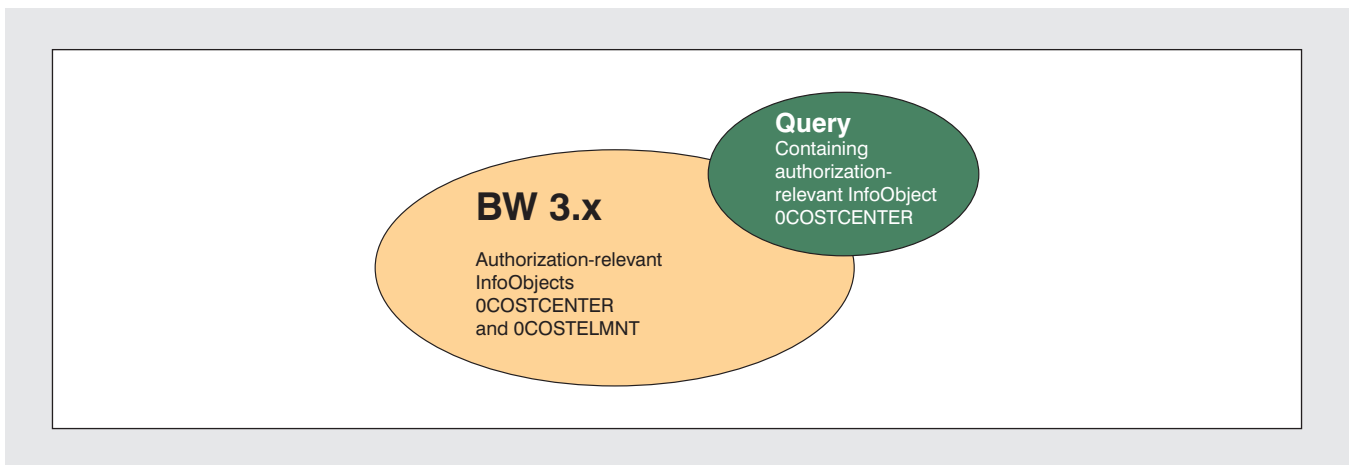


Figure 1 A diagram of the query display status in BW 3.x security

To illustrate the benefits of and rationale for the new security concept provided by BI 7.0, let's use a cost-center authorization example and compare BW 3.x security with BI 7.0 security.

Turning on analysis security authorization globally

In BW 3.x, you turn on analysis security by InfoCube. If you create an authorization InfoObject that restricts access to a specific cost center (e.g., InfoObject 0COSTCENTER) and cost elements (e.g., InfoObject 0COSTELMNT), you usually turn on this authorization for your CO InfoCubes via transaction RSSM (Authorizations for Reporting). At this point, everything is secure.

As your company grows, you typically have to create additional InfoCubes or split the data into two InfoCubes (e.g., one for actual-year data and one for historical data). Then, you need to turn on security for each new InfoObject you create, thus controlling access (in this case, to InfoObject 0COSTCENTER). If you were to forget to turn on security for a newly created InfoCube, you wouldn't have control over who could access it, so authorized and unauthorized users alike could access it without restriction.

In BI 7.0, all InfoObjects are secure from the moment you turn on the authorization-relevant

flag in the InfoObject maintenance dialog; in essence, this makes transaction RSSM obsolete. (*Authorization-relevant* means that the data warehouse designer is expecting restricted access to those InfoObjects.) You cannot fail to turn on security for an individual InfoCube because the new analysis authorization automatically turns it on globally for all InfoProviders, which saves you time and eliminates potential errors.

Note!

BW 3.x *turns on* an authorization while BI 7.0 *maintains* it. This is an important difference. If there is no authorization to maintain but the contained InfoObject is authorization-relevant, no query result is available at all. Security comes first!

Figure 1 shows that based on BW 3.x the InfoObjects 0COSTCENTER and 0COSTELMNT are authorization relevant. You can use the InfoObject 0COSTCENTER in queries, even when you haven't maintained authorization for the InfoObject 0COSTELMNT. You can run the query, which then displays the correct data for the InfoObject 0COSTCENTER, indicated by the smaller green oval

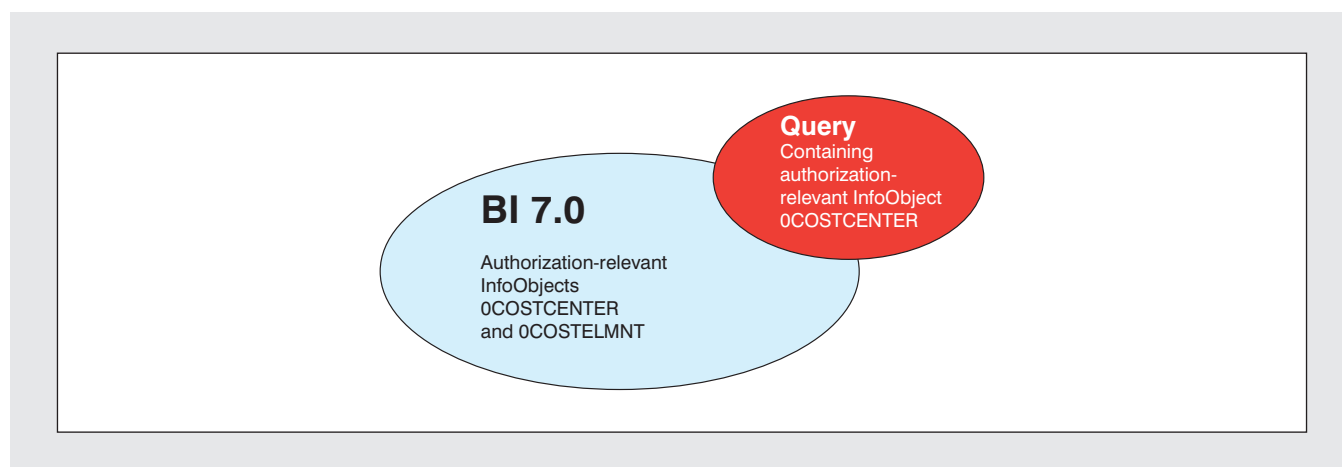


Figure 2 A diagram of the query display status in BI 7.0 security

on the right (status green = OK). The relevant InfoObjects 0COSTCENTER and 0COSTELMNT are available with BW 3.x authorization.

In BW 3.x, you could run a query that contains an InfoObject that isn't maintained, such as 0COSTELMNT, **without a problem**. In BI 7.0, it is mandatory to maintain all the authorization values (restrictions) even if they're not included in a particular query (see **Figure 2**). The query displays no result, not even for 0COSTCENTER, as indicated by the smaller red oval on the right.

If you use only the InfoObject 0COSTCENTER in a BW 3.x query, you must provide authorization-relevant variables in the query by reading the maintained authorization to establish proper access to the data. If the InfoObject is maintained correctly, the query displays its results in MS Excel or Web reporting. This is true even if the InfoObject 0COSTELMNT is authorization relevant; in BW 3.x, the query doesn't take into consideration whether you maintain the InfoObjects you use in this query.

If you run a Human Resources (HR) query to access authorization-relevant HR information about an employee and his or her salary (e.g., InfoObject 0EMPLOYEE), the BW 3.x query doesn't care about any other authorization-relevant InfoObjects, such as 0COSTCENTER. Although the InfoObject 0COSTCENTER has authorization-relevant variables,

the query indicates "No Authorization" in a Web- or MS Excel-based report. Because of the importance of security maintenance in the new BI 7.0 analysis authorization, even unused objects have to be maintained prior to the start of reporting to enable proper reporting and security.

Specifying authorization on attributes without restricting the parent object

Because of the design of security in BW 3.x, there is only one way to hide authorization-relevant attributes, such as the attributes of the SAP-supplied InfoObjects 0EMPLOYEE and 0PERSON. For example, let's say that you need to hide confidential information, such as salary information, from reporting users who are not allowed to see it, but you still need to display non-confidential information, such as the employee's department. As a result, you have to enforce authorization on the complete InfoObject. Instead of restricting or hiding just the sensitive attributes for an employee, you have to provide an authorization or restriction for the InfoObject itself. This means all other information for that employee, such as date of birth, city, or cost center, is not available. In BI 7.0 you hide only the sensitive attributes, such as salary, but the parent object, such as the employee's name, and other attributes remain visible.

As a workaround for BW 3.x, you can create new InfoObjects, ZEMPLOYEE and ZPERSON, by making a copy of the original InfoObjects and then modifying the attributes of the “Z” InfoObjects to restrict the access available from the original SAP-supplied attributes. However, this approach can result in storing redundant information and having twice as many InfoObjects in the BI system.

