Tabellen-Debugging in SAP R/3 – Eine unterschätzte Gefahrenquelle in der Administration

Von Dipl.-Betriebswirt Christoph Wildensee, Hannover [1]

Einführung

Debugging ist ein Begriff aus der Softwareentwicklung und beschreibt ursächlich das Entfernen von Fehlern (Bugs) aus dem Softwarecode. Dabei wird der Programmcode sequenziell abgearbeitet und die Quelle der Fehlererzeugung sukzessive eingegrenzt, bis der Fehler gefunden und korrigiert ist.

Das SAP-System arbeitet tabellenorientiert und weist somit je nach Installation, d.h. Release-Stand 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. Das manuelle Ändern von Tabellen ist vor dem Hintergrund vorhandener Ordnungsmäßigkeitskriterien (GoB, HGB, KonTraG, IDW PS330 etc.) nicht zulässig und somit grundsätzlich zu unterbinden. Manipulationen sowohl in den operativen Geschäftsdaten als auch in der Steuerung des Gesamtsystems können zu Inkonsistenzen und somit zum Verlust der Systemintegrität führen. Vereinzelt kommt es vor, dass ein manueller Eingriff notwendig ist, dieser Schritt darf jedoch nur in Ausnahmefällen und durch fachkundiges Personal in üblicherweise unkritischen, d.h. nicht rechnungslegungsrelevanten, Tabellen durchgeführt werden und muss sachgerecht dokumentiert werden. Somit ergibt sich ein sinnvoller Ansatz, den Zugriff auf Tabellen insbesondere des Moduls FI und der Basis auch für SAP-Administratoren zu unterbinden, sie nur auf die für sie relevanten Tabellen zuzulassen, die kritischen Zugriffe aus der Rolle des Moduladministrators herauszulösen und – wenn möglich – ausschließlich einem Ausnahmeuser (z.B. NOTFALL oder einem spezifischen FI-Modulbetreuer) zu überlassen.

Es ist davon auszugehen, dass das SAP-System vor unberechtigten Zugriffen durch ein entsprechendes Berechtigungskonzept ausreichend geschützt wird und Manipulationen nicht vorkommen können. **Leider muss festgestellt werden, dass der Grad der Manipulierbarkeit sehr hoch ist.**

Der vorliegende Artikel soll zum einen die Möglichkeit des Tabellen-Debuggings im SAP aufzeigen und zum anderen die Hintergründe darlegen, die dazu führen, dass solche Aktionen durchführbar sind.

Voraussetzung

In SAP R/3 sind verschiedene Transaktionen vorhanden, die den Zugriff auf Tabellen ermöglichen. Einige hiervon lassen nur das Anzeigen zu, während andere auch das Ändern und Anlegen erlauben.

Transaktion	Beschreibung
SE11	R/3 Data-Dictionary – Komplettzugriff auf Tabellen, Views, Domänen etc.
SE12	R/3 Data-Dictionary – Anzeigezugriff auf Tabellen, Views, Domänen etc.
SE16*	Data Browser – Anzeigen von Tabellen / Allgemeine Tabellenanzeige
SE17	Allgemeine Tabellenanzeige
SE38	ABAP Editor – Entwicklungsumgebung mit Verzweigung in Objekt-Ansicht
SE80	Objekt-Navigator – Zugriff auf div. Objekte wie Programme, Strukturen etc.
SM30	Tabellen und Sichten – Anzeige, Pflege etc.
SM31	Tabellen und Sichten – Anzeige, Pflege etc.

Tab. 1: Beispiel-Transaktionen zum Tabellenzugriff

Das Anzeigen von Tabellen wird generell als nicht kritisch angesehen. So wird den Administratoren auch meist die Anzeigeberechtigung auf alle Tabellen (oft mit Ausnahme von HR) zugestanden. Zusätzlich erhalten sie Pflegerechte, jedoch nur auf Tabellen der von ihnen betreuten Module. **So soll der Manipulationsgrad nach Relevanz bzw. Zugehörigkeit dezidiert steuerbar sein.**

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

Beispieleingrenzung Classic-System ohne HR: Administrator für alle Module - außer Basis und FI

