

# Datenbank-Auditing

AID 072 DE



© 2017 ADITO Software GmbH

Diese Unterlagen wurden mit größtmöglicher Sorgfalt hergestellt. Dennoch kann für Fehler in den Beschreibungen und Erklärungen keine Haftung übernommen werden. Wir sind für Feedback zu den Themen, Inhalten, aber auch noch vorhandenen Fehlern dankbar und würden uns freuen, Ihre Meinung zu hören. Die in diesen Unterlagen enthaltenen Daten und Angaben, einschließlich URLs und anderer Verweise können ohne vorherige Ankündigung geändert werden. Alle in diesen Unterlagen aufgeführten Produkt- und Firmennamen sind unter Umständen Marken oder geschützte Zeichen der einzelnen Firmen. Ohne ausdrückliche schriftliche Einverständniserklärung der ADITO Software GmbH darf kein Teil dieses Dokumentes vervielfältigt oder in einer Datenverarbeitungsanlage gespeichert oder in diese eingelesen werden. Diese Einschränkung gilt unabhängig von Art und Weise der Datenerfassung.

Autor: MW, KN. Version 10.3. Zuletzt geändert 25.10.2017

Version	Änderungen
<b>10.3</b>	kleinere Änderungen in den Beschreibungen
<b>10.2</b>	Anpassung der Formatierungen
<b>10.1</b>	Ergänzen der Einstellungen für ADITO4.6
<b>10.0</b>	Anpassung an ADITO4.6
<b>2.0</b>	Letzter Stand vor Übernahme in Versionierung

# Inhaltsverzeichnis

<b>1.</b>	<b>Datenbankprotokollierung in ADITO .....</b>	<b>4</b>
1.1.	Aktivierung der AUDIT-Funktion in ADITO online 3.1 .....	4
1.2.	Aktivierung der AUDIT-Funktion in ADITO4 .....	4
<b>2.</b>	<b>Auditing in ADITO aktivieren .....</b>	<b>5</b>
<b>3.</b>	<b>Daten protokollieren .....</b>	<b>8</b>
3.1.	Einstellung der Protokollierung in ADITO online 3.1 .....	8
3.2.	Einstellung der Protokollierung in ADITO4 online .....	8
3.3.	Einstellungsmöglichkeiten der Protokollierung .....	9
<b>4.</b>	<b>Die Systemtabelle ASYS_AUDIT .....</b>	<b>10</b>
4.1.	SQLAction .....	10
4.1.1.	I (Insert) / System .....	11
4.1.2.	U (Update) / System .....	11
4.1.3.	D (Delete) / System .....	11
4.1.4.	X (Unbekanntes Statement) / System .....	11
4.1.5.	P (Process) / User .....	11
4.1.6.	C (Change) / User .....	11
4.1.7.	T (Transaction) / User .....	11
<b>5.</b>	<b>Protokollieren ohne Datenbank .....</b>	<b>12</b>
5.1.	Beispiel .....	12
<b>6.</b>	<b>Datenfilterprotokoll .....</b>	<b>15</b>

## 1. Datenbankprotokollierung in ADITO

Zur Nachverfolgung von Änderungen an Datensätzen kann über den ADITO Designer für jede Datentabelle gesondert ein Auditing (getrennt nach Datentypen ohne LOBs, Daten inkl. CLOBs bzw. Daten inkl. CLOBs und BLOBs) aktiviert werden. Die Aktivierung bzw. Deaktivierung des Auditings ist den ADITO Administratoren vorbehalten. Damit werden alle Datenfeldänderungen von Benutzern in einer zentralen Tabelle protokolliert.

Diese umfasst den Benutzerlogin, den Zeitstempel der Änderung, die ausgeführte Art der Änderung, den Datenbankalias, den Namen der Tabelle und der Spalte, den neuen Datenwert, eine eindeutige Protokollkennung und den Zeitstempel der Protokollierung.

Sollen die initialen Datenwerte erfasst werden, muss das Auditing vor dem Import der Daten aktiviert worden sein. Über dieses Protokoll kann für Revisionszwecke eine Liste aller Datenänderungen zu einem Datensatz erfolgen. Da dies keine Endbenutzerfunktionalität darstellt, existiert hierfür keine Benutzeroberfläche. Diese Auswertungen werden üblicherweise durch den Datenbankadministrator erzeugt. Eine Bereinigung bzw. ein Abschneiden oder Löschen von Daten in dieser Protokolltabelle durch ADITO erfolgt nicht, sondern die Tabelle muss durch den Administrator gekürzt werden.