The difference in the technical names (e.g., OEMPLOYEE and ZEMPLOYEE) also leads to problems when using MultiProviders, because that difference essentially disables the MultiProvider union functionality. For example, you might try to join basic InfoCubes for CO that use ZEMPLOYEE with basic InfoCubes for HR that you designed with OEMPLOYEE. It won’t work. **This is true even if the data contains the same key** because the MultiProviders can only create a union with the exact same InfoObject (e.g., OEMPLOYEE = OEMPLOYEE works, but OEMPLOYEE = ZEMPLOYEE doesn’t). It also doesn’t work if you try to integrate reports for ZEMPLOYEE and basic InfoCubes in HR using MultiProviders. The MultiProviders cannot join data based on ZEMPLOYEE with data based on OEMPLOYEE. With BW 3.x, you had to perform extra work to create reporting that could bypass these types of technical issues. With BI 7.0, you can avoid the need to resolve technical limitations, making your blueprints for authorization and design much easier.

Accessing security from a single point of entry via transaction RSECADMIN

Underlying the new authorization concept is the ability to access basic security and reporting-relevant security from one point of entry without having to run various transactions, such as RSSM, RSSMQ (Start Query with User), and RSSMTRACE (Reporting Log Authorization). With BI 7.0, all you need to provide or assign security is RSECADMIN, which serves as a single point of entry for all needed information. Although the basic RSECADMIN functionality stems from earlier security-related transactions, let’s look at the most important settings for analysis authorization maintenance: creating and managing analysis authorization.

Figure 3 shows the entry screen of transaction RSECADMIN. Here, you create and maintain your analysis authorization InfoObjects using the tools provided on the three tabs that you see: Authorizations, User, and Analysis. (We’ll discuss the tools on each of these tabs later in this article.) The Maint. button is one of the new features provided in BI 7.0.

Now, let’s set up security using a scenario common to many companies.

How to set up security

This is a simple but real scenario for setting up authorizations for BI analysis, as well as other authorizations (to demonstrate non-analysis authorization) for the use of BI analytics and the Data Mining Workbench. In this scenario, we’ll:

- Examine the scenario
- Turn on the authorization-relevant indicator for InfoObjects and individual attributes
- Activate the new analysis authorization InfoObject that BI 7.0 provides
- Implement scenario analysis authorizations
- Test the security

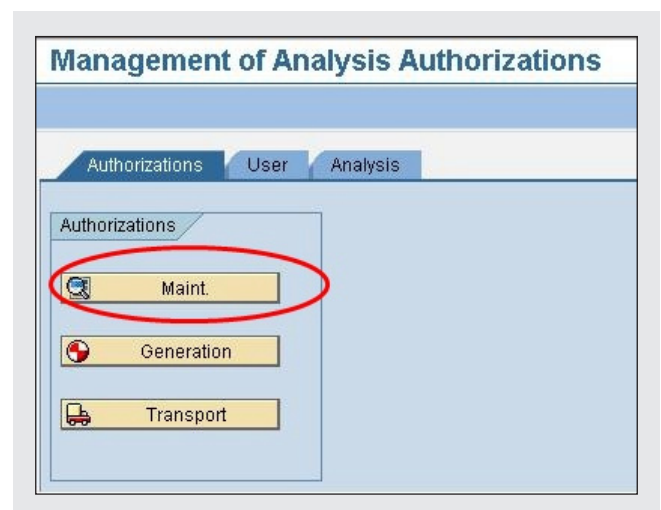


Figure 3 Creating and managing analysis authorizations

Examine the scenario

To illustrate the process, let's assume that we have to create security for two types of reporting users: user group A and user group B. As with any task or project, planning is critical. Consider the following descriptions of each group. In both cases, reporting is not limited to employee data, and salary information is restricted.

User group A (display-only users) includes those users who:

- Generate reports
- Need to connect to a specific area of reporting (e.g., SD)
- Create or modify queries
- Save query views using spreadsheets (i.e., MS Excel workbooks)
- Don't need access to administrative roles or any BI transactions

User group B (reporting and power users) includes those users who:

- Need to connect to a specific area of reporting (e.g., SD)
- Create, delete, and modify queries for all activities
- Save query views using spreadsheets and Web templates
- Need access to the following BI transactions:
 - RSCRM_BAPI (Extract Query Result to File) to provide query results for non-SAP analysis
 - RSANWB (Model the Analysis Process) to design SD-specific requests for data mining
 - RSDMWB (Customer Behavior Modeling) Data Mining Workbench to design weighted score-cards, ABC analysis processes, "what if" predictions, and customer target groups for marketing or sales campaigns

Next, you need to know where to turn on analysis security for these user groups.

Note!

You can adapt the next section to your needs if you want to build your own security for BI 7.0.


Turn on the authorization-relevant indicator

BI 7.0 enables you to restrict access not only to a complete InfoObject but also to individual InfoObject attributes. Before you can apply or use any analysis authorization, however, you must identify which authorization-relevant InfoObject you need and turn on its authorization.

Before you run transaction RSEADMIN to assign security, you have to turn on the individual InfoObject's authorization-relevant attribute by selecting the applicable InfoObject (in this case, the 0EMPLOYEE InfoObject), as shown in **Figure 4**. To modify the InfoObject or its attribute, switch to change mode by right-clicking on the InfoObject and selecting "Change" from the context menu.

Turn on the authorization-relevant indicator for a parent InfoObject

For this scenario, instead of using the parent InfoObject, such as 0EMPLOYEE, you'll use an attribute.

To enable authorization-relevant for the complete InfoObject including its attributes, switch to the Business Explorer tab in InfoObject maintenance, and then select the AuthorizationRelevant checkbox. Finally, activate the object either by selecting the  button from the toolbar or by pressing Ctrl-F3.

By turning on the authorization-relevant indicator for a parent InfoObject, you automatically restrict access to this InfoObject and all of its attributes. From now on, you have to provide proper authorization to those users who would otherwise not have access (e.g., a user who needs access to an employee record).

The screenshot shows the configuration interface for the InfoObject '0EMPLOYEE'. The 'General' tab is selected. In the 'General settings' section, the 'AuthorizationRelevant' checkbox is highlighted with a red circle. Other settings include 'Display' set to 'Key and Text', 'Text Type' set to 'Default', 'BEx description' set to 'Short description', 'Selection' set to 'No Selection Restriction', 'Query Def. Filter Value Selection' set to 'Values in Master Data Table', 'Query Execution Filter Val. Selectn' set to 'Only Posted Values for Navigation', 'Filter Value Repr. At Query Exec.' set to 'Selector Box Without Values', 'Base Unit of Measure' is empty, 'Units of Measure for Char.' is empty, and 'Currency attribute' is empty. The 'BEx Map' section shows 'Geographical type' set to 'No geo-characteristic' and 'Geographical attribute' is empty. At the bottom, there are buttons for 'Upload Shapefiles', 'Edit shape files', 'Geo Data Download (Everything)', and 'GeoData Download (Delta)'.

Figure 4 Turning on the AuthorizationRelevant indicator for a parent InfoObject, not an attribute

Note!

Remember, turning on authorization-relevant (i.e., activating the InfoObject) should be the last step in a BI project. If you turn it on immediately, you restrict testing for other members of the development team. Team members implementing authorizations might stop the process of creating a state-of-the-art report for other members. Turning on authorizations requires authorization maintenance immediately, before seeing any report result. After turning on the authorization-relevant indicator, queries only display data after proper authorization value maintenance. Therefore, first finish your design with pre-testing before you turn on authorization so that you won't delay the design process.

Restrict access to individual attributes

In this example, you need to restrict access to the individual attributes of the 0EMPLOYEE InfoObject. You can restrict specific users' access to attributes,

such as Pay Grade Level, by checking the box under the AuthorizRelevant column for each attribute (see **Figure 5** on the next page). The users in user group A need to see all attributes of the InfoObject except the Pay Grade Level attribute (0SALARYLV).

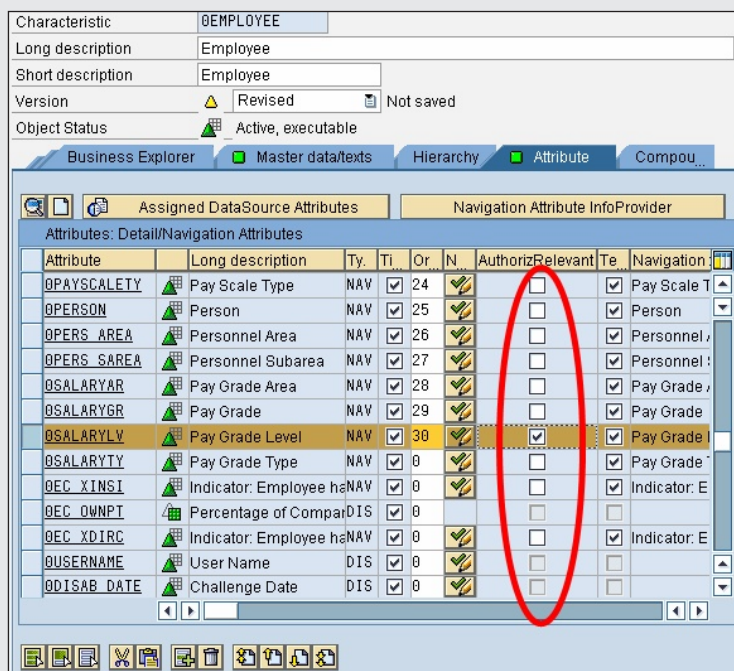


Figure 5 Restricting access to one attribute, Pay Grade Level, of the InfoObject

As you can see from this discussion, it is fairly easy to grant access to various different parts of the data; however, I would recommend that you consider the following guidelines when you do so:

- Be strict about turning on authorizations, and don't skip the details. I often find authorization based on directly visible InfoObjects (e.g., cost centers or employees), but designers sometimes completely forget to take account of the authorization of individual attributes, such as salary.
- Start implementing BI 7.0 analysis authorization.
- Add some other new InfoObjects that you want to include in your BI 7.0 analysis authorization.