Anzeigeberechtigung auf alle Tabellen:		
Berechtigungsobjekt	:	S_TABU_DIS
Aktivität	: Anzeigen	03
Berechtigungsgruppe	: alle	*
zusätzlich:		
Änderungsberechtigung auf ausgesuchte Tabellen:		
Berechtigungsobjekt	:	S_TABU_DIS
Aktivität	: alle	*
Berechtigungsgruppe	: alle, außer Basis und FI	A*, C*-E*, G*-R*, T*-Z*
somit	: kein B*, F*, S*	
zusätzlich:		
Änderung mandantenunabhängiger Tabelle:		
Berechtigungsobjekt	:	S_TABU_CLI
mand.unabh. Pflege	: kein Zugriff	‘ ‘
zusätzlich:		
ABAP-Workbench:		
Berechtigungsobjekt	:	S_DEVELOP
Aktivität	: alle (Anz., Änd., Aktiv....)	*
Entwicklungsklasse	: alle	*
Objektname	: alle	*
Objekttyp	: alle, außer Programme	A*-O*, Q*-Z*
Ber.gruppe ABAP-Prog	: alle	*

So ist gewährleistet, dass keine Entwicklung im Produktionssystem erfolgen kann, jedoch alle notwendigen Anpassungen möglich sind, die im operativen Geschäft eines Moduladministrators auftreten. Er hat eine Anzeigeberechtigung auf alle Tabellen, kann jedoch nur die Tabellen direkt ändern, die seinem Aufgabenspektrum entsprechen. Der Ausschluss des Moduls FI und der Basis bei zusätzlichem FI- und Basis-Transaktionsentzug bedeutet eine Erhöhung der Sicherheit.

Es ist jedoch ein Trugschluss anzunehmen, der Administrator darf mit dieser Eingrenzung nicht auf **alle Tabellen** im SAP-System ändernd zugreifen !

Vorgehen

Über den Debug-Modus des Systems ist er berechtigt, Inhalte in allen Tabellen zu ändern, selbst wenn er über die Transaktion SE16 (Tabellen anzeigen) den Zugriff realisiert – auch in Tabellen, die nicht in seiner Berechtigungseingrenzung liegen.
Hierzu muss er z.B. wie folgt vorgehen:

1. Aufruf der Transaktion SE16 – Data Browser - Einstieg
2. Angabe des Tabellennamens, z.B. TJ02 (Systemstatus); diese Tabelle gehört zur Berechtigungsgruppe BS = Steuerung SAP, auf die der Administrator lediglich lesenden Zugriff hat
3. Button Tabelleninhalt (F7) zur Auswahl und Anzeige der Tabelleneinträge
4. Das Ausführen (F8) stellt alle Inhalte in einer Liste dar
5. Auswahl eines Datensatzes zur weiteren Ansicht per Doppelklick
6. Der Datensatz wird angezeigt, keine Änderungsmöglichkeit
7. Im Kommandofeld wird das Debugging eingeschaltet (/h + Return) und danach noch das zusätzliche Betätigen der Return-Taste
8. Quellcode erscheint

9. Bei Feldnamen CODE (Return-Taste) und bei Feldinhalt EDIT (Großschreibung) eintragen; es ist zu beachten, dass die Return Taste nach dem Eingeben von EDIT nicht betätigt werden darf, sondern nur der Stift (2.), dann muss diese Eingabe gespeichert werden über das Save-Symbol (3.)
10. Das Programm muss nun weiter laufen (F8) [=> Replace-Funktion]
11. Nun erscheinen alle Felder mit weißer Hintergrundfarbe – die Felder können geändert werden
12. Nach der Änderung muss diese mit dem Save-Button gespeichert werden, es erfolgt die Meldung ‚Datensatz erfolgreich weggeschrieben'

Abb. 1: Debug-Modus über /h aus SE16 heraus - code-Variable vor Umschalten in EDIT