Activate the new analysis authorization InfoObject that BI 7.0 provides

BI 7.0 presents a conceptual change in authorization (i.e., if authorization-relevant is turned on in one InfoObject, it's mandatory to maintain authorization values for all InfoObjects). As a result, SAP now delivers some new authorization InfoObjects (OTCAACTVT, OTCAIFAREA, OTCAIPROV, OTCAVALID, and OTCAKYFNM) without activating the authorization-relevant indicator (see **Figure 6**). To use any of these objects, you select the object from BI content, and activate it as you would any other InfoObject.

Before you can use these InfoObjects, you must turn on the authorization-relevant indicator for each InfoObject. (For the cost center example in this article, you need to turn on the authorization-relevant indicator for the first four objects.) After activating the objects to be maintained, you can use them in authorizations. Be aware of these SAP-delivered

InfoObject	Function	Description
OTCAACTVT	Activity in analysis authorizations	Restricts the access to access codes, such as 01 = Create, 03 = Display, and 16 = Execute
OTCAIFAREA	InfoArea for analysis authorizations	Restricts the access to specific InfoAreas, such as CO or HR
OTCAIPROV	Authorizations for InfoProvider	Restricts the access to specific InfoProviders such as InfoCube ABC and XYZ
OTCAVALID	Validity of an authorization	Restricts the validity of an authorization. If you don't use OTCAVALID, the authorization won't be limited to a date. If you use the option with an asterisk indicating "unlimited," it also won't be limited to a date. I recommend you use the OTCAVALID item and if unlimited, maintain the *. If limited, use a specific date when the authorization will expire. As of 10/31/2008, this option won't work anymore.
OTCAKYFNM	Key figure in analysis authorizations	Restricts the access to specific key figures, such as key figure 0AMOUNT (amounts such as net sales in dollars) so that reporting users only see the key figures that they have permission to see in the reports — for example, key figure 0QUANTITY (Everyone can see quantity but not all reporting users can see the amount in dollars.)

Figure 6 New authorization InfoObjects from SAP in BI 7.0

objects. If you haven't activated a particular Info-Object, your users might not be able to generate the reports that they need.

You also have to include the restriction settings for these InfoObjects. You must assign these objects to each user with at least one authorization. Query processing no longer checks the pre-BI 7.0 authorization objects, such as S_RS_ICUBE, S_RS_MPRO, S_RS_ISET, and S_RS_ODSO.

Implement scenario analysis authorizations

You need to implement both user groups in this example. Let's begin by setting the security for user group A. Creating a security role for a user group requires the following four steps:

1. Run transaction RSEADMIN to create an analysis authorization InfoObject, and then click on the Maint. button (**Figure 3**).
2. Enter a technical name for authorization in the

dialog that appears, and then click on the Create button (see **Figure 7**).

This dialog provides a description for the authorization that you want to create. In the example, ZAU_SDUSER is the technical name and it has three

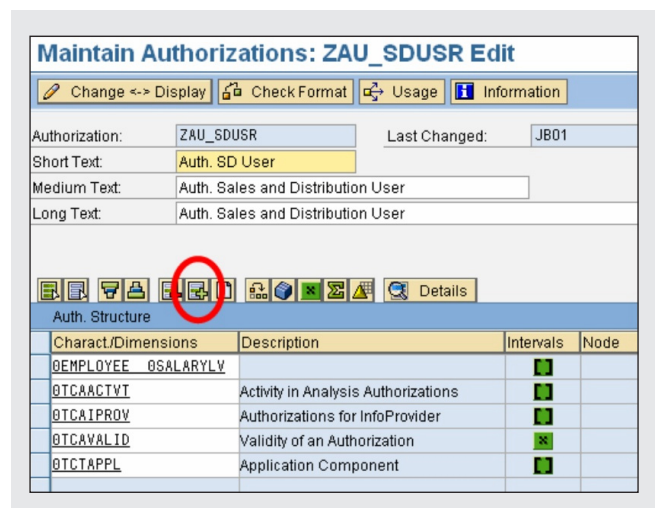


Figure 7 Maintaining authorizations for the SD user

variations of the same description (i.e., Authorized Sales and Distribution User): Short Text, Medium Text, and Long Text. In the lower portion of the screen, you add the following objects to the authorization:

- 0TCAACTVT: Grants the level of access to an activity, such as read only (03) and create (01)
 - 0TCAIPROV: Gives authorization to particular InfoProviders
 - 0TCAVALID: Confers authorization to specific time periods
 - 0TCTAPPL: Enables access rights to specific application components
3. Click on the 0TCAACTVT row, and then click on the Details button above it.

Now, you can modify the default values for 0TCAACTVT's attributes. You want to restrict access to the authorization activities (which access your queries in reporting). 0TCAACTVT is an authorization-relevant field in the BI system, so you need to provide an authorization or an asterisk (*) to grant authorization to all values.

As shown in **Figure 8**, you enter single values, such as "03," to enable the display of the attribute

(e.g., the value of this attribute is read-only) and "16" to execute the activity. Maintaining the details of the attributes or of an entire InfoObject grants or restricts the desired access.

In the detail menu, you use operators such as "EQ" (equal) to specify single values, "BT" (between) to enter ranges into the "from" and "to" input fields, and "CP" (contains pattern) to allow single values or ranges (e.g., "abc*" means every value that starts with "abc"). You maintain the Pay Grade Level attribute (e.g., 0EMPLOYEE_0SALARYLV) and all other authorization-relevant InfoObjects in the same way.

	Technical Character. (from)	Technical Character. Value (to)
I EQ	03	
I EQ	16	

Figure 8 Maintaining the authorization details (permissions and restrictions)

Tip!

Sometimes, certain key figures (e.g., sales revenue) need to be authorization-relevant. For example, if you have a report that displays sales revenue, you might not want the reporting user to have access to the profit margin. In this case, you can grant authorization to specific fields by using the SAP-provided authorization InfoObject 0TCAKYFNM that allows access only to the sales revenue. The system won't display any other key figures requested in the query unless you maintain them by name.

Figure 9 Creating and managing user role assignments

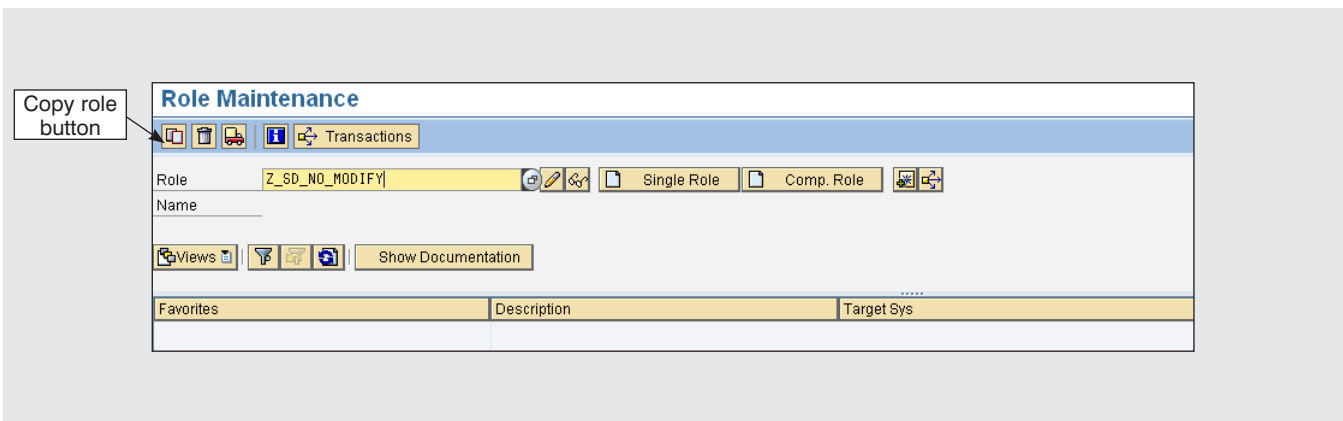


Figure 10 Creating a new role

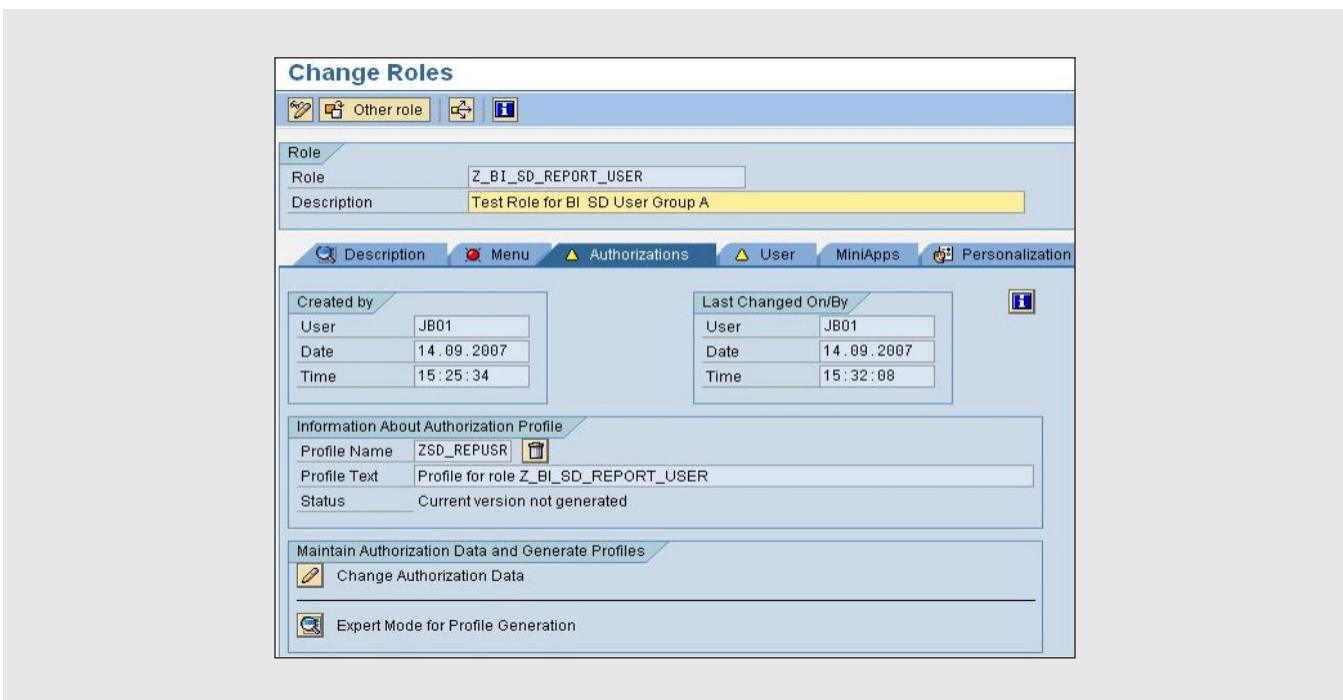


Figure 11 Maintaining a role

4. Before you can assign authorization to an existing user, you must provide the user with a specific role. If necessary, run transaction RSECADMIN and select the User tab (see **Figure 9**). Then, click on the Role Maint. button, which displays the Role Maintenance screen shown in **Figure 10**. This same screen displays when you run transaction PFCG (Profile Generator) in either BW 3.x or BI 7.0.

Select the Single Role button, enter a name and description for the role, and then switch to the Authorizations tab, as shown in **Figure 11**. The name and description of the role that you entered on the User tab appear in the upper portion of the screen.

Now, you can modify the authorization values that a user in this role (i.e., Z_BI_SD_REPORT_USER) can

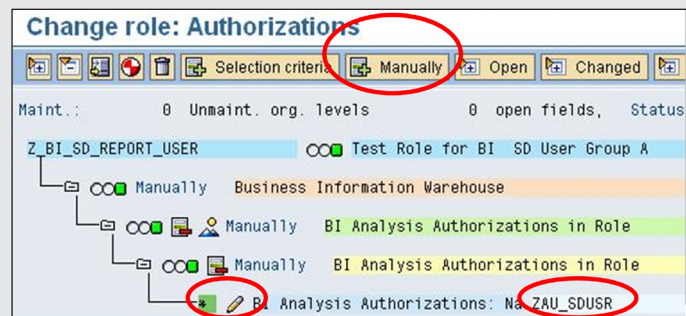


Figure 12 Manually creating analysis authorization object ZAU_SDUSR

access. Let's begin with the authorization object ZAU_SDUSR, which grants access to all authorization values that the system currently maintains. Click on the Change Authorization Data button, and then click on the Manually button.

In the screen that appears, enter the name of the authorization object S_RS_AUTH, and then click on OK. The result enables you to maintain the authorization values for authorization object S_RS_AUTH. This authorization grants access to all the authorization values currently maintained for activities (0TCAACTVT).

To connect the automatic retrieval from the maintained analysis authorization (which stores the maintained values) to the authorization InfoObject (read authorizations), click on the change icon (the pencil), and enter the name of the analysis authorization (ZAU_SDUSR). See **Figure 12**.

You can also add other authorizations to allow users to distribute their reports to email recipients or the printer. You must maintain the list of authorization objects shown in **Figure 13**.

Click on the Manually button, type in the authorization InfoObject name (e.g., ZAU_SDUSR), and maintain restrictions or activities. (Use the descriptions next to the InfoObject names to locate the relevant information, i.e., Display, Execute, etc.) Let's take a look at the authorizations you can set:

- **S_RS_BCS (SAP Business Explorer, or BEx, Broadcasting Authorization to Schedule)** enables users to distribute their reports to other users and themselves. As shown in the example, you can set S_RS_BCS to all activities and full authorization. Don't allow users to distribute only specific reports; rather, consider restricting confidential reports from being automatically distributed by entering only the usable BI report IDs.
- **S_RS_BTM (BEx Web Templates)** enables users to run queries, such as the following, in a Web environment: single reports or complex reports that you integrate with your SAP NetWeaver Portal and that Internet Explorer accesses. You can set this option to Display and Execute so that the users of this role cannot change a WebReport or its design. They can only be report consumers.
- **S_RS_COMP (Components)** creates queries, structures, variables, and any query components. Here you can grant users the rights to execute queries and to view them, so users can't create queries, structures, variables, or any query component unless they have the appropriate rights. To create a power user (a user who can maintain, create, delete, or modify existing queries), you must grant the user all rights to queries, structures, variables, and any query components. Authorization is granted to queries, query views, and Display and Execute activities.

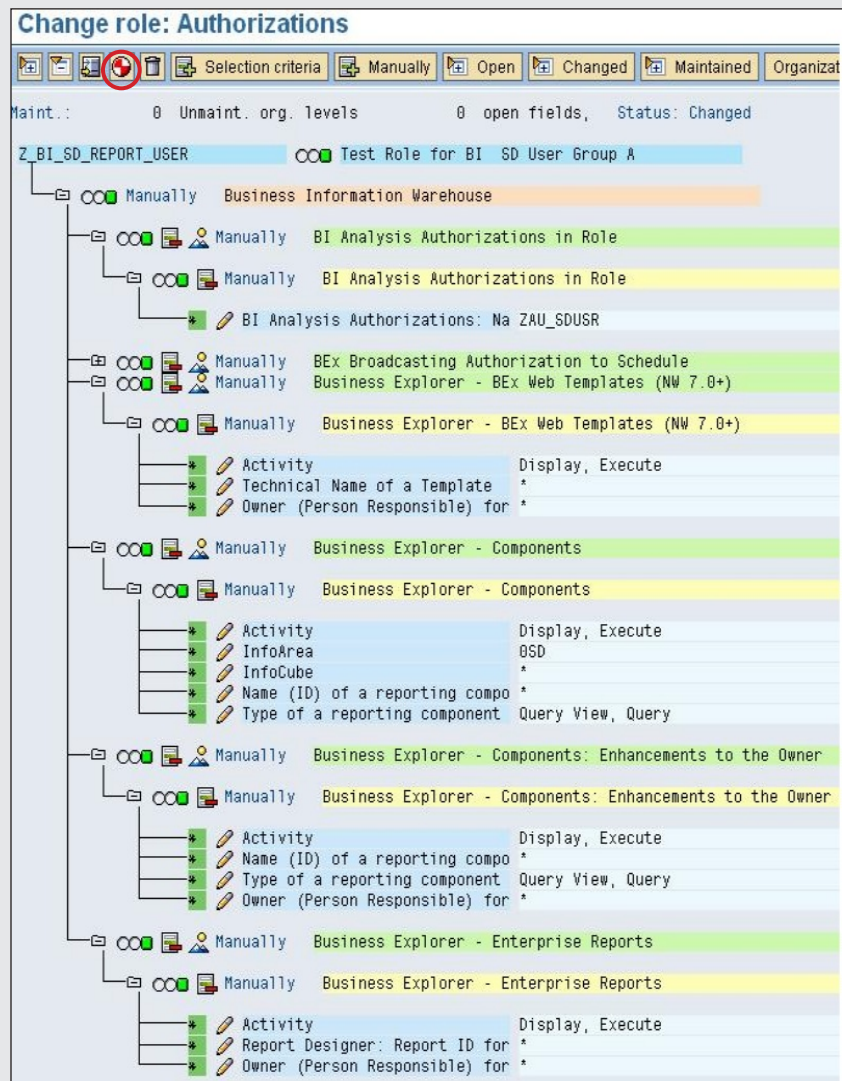


Figure 13 Setting up role authorizations for user group A

- S_RS_COMP1 (Components: Enhancements to the Owner) enables users only to execute, display, or change (depending on your settings) queries or views that a specific user created. (The person who creates a report is defined as the owner in SAP terms.) You can enable the same settings as S_RS_COMP, but you must enter the “all access” indicator (*) in the owner (person responsible) field. You can allow access to only the particular

user who can change the query, or you can assign the role to all users and restrict a specific user from changing the query. Using this setting you can separate the activities maintained in S_RS_COMP (for all users) from those of the query owner in S_RS_COMP1. For example, I can’t change a query that my colleague created, but I can change my own queries if I have permission to write them.