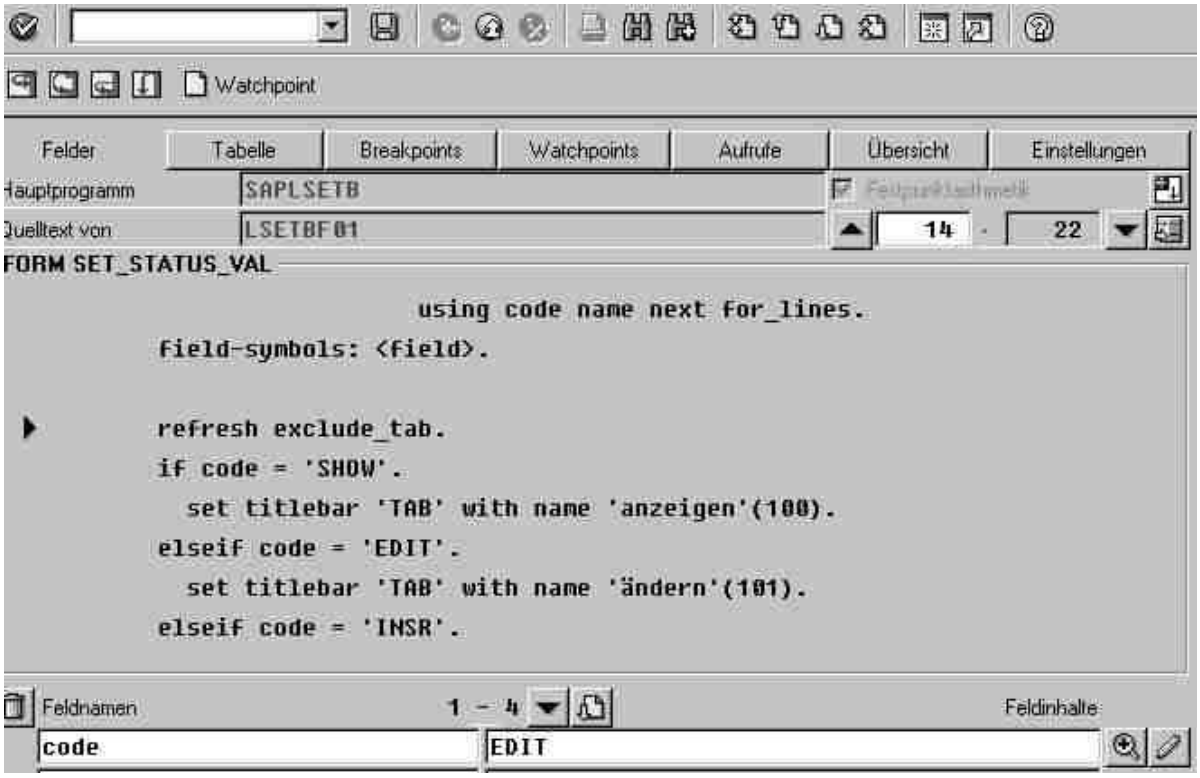

Abb. 2: Debug-Modus über /h aus SE16 heraus – code-Variable geändert und Speichervorgang

Tabelle	Bezeichnung
DD02L	Eigenschaften von Tabellen, Views etc.
DD02T	Texte zu den R/3-Tabellen
TBRG	Tabellen-Berechtigungsgruppen
TDDAT	Zuordnung von Tabellen zu den Berechtigungsgruppen

Tab. 2: Tabellen zum Überprüfen von Tabellen

Berechtigungen

Die meisten der o.g. Transaktionen überprüfen nur die Berechtigungsobjekte S_TABU_DIS und S_TABU_CLI auf Vorhandensein notwendiger Berechtigungen im Berechtigungssatz. So erfolgt beim Aufruf der Transaktion SE16 die Abfrage, ob der Benutzer die Tabelle, die er ausgewählt hat, **einsehen** darf - gesteuert über die Tabellenberechtigungsgruppe (S_TABU_DIS mit Aktivität = 03 und Berechtigungsgruppe = xy [TBRG] und ggf. S_TABU_CLI mit CLIIDMAINT = 'X', wenn Tabelle mandantenunabhängig). Da diese Berechtigung vorhanden ist (03/*), ist hier noch kein Problem zu erkennen.

Der Aufruf des Debuggings (/h) in der Kommandozeile bewirkt, dass der Debugger aktiviert wird, d.h. das Programm hält an und räumt die Möglichkeit ein, jeden Programmschritt einzeln anzustoßen und das Verhalten des Programme z.B. unter Verwendung von Watch-/Breakpoints und anhand der Beobachtung der Programmvariablen zu analysieren. Die Änderungs- / Speicher (Replace-) funktion im Debug-Mode ist an die Änderungsberechtigung (Aktivität 02) des Berechtigungsobjektes S_DEVELOP gekoppelt. Wer hier eine Eingrenzung im Berechtigungssatz vorliegen hat, ist in der Lage, Änderungen zu speichern. Eine Eingrenzung wie oben dargestellt ermöglicht den Aufruf von Tabellen und das Speichern von Inhaltsänderungen.

Da beim Debugging keine erneute Überprüfung des Tabellenzugriffs erfolgt, ob der Nutzer auch hierauf ändern darf, ist auch das Ändern von Tabellen aus nicht autorisierten Berechtigungsgruppen möglich. **Eine Absicherung als kritisch erkannter Tabellen(berechtigungsgruppen) oder sogar des gesamten Moduls FI und der Basis ist somit ausgeschlossen !**

Protokollierung

Aus Sicht der Revision ist die Protokollierung in SAP ein wichtiger Bestandteil, um Hinweise für eine nicht ordnungsgemäße Nutzung der Funktionalität zu erhalten. Das SysLog unter SM19 – Security Audit: AuditProfil verwalten – und SM21 – AuditLog Auswertung – bieten hierfür eine Möglichkeit, sofern die Protokolldaten zeitlich sinnvoll vorgehalten werden. Die Eingrenzung der Datenmenge über den Expertenmodus und die Nutzung der Meldungskennungen bieten eine sinnvolle Selektion.

Relevante Meldungskennungen gem. Tabellen TSL1D/TSL1T:

A1*	Meldungen zur Änderung, Initialisierung/Generierung, Laufzeitfehler, Replace
A14	in Programm ..... Zeile .... Ereignis.....
A19	**Feldinhalt verändert ....**

Über die Transaktion SM21 und der Eingrenzung auf die Meldungskennungen A14 und A19 im Expertenmodus sowie einer sinnvollen Datumseingrenzung – z.B. über ein halbes Jahr – und dem Einlesen aller entfernten SysLogs können alle Änderungen, die über die Debug-Funktionalität (Replace) erfolgt sind, eingesehen werden.

SysLog: Entfernte Auswertung für alle Instanzen 2

Zeit	Instanz	Typ	Nr	Man	Benutzer	Tcod	MNr	Text Datum: 17.02.
07:59:17	[redacted]	DIA	0		[redacted]	SE16	A19	Feldinhalt verändert: code - EDIT
07:59:17		DIA	0			SE16	A14	> in Programm LSETBF01 , Zei 0040, Ereignis SET_STATUS_VA
08:01:26		DIA	0			SE16	A19	Feldinhalt verändert: code - EDIT
08:01:26		DIA	0			SE16	A14	> in Programm LSETBF01 , Zei 0017, Ereignis SET_STATUS_VA
08:36:20		DIA	1			SE16	A19	Feldinhalt verändert: code - EDIT
08:36:20		DIA	1			SE16	A14	> in Programm LSETBF01 , Zei 0017, Ereignis SET_STATUS_VA
08:36:30		DIA	1			SE16	A19	Feldinhalt verändert: code

Abb. 3: SM21 – Protokollierung der EDIT-Aktivität im SysLog

In der Detaillierung ist selten die genaue Feldänderung feststellbar. Der Verweis auf Programme und Programmzeilen ermöglicht zwar die Position der Veränderung, eine Historie dieser wird jedoch nicht mitgeführt. So ist nicht erkennbar, welchen Wert ein Feld vor der Manipulation aufwies.