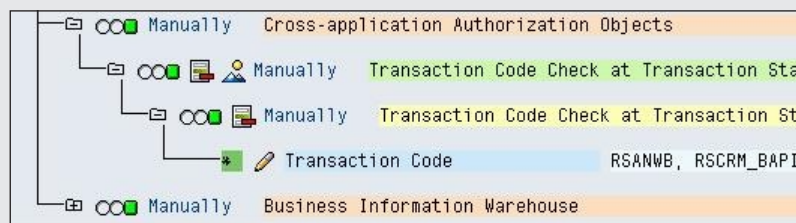


Figure 14 Adding the functionality to call specific transactions to the role of user group B

- **S_RS_ERPT** (Enterprise Reports) grants users privileges to run or create reports.³ For example, you can set this option to Display and Execute only, so the user can view and run only those reports that have been created with SAP Report Designer. The user in user group A needs to be able to execute the report with display-only access.

After you finish maintaining the authorizations, you need to generate a profile for user group A by selecting the Generate button or by pressing Shift-F5. The system prompts you for a profile name. (SAP generates the default name T-Pxxx.) Enter a name that adheres to your company's naming conventions, or use the default name and maintain a description for this profile, for example, "Profile for Role SD User (No Modify)."

Now, let's create user group B. To create the role for user group B, you can make a copy of the role you created for user group A. To copy a role, go back to the initial screen of Role Maintenance and select the Copy role button (**Figure 10**) to the left of the trash can or press Shift-F11. Then, provide a name and description for the role copy. The only difference between creating user group A and user group B is that you want to provide user group B with access to different transactions and add the ability to call specific transactions, as shown in **Figure 14**.

In addition to the authorization objects for user group A, let's add the following three authorization objects to grant rights for deep data analysis to user

group B via the **Analysis Process Designer (APD)**, a graphical tool that uses data-mining functionality from within BI to analyze your data. (These objects are just examples; they could be any transactions.)

- **S_TCODE**: This grants authorization to all the transactions that a user can call. You might provide access to transaction **RSANWB**, the analytical workbench of the APD, and **RSCRM_BAPI**, a transaction used to extract query data to comma-separated value (CSV) files or SAP tables.
- **RSANPR**: This gives authorization for a user to call analysis processes in the APD. For a deeper look at the functionality of the APD, see *BW/BI Expert* online knowledgebase at <http://www.BW-BIexpertOnline.com> and SAP Portal Help at <http://help.sap.com>.
- **RSDMEMODEL**: This confers authorization on a user to create and use data-mining models, such as target groups. For example, with the help of this data-mining model, you can create new characteristic values, such as ABC customer groups, by calculating revenues and automati-

Note!

I didn't restrict the last two authorizations, **RSANPR** and **RSDMEMODEL**, but you should grant or restrict rights to a specific functionality for a specific user.

³ BI 7.0 also delivers the new SAP Report Designer, a discussion of which is beyond the scope of this article.

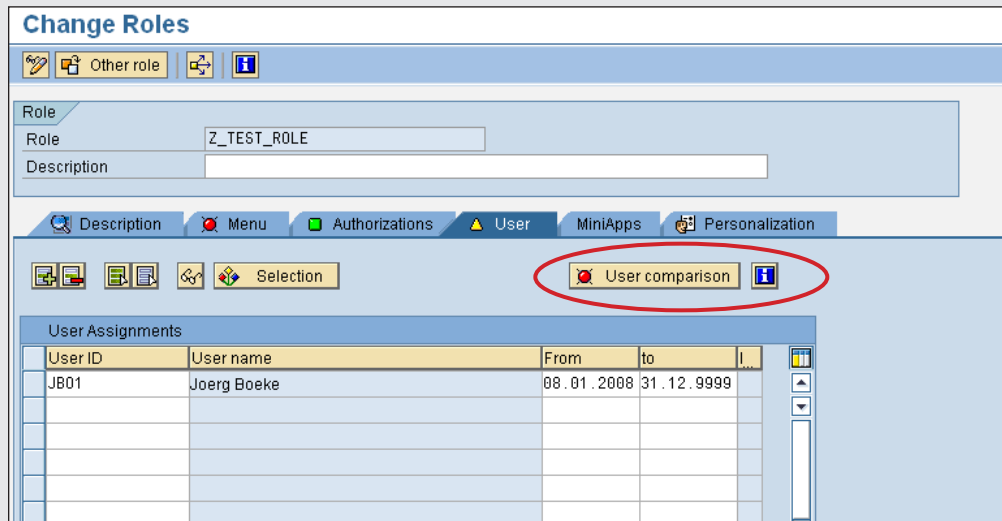


Figure 15 User assignment in Role maintenance

cally populating the attributes within BI. (SAP Portal Help provides a good source for this scenario as well.)


The three authorizations above show you the importance of a global view of authorization. All three of these allow users to see data that might be sensitive even if you properly maintain the analysis authorization responsible for any reporting access. With the first transaction, `S_TCODE`, users might be able to directly access tables of InfoCubes if you don't restrict them. With the help of the APD and data-mining functionality, users can access data directly without needing to query and manipulate the data internally in BI. When using authorizations, it's a good idea to think about what the users need and then restrict that functionality to only those with proper access.

After you generate an authorization profile and save the role for user group B, you can assign users to them on the Role maintenance screen, User tab (as shown in **Figure 15**) and after the alignment via

Note!

I recommend that you always assign a test user (copied from a real user) and test your authorizations before assigning them to real users. You might have overlooked something important that you can find in testing. (See the explanation in the following section, "Test the security.") All the examples in this article describe changes in the development system only. I strictly do not recommend changing the live production environment.

the User comparison button. It automatically adds the role to the users assigned (J001). Use of this button aligns (or synchronizes) the authorization role with the user parameters. The system populates the user data with this new authorization value and distributes it to all internal tables. From then on, the selected user will

only be able to perform functions dependent on the authorizations just created. You save the role by clicking on the Save button .

You can also accomplish user assignment in transaction RSECADMIN (**Figure 9**) by clicking on the User Maint. button. It takes you to user maintenance — the same as transaction SU01 (User Maintenance) does — to assign roles to your selected users. After user assignment, users can run reports and their functionalities securely.

Test the security

To make sure security works as you expect, you must test it in depth. If you restrict a user from changing a query in the SD area (as in user group A), you should test to make sure that the user really cannot change it. This user might have other rights in other areas, such as HR, and perhaps he or she can create queries in that area through a different role. You should make sure that this authorization still works too and doesn't interfere with your role or profile. With authorization, you can assign a user multiple roles.

If a user has no restrictions (e.g., for transactions in one role), security ignores any role restrictions that you might have created. With SAP, the highest authorization (i.e., the least-limited access) wins. To test the role, you can use transaction RSECADMIN and click on its “Execution as” functionality (see **Figure 16**).

Enter the user ID you want to test (e.g., JBO_TEST), and check the With Log checkbox to trace execution and security for that user (see **Figure 17**).

The default setting is RSRT (Select an Existing Query), which allows you to select a query in RSRT query testing and execute it with the selected user ID. Click on Start Transaction. Then, you click on the Query drop-down list, as shown in **Figure 18**, and select the query that you will use for authorization testing.