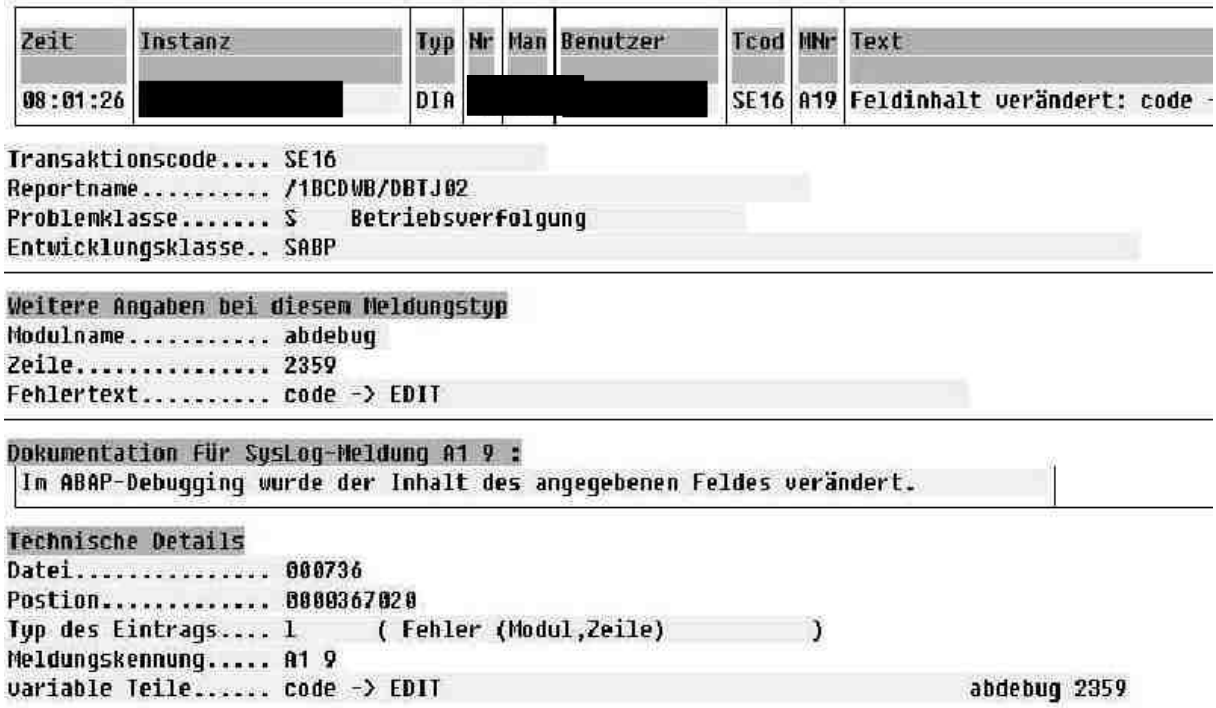

Abb. 4: Detaillierung des SysLog-Eintrags

Die Protokollierung dieser Vorgänge ist zwar grundsätzlich unzureichend, kann jedoch durch die Auswertbarkeit der Ereignisse zumindest als Hinweisgeber dienen. Sofern die Häufigkeit solcher Direktmanipulationen auffällig ist, muss darauf geachtet werden, dass im zuständigen Fachbereich eine entsprechende Dokumentation mit Vorher/Nachher-Hardcopy erstellt wird. Da Datenänderungen in der Regel nicht ohne Anweisung aus einem Fachbereich kommen (z.B. Finanz- und Rechnungswesen), sollte dort die Dokumentation vorgehalten, regelmäßig durch die Revision eingesehen und mit SM21-Auswertungen abgeglichen werden.

Sensibilisierung

< Im ZIR-Artikel wird die oben beschriebene Vorgehensweise der Replace-Funktionsnutzung nicht dargestellt. Sie ist zu umfassend und zu technisch, jedoch m.E. für einen prüfenden Revisor unverzichtbar. >

Entscheidend für mich ist, dass eine Führungskraft – nicht nur der Revisionsleiter, sondern auch z.B. der Leiter des Finanz- und Rechnungswesens als „Hüter“ per Gesetz zu schützender Informationen – sensibilisiert werden muss, dass kein Expertenwissen benötigt wird, um in SAP R/3 manipulierend tätig zu sein. Entsprechend **müssen** die Berechtigungen massiv eingeschränkt werden, da ansonsten trotz definierter Sicherungsmaßnahmen auf Berechtigungsobjektebene in den Modulen keine tatsächliche Sicherheit gegeben ist.