The test query, ZSDM_C03/BI_PORTAL_2007_MP_Q01, displays customer data. You can also select how you want to display the query results (in this case, as a list). Click on the Execute button to display your

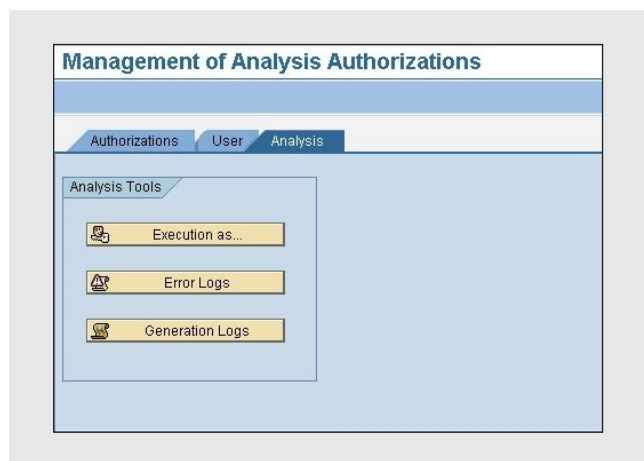


Figure 16 Testing analysis security with transaction RSECADMIN

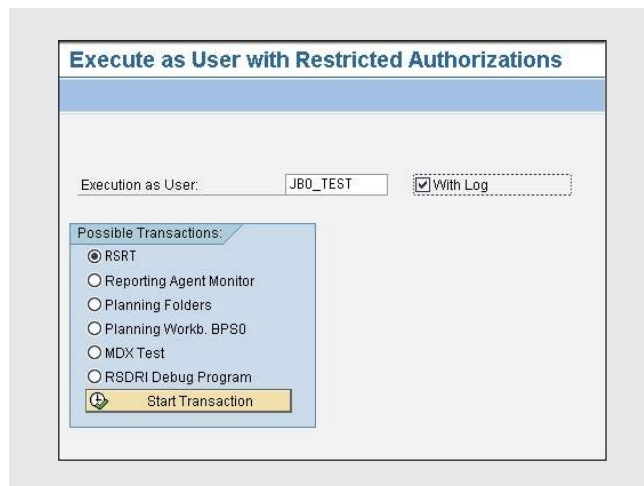


Figure 17 Executing the query with different users for testing purposes

query's data as a static data screen. If you choose the BEx display option from the drop-down query display, the data will display in a format similar to BEx, which means context menus will work like they do in BEx Excel Analyzer. You can navigate via context menus (right-click on the menu that BI delivers), so you can test your authorizations even during navigation. This might be a good test when restricting drill-down menus to certain levels of data. The third option is HTML, which allows you to interact with your query as if it were a regular Web query. So, you can test navigation buttons as well as context menus.

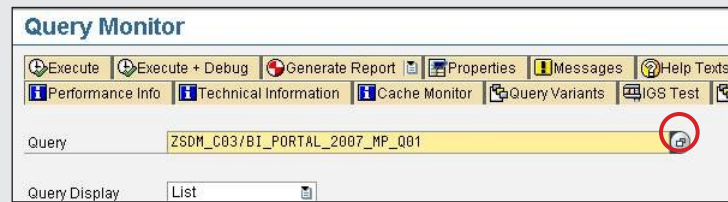


Figure 18 Monitoring query execution and selecting results format

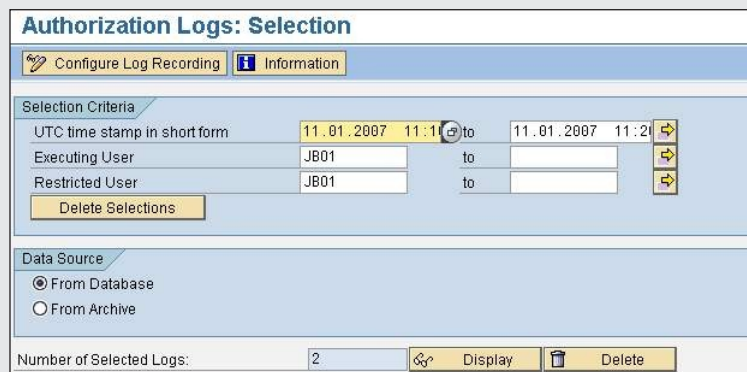


Figure 19 Accessing the authorization error logs

After you execute the query, you either see the expected results or a No Authorization message. This depends on whether you test with query values that you've allowed the user to see, or with restricted query values that you haven't allowed the user to see. To leave the query testing and to access the execution and security trace that you enabled with the With Log option, click on the back button (not shown in the figure), leaving the report displayed.

Now, you can begin to analyze whether your authorization worked properly or, if you received "No Authorization," what caused the problem. To see any errors logged during the query's execution, click on the Error Logs button (**Figure 16**) to access the execution and security trace file dialog (see **Figure 19**).

You can access the trace logs from the previous query authorization test. If you run the test repeatedly,

you will have the same number of logs as tests run. Click on the Display button at the bottom of the screen to see the trace log that is provided as an HTML output. The details in the trace log enable the user to investigate if there are authorization problems.

Figure 20 on the next page displays the authorization error log. In the example, there weren't any errors. By turning on Presentation Options and clicking on the Update Display button, the system log goes into even more detail (circled in the figure). Based on this level of detail, if there's a problem, you can see whether the system turned on authorization for a specific attribute. BI covers even the authorization trace and analysis. This is how you would regularly set up authorizations for analysis and additional authorizations for the analytical functionality within BI.

Display Error Log

Print Document Save Document Update Display

Presentation Options

Input Help and Variables

- ☒ Attribute Authorizations
- ☒ Value Authorizations
- ☒ Node Authorizations

Check Components

- ☒ Relevant InfoObjects
- ☒ InfoProvider Checks
- ☒ Authorization Check

Optimizations

- ☐ Buffering of Authorization Data

Authorization Check Log

Date and Execution Time (Local Server)
 Execution Date: 08.01.2008
 Execution Time: 11:31:39
 Executed Query: YTC_WHM/JBO1_BIPORTAL08_WHM_Q02
 Transaction RSRT (BW - output test)
 Executed by User JBO1
 Executed with Analysis Authorizations of Another User MMO

InfoProvider Check

Building the Buffer...
 ...Buffer Built
 Are there authorizations for accessing InfoProvider YTC_WHM with activity 03?
 Authorization exists for general access to InfoProvider YTC_WHM with activity 03 ✓

Relevant Characteristics for Detailed Authorization Check
 (Characteristics with Full Authorization Are Not Listed!)
 List of Effective Authorization-Relevant Characteristics for InfoProvider YTC_WHM:
 List Is Empty.
 There Are No Characteristics That Have to Be Checked in Detail

Attribute Authorizations
 Attribute Authorization for Characteristic 0CALMONTH
 Following Attributes Exist

CHABASNM	OBJVERS	ATTRNM	POSIT	ATTRTP	ATTRIMFL	F4ORDER
0CALMONTH	A	0CALMONTH2	0002	DIS	0	02
0CALMONTH	A	0CALYEAR	0001	DIS	0	01
0CALMONTH	A	0DATEFROM	0005	DIS	0	05
0CALMONTH	A	0DATETO	0006	DIS	0	06
0CALMONTH	A	0NUMDAY	0003	DIS	0	03

Figure 20 Display of authorization trace (log file)

Authorization implementation is not a single-step process. When you have to provide authorization values for multiple customer-based or SAP-based objects, more users with more individual restrictions force you to create multiple authorizations. You must maintain the specific authorization-relevant informa-

tion for each InfoObject to restrict the object so that it shows only the desired value.

Be aware that the example specified the settings for only one user role (**Figure 7**). Implementing authorizations can be a time-consuming job unless you automate the population of user profiles.

Human Resources	0HCM	Cha
Organizational Management	0PAOS	Cha
Alternative Position	0ALP_POSTN	= Cha
Position	0HRPOSITION	= Cha
DSO (Sysko) for Authorization Add-On	/BA1/AUTH_1	= Mar
HR Structural Authorizations - Values	0PA_DS02	= Mar
HR Structural Authorizations - Hierarchy	0PA_DS03	= Mar
DSO (Sysko) for Authorization Add-On	AUTH_2	= Mar

Figure 21 SAP content InfoProvider storing OLTP HR authorization values

Automating the user profile population

Populating specific authorization values or restrictions for individual users (e.g., user A is only allowed to see cost centers 100 and 300 and user B is only allowed to see cost center 200) can be time-consuming. With a global security concept, it is especially hard to keep authorization changes that might take part in the ERP system up to date with your BI environment. Using BI 7.0, if a user loses the right to see a particular cost center in the ERP system, you can align this change automatically with BI without needing any further manual changes in the BI roles.

The SAP way

How do you use the authorization extraction, warehouse objects, and profile-generation content shown in **Figure 21**? First, you must activate the InfoObjects from SAP content via Administrator Workbench, as usual. Using the plug-in for SAP Online Transaction Processing (OLTP) systems, you have already installed in your ERP system access to SD or CO data from some extractors in the FI, CO, and HR areas for authorization values. These extractors retrieve maintained authorization information from your ERP system (user A is only allowed to see cost centers 100 and 300) and return this information to BI during the extraction process. SAP also delivers the storage objects in BI to store and use the authorization values.

BI 7.0 provides the so-called DataStore Objects (DSOs — formerly Operational Data Store or ODS) for storing authorization data. For example, BI 7.0 will extract authorization for the HR area (as in the example case) and store it in these DSOs:

- DSO 0PA_DS02: This InfoObject stores the characteristic “flat” values, such as cost center = EMEA, and cost centers between 100 and 300. All entries here are the basic values maintained in SAP ERP authorizations.
- DSO 0PA_DS03: This InfoObject stores the hierarchy values (such as cost element node = “Energy costs”), including all subnodes (such as electricity, heating, and gas). If you use hierarchy authorizations to access specific levels of information, the system extracts those SAP ERP entries and stores them in this InfoObject.

SAP content includes authorization DSOs for the CO area (see **Figure 22**), as well as other areas. To activate those objects or look up your specific objects,

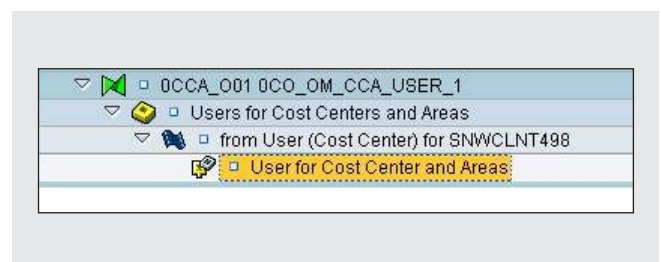


Figure 22 Activating an authorization DSO in the CO area

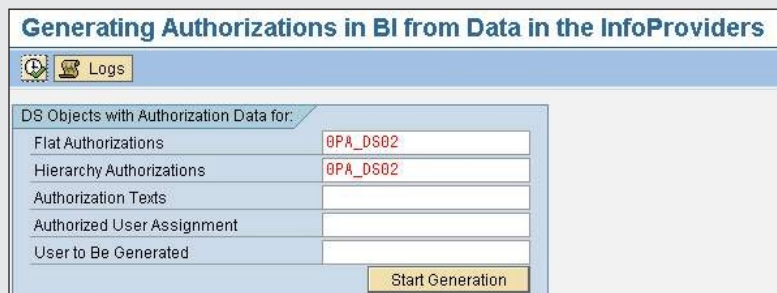


Figure 23 Displaying the dialog to automatically generate profiles from extracted authorization values

use the standard content in Administrator Workbench and search within the DSOs for “authorization.”

For example, after activating the standard HR Structural Authorizations content using Administrator Workbench, you can apply the SAP-delivered extractors in BI to load authorizations from ERP into BI automatically. Instead of manually checking the differences between ERP and BI restrictions in data access and manually aligning the analysis authorizations (**Figure 7**), you can generate authorization profiles for each type of user and then assign the profiles automatically to your BI users.

You can also generate authorization profiles with transaction RSEADMIN. All topics use RSEADMIN somehow, but you can use it stand-alone as well as in a migration. Users who already have security might consider using this automation. Also a user could have a new system and thus doesn’t need to migrate. Keep in mind, however, that this article is not specific to upgrades or new BI 7.0 users.

To generate the authorization profiles, click on Generation (**Figure 3**) from SAP’s DSOs or your user’s DSOs. The system prompts you to enter the InfoObjects for authorization generation.

Next, you can divide your selection for flat-authorization (i.e., single or range values only) DSOs containing extracted flat values, such as cost center = “1234.” You can generate the values for hierarchies, such as HierNode = Cost element “Energy costs,”

either in parallel or sequentially. Click on the Start Generation button (as shown in **Figure 23**), and the system generates profiles for characteristics and hierarchies for selected DSOs for all or specified users.

In this case, when using 0PA_DS02 for flat authorizations and clicking the Start Generation button, BI functionality (in terms of function modules and reports) reads the authorization values for each individual user from the DSOs. BI 7.0 automatically creates a profile of it in the background and then assigns this profile to the user without further manual interaction.

Note!

A “HierNode,” or hierarchy node, is either a subnode or a parent node. For example, if you think of a family hierarchy, a HierNode can be a parent or grandparent. The “flat” values underneath your parents are Mom and Dad. By selecting hierarchy nodes, you can be flexible to structural changes. In this example, if your parents divorce, the item Dad or Mom might get a new value, such as Stepmother or Stepfather, but when accessing the parent HierNode, you still access this level of relatives.

In case of restriction changes in SAP ERP, these changes come into the DSO, and when using process chains to create automatic workflows, can automatically realign the specific profiles without any manual interference. Instead of requiring you to access the profile manually via RSEADMIN, maintain its values, regenerate it, and save it, SAP technology does the work for you.

This process, depending on the number of different aspects in your authorizations, can be time consuming. For example, if your company is large and you need to create hundreds of roles, processing these roles require DSO populating and generating new changed profiles that will take anywhere from several minutes to many hours. This becomes a project that should be done over a weekend if it takes too long. I recommend testing this timing in depth.

These automatically generated profiles are not readable in terms of self-explaining profiles and names (see **Figure 24**). No one can see which user is restricted by what value just by checking the generated profile, which receives an automatically generated technical name but no description.

Although time intensive, this automated process is less labor intensive because it dynamically extracts data from the OLTP system and adopts all the changes automatically in BI. Also, if a user loses authorization to a specific employee in the HR area of OLTP, he or she loses this authorization in BI right after the system updates the data and generates the profiles.



Restrictions	
BI Authoriz.	Short Description
RSR_00006492	
RSR_00006493	
RSR_00006494	

Figure 24 Automatically generated profiles

The enhanced way

A few lines ago, I explained how to use SAP authorization extraction, populating the SAP content DSOs and the automatically generated profiles. In large enterprises, this generation can create thousands of profiles and a heavy load on the system, which might not fit into your time-frame for uploading data and changes during the day.

Tip!

With BI 7.0, it's possible to turn regular variables into authorization variables, so you should use authorization user exit variables whenever possible.

In BW 3.x, you have to implement authorization variables into each single query to fulfill authorization purposes. BI 7.0 makes your life dramatically easier. You don't need to enter the variables into the query because the system automatically checks out the authorization in the background.

Let's use the variable approach to ease the use of authorization. You create a variable that grabs its value from the authorization DSOs. This is an easy-to-use and flexible approach. You'll save time by scaling down from thousands of profiles to a handful that just cover the access to various transactions but no longer cover the access to authorization-relevant InfoObjects and their values.

The first thing you have to do is create a Customer Exit variable for each authorization-relevant object, as shown in **Figure 25** on the next page. In the excursion for my imaginary company, I use the following objects to restrict the Organizational Unit by flat (in this case, "1234") and HierNode, as well as by specific employees. The Technical Name (Variable Name)

InfoObject	Variable name
00ORGUNIT (Flat)	PA_AFORG
00ORGUNIT (HierNodes)	PA_AHORG
0EMPLOYEE (Flat)	PA_AFEMP

Figure 25 Customer Exit variables for authorization-relevant InfoObjects

I use refers to the HR area (e.g., PA) but you can use any technical name you like, following your company's naming convention for variables.

Create a new variable and maintain its properties by using the new Query Builder. Define a new variable for the authorization-relevant object (see **Figure 26**) by right-clicking the Characteristic value variables folder while in Query Designer. Then, maintain the properties, such as Description and Technical Name, and select the Customer exit option in the Processing By field, as shown in **Figure 27**.

In the Details tab, I recommend that you set the Variable Is field to Mandatory (see **Figure 28**). That way, if for any reason no values are maintained, the system sends a warning instead of displaying everything. Make sure that the box "Variable is Ready for Input" is *not* checked; you don't want to see the variable window pop up; you want it to run automatically in the background instead.

If you need to select some of the authorization values from the customer exit, you might create a new variable that uses new BI 7.0 functionality to derive

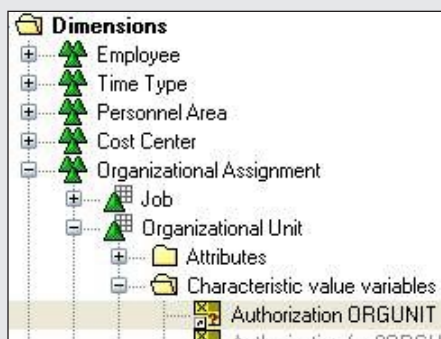


Figure 26 Creating a new query OLAP variable



Figure 27 Creating a variable-type customer exit

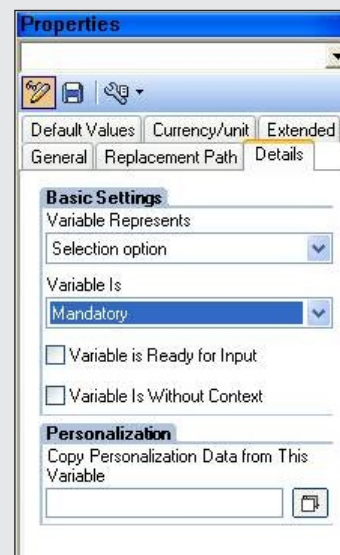


Figure 28 Maintaining variable properties

values from variables by using Replacement Path options.⁴ After finishing and saving the variable in Query Builder, you need to set up the customer exit.

⁴ A discussion of Replacement Path options is beyond the scope of this article.

Setting up the customer exit

To begin, enter transaction CMOD (Project Management of SAP Enhancements), and enter a name for your exit enhancement project, for example, BW_EXIT, as shown in **Figure 29**.

If you have already used the exit for variables, you might have an existing project. Otherwise, you can use a new name that corresponds with your company's naming conventions. Then, click on Display to see the variables, as shown in **Figure 30**.

In addition, you have to maintain the code in ZXRSRU01, as shown in **Figure 31**, and EXIT_SAPLRRS0_001.