Ausschluss

Das Gefahrenpotential ist also leicht erkennbar. Bei Vorhandensein dieser Berechtigungen kann nicht sichergestellt werden, dass die als kritisch erkannten Tabellen einzelner Module und der Basis abgesichert werden. Selbstverständlich muss gerade den Administratoren ein Mindestmaß an Vertrauen entgegengebracht werden. Jedoch sollte die Zahl derjenigen, die man als vertrauenswürdig einstuft,

nicht unnötig erhöht werden. Aus der Erfahrung heraus lässt sich festhalten, dass es ausreicht, zwei oder drei Administratoren zuzügl. Notfalluser den Komplettzugriff zu ermöglichen, während alle anderen mit Teilfunktionsabdeckung zu definieren sind. Dies ist eine ausreichende Handhabung auch bei sehr großen SAP-Konstrukten. Der Aufbau einer „Zweiklassengesellschaft“ in der Administration ist jedoch eher als kontraproduktiv zu sehen und verstärkt den Eindruck einer Misstrauenskultur.

Ein Ausschluss der Replace-Funktion ist nur möglich, wenn die Änderungsberechtigung im Berechtigungsobjekt S_DEVELOP in der Berechtigungsdefinition der Administratoren entzogen wird. Hierdurch entfällt gänzlich die Änderungsberechtigung – auch auf Tabellen, die geändert werden dürfen. Inwieweit sich eine solche Maßnahme also im operativen Geschäft der Administratoren niederschlägt, hängt von der Definition der Fachbereichs- und Administratorrollen in den Systemen Produktion / Integration / Test-QS, der Abgrenzung der Aufgaben dieser beiden Gruppen und der Kommunikationsfähigkeit der Mitarbeiter untereinander ab.

Fazit

Über den Entzug der Replace-Funktion im Produktionssystem sollte grundsätzlich nachgedacht werden. Das Gefahrenpotential, das mit der Vergabe dieser Berechtigungen verbunden ist, ist zu hoch, um damit nach dem Grundsatz des ‚Laissez-faire‘ umzugehen. Das Erkennen der Gefahrenquelle, das regelmäßige Analysieren durch die Administration selbst und die Revision und das Definieren einer klaren **Richtlinie** für den Fall der Nutzung des direkten Tabellenänderns incl. Verpflichtung der Berechtigten und einer angemessen-zentralen Dokumentation kann in diesem Bereich ein Mindestmaß an Sicherheit erzeugen, reicht jedoch m.E. nicht aus. Der restriktive Einsatz, d.h. die Beschränkung maximal auf zwei Basis-Administratoren und den Notfalluser, ist der einzig gangbare Weg, um adäquate Sicherheit zu erzeugen und auch in rechtlicher Hinsicht für Klarheit zu sorgen.

Literaturbeispiele

Thomas Tiede	Ordnungsmäßigkeit und Prüfung des SAP-Systems, 1. Auflage, Hamburg, 2000
Geesmann/Glauch/Hohnhorst	SAP R/3 Datenschutz und Sicherheitsmanagement, 1. Auflage, Hamburg, 2000
Striebeck/Glauch/Kumpf u.a.	SAP R/3 Sicherheit und Prüfung, 2. Auflage, Hamburg, 2003
Beyer/Fischer/Jäck/Probst u.a.	SAP Berechtigungswesen, 1. Auflage, Bonn, 2003

Internetseiten:

http.//www.sap-press.de

http://www.osv-hamburg.de

http://www.wildensee.de/veroeff.htm

* Möglichkeit des Entzugs: Wenn eine Eingrenzung von S_DEVELOP erfolgen soll, die weiterhin die Aktivität 01 oder 02 zulassen, ein Replace aber verhindern soll, so muss S_DEVELOP mit Aktivität 01 oder 02 eine Eingrenzung bei Objekttyp beinhalten, die nicht nur PROG ausschließt, sondern auch weitere untergeordnete Objekttypen wie TABL u.ä., da bei ausschließlichem Fehlen von PROG ein Replace trotzdem noch möglich ist. Ein Beschränken auf SYST (Laufzeitanalyse...) ist z.B. möglich. *