First, you have to maintain the coding for your customer exit variable (see **Figure 32** on the next page). I won't go any deeper into the coding, but to finish my example:

- At runtime, the query grabs the specific value from the DSO and runs your report with only correct values.
- Even in debugging, the sy-uname (username) can't be changed. Therefore, authorizations are bullet-proof even in debug mode, and nobody can alter them.

The program reads the data for specific users from the DSO object. It also includes the existing variables for OORGUNIT.

Note!

Make sure you set a breakpoint for variables in the user exit if you want to thoroughly debug DSO access.

Finally, you need to maintain your new analysis authorization so that it uses the variable instead of profiles or hard-coded values. Enter the transaction RSEADMIN and select the Maint. button; this leads you to the next process step (**Figure 3**). Then,

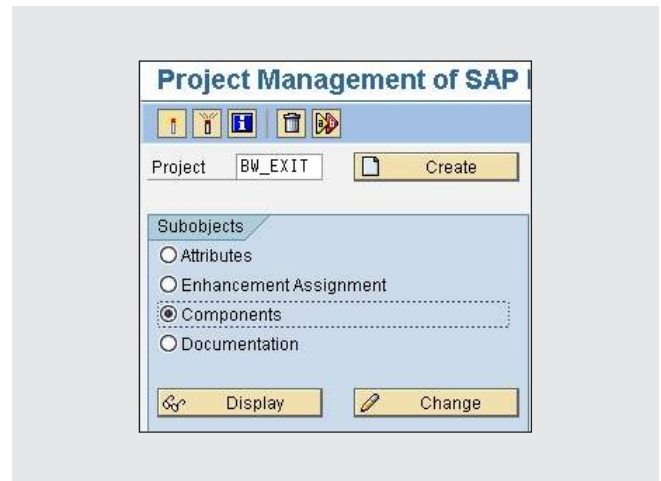


Figure 29 Naming your exit enhancement project

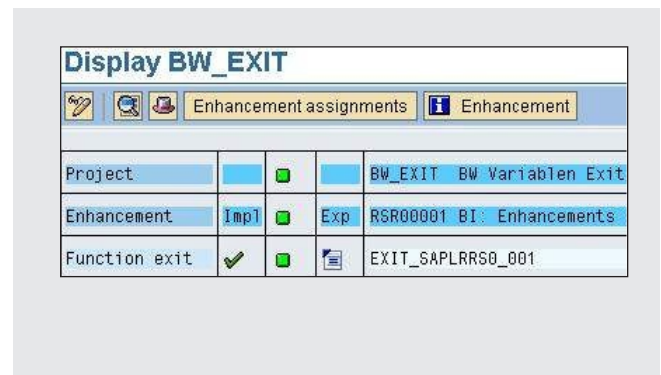


Figure 30 Setting customer function exit for variables

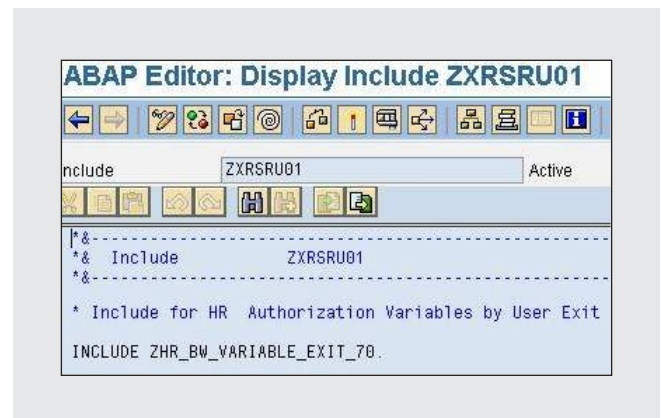


Figure 31 Maintaining the code in ZXRSRU01 for the customer exit


```

*&-----*
*& Include      ZXRSRU01Z
*&-----*

* by Joerg Boeke syskoplan AG
* Data definition for Internal table copies of DSO Authorization objects
TABLES:/BIC/AZAU_BWF00.

*Definition for temp range objects
DATA: l_t_range_t TYPE rsr_s_rangesid ,
      l_t_range like standard table of l_t_range_t ,
      l_s_range like line of l_t_range.

DATA:
* Definition for flat values
      IT_ZAU_BWF_t type /BIC/AZAU_BWF00,
      IT_ZAU_BWF like standard table of IT_ZAU_BWF_t,
      IS_ZAU_BWF like line of IT_ZAU_BWF.

Case i_vnam.
WHEN 'PA_AFORG'.
* Variable for flat Orgunit values
* make sure no false entries in IT
      clear IT_ZAU_BWF .
      clear IS_ZAU_BWF.
      select * from /BIC/AZAU_BWF00 into table IT_ZAU_BWF where
                                TCTUSERNM = sy-uname AND
                                TCTIOBJNM = 'OORGUNIT'.
* Read selected values from internal table
      loop at IT_ZAU_BWF into IS_ZAU_BWF .
        clear l_s_range.
        l_s_range-sign = 'I'.
        l_s_range-opt  = 'EQ'.
*   Get the specific value for Authorization from IT
        l_s_range-low  = IS_ZAU_BWF-TCTLOW.
*       My_s_range-high = IS_ZAU_BWF -TCTHIGH.
        APPEND l_s_range TO e_t_range.
      Endloop.
ENDCASE.

```

Figure 32 Include definition for DSO authorization InfoObjects

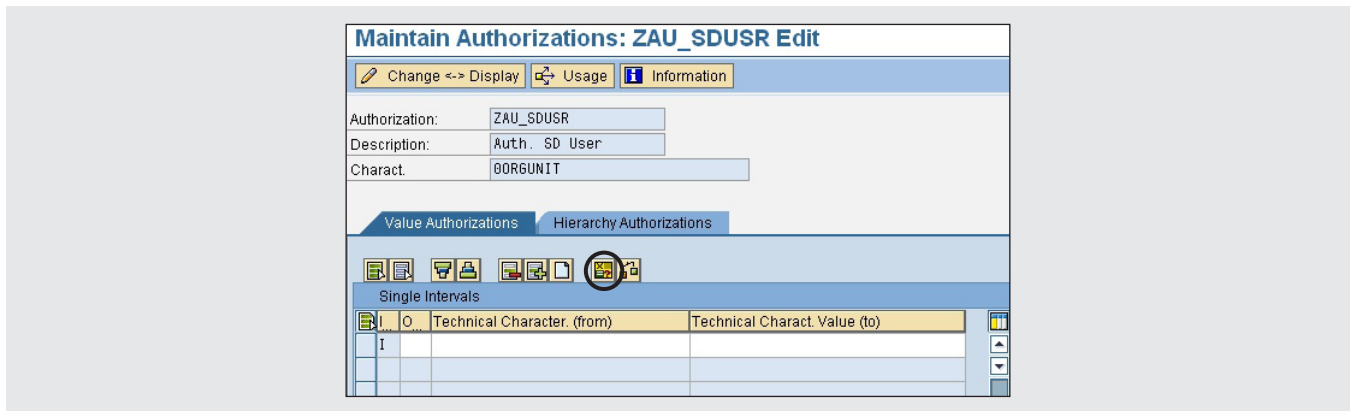


Figure 33 Detail maintenance variable assignment

enter the authorization name (both the technical name, e.g., ZAU_SDUSR, and a description).

Instead of entering manual values, let's assign the variable to the authorization you just created. To assign the variable, select the Insert Exit Variable button in the detail maintenance section of your authorization profile (see **Figure 33**).

All available customer exit variables for each selected InfoObject appear. If only one variable exists, the system will insert it into your authorization automatically. From here on, the system reads the restrictions per user dynamically from the authorization DSO. Whenever changes are loaded into BI, they are active immediately after warehouse management activates new data. By generating profiles in a smooth and flexible way, you eliminate any delay.

After saving the authorization, you can also use it immediately in authorization maintenance and for reporting purposes.

Caution!

Whenever you change settings in these new analysis authorizations, the changes are active immediately; you don't need to reactivate the role.

With some simple steps, you can swap a good many roles and ease the design of authorizations in BI 7.0. Just be sure to double-check your authorizations before you transport them to your productive systems.

Conclusion

Hopefully, you now have a better understanding of the importance of ongoing analysis security. You can use the new analysis authorization and security features in SAP NetWeaver BI 7.0 to ensure and track security across your system landscape. The benefits gained from implementing this new concept include the following:

- Using authorization-relevant attributes instead of creating redundant objects over and over again saves you time and money.
- The new analysis authorizations, in conjunction with authorization extraction, ease the daily business of your administrators by cutting down the amount of time required for manual maintenance when using the authorization extraction and profile generation that SAP provides.
- Using variables instead of generated profiles provides more time for warehouse management because you've eliminated the whole generation process.

- By activating the SAP authorization content object and loading the authorization values into DSOs, you can use SAP BEx reports to keep an overview of the security for each individual in your enterprise (i.e., SAP NetWeaver BI, SAP CRM, or SAP ERP).

If you have specific requests about how to enforce security, while keeping it simple for your enterprise, please don't hesitate to contact me.