### 1.1. Aktivierung der AUDIT-Funktion in ADITO online 3.1

Die Audit-Funktion wird im Designer von ADITO online im Preferences-Editor, Reiter *Database* aktiviert. Damit sie genutzt werden kann, muss für jede Tabelle, bei der Auditing gewünscht ist, einzeln diese Funktion aktiviert werden. Das geschieht im ALIAS-Editor, im Reiter *Sequences*. Um Auditing zu aktivieren, muss bei `ID_AUDIT` die Art des Auditing gewählt werden. Standardmäßig ist die Funktion deaktiviert - dies wurde so gewählt, um unnötige Zugriffe auf die Datenbank zu verhindern. Der Einzelfall soll entscheiden, wann eine Tabelle mit protokolliert werden soll, und wann nicht. *Standard* protokolliert dabei alle Änderungen an Standard-Datentypen, *CLOB* zusätzlich alle Änderungen in *CLOB*-Feldern, *BLOB* analog zu *CLOB*. Standardmäßig werden alle Informationen in der Tabelle `AOSYS_AUDIT / ASYS_AUDIT` gespeichert. Diese kann dann per SQL oder in einem Frame abgefragt werden, um die protokollierten Änderungen auszuwerten.

### 1.2. Aktivierung der AUDIT-Funktion in ADITO4

Die Audit-Funktion wird im Designer von ADITO4 im Alias-Editor eingerichtet in dem Reiter, der so benannt ist wie der Alias selbst. Damit sie genutzt werden kann, muss für jede Tabelle, bei der Auditing gewünscht ist, einzeln diese Funktion aktiviert werden. Das geschieht im ALIAS-Editor, im selben Reiter. Um Auditing zu aktivieren, muss bei `ID_AUDIT` die Art des Auditing gewählt werden. Standardmäßig ist die Funktion deaktiviert - dies wurde so gewählt, um unnötige Zugriffe auf die Datenbank zu verhindern. Der Einzelfall soll entscheiden, wann eine Tabelle mit protokolliert werden soll, und wann nicht. *Standard* protokolliert dabei alle Änderungen an Standard-Datentypen, *CLOB* zusätzlich alle Änderungen in *CLOB*-Feldern, *BLOB* analog zu *CLOB*. Standardmäßig werden alle Informationen in der Tabelle `AOSYS_AUDIT / ASYS_AUDIT` gespeichert. Diese kann dann per SQL oder in einem Frame abgefragt werden, um die protokollierten Änderungen auszuwerten

## 2. Auditing in ADITO aktivieren

Auditing wird im Preferences-Editor des ADITO Designers aktiviert.

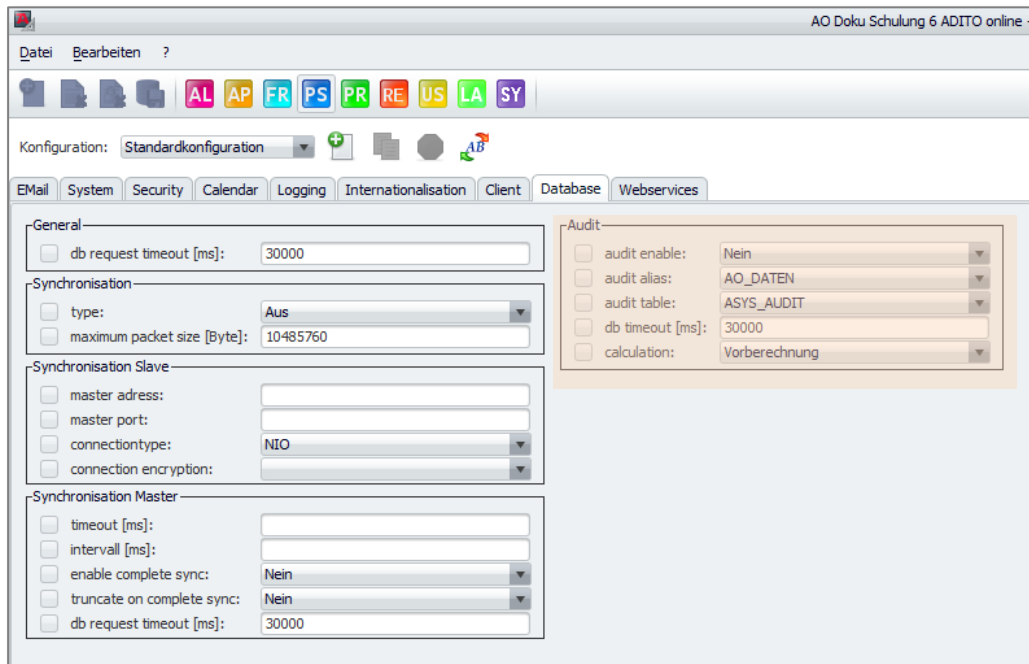


Abbildung 1: ADITO online 3.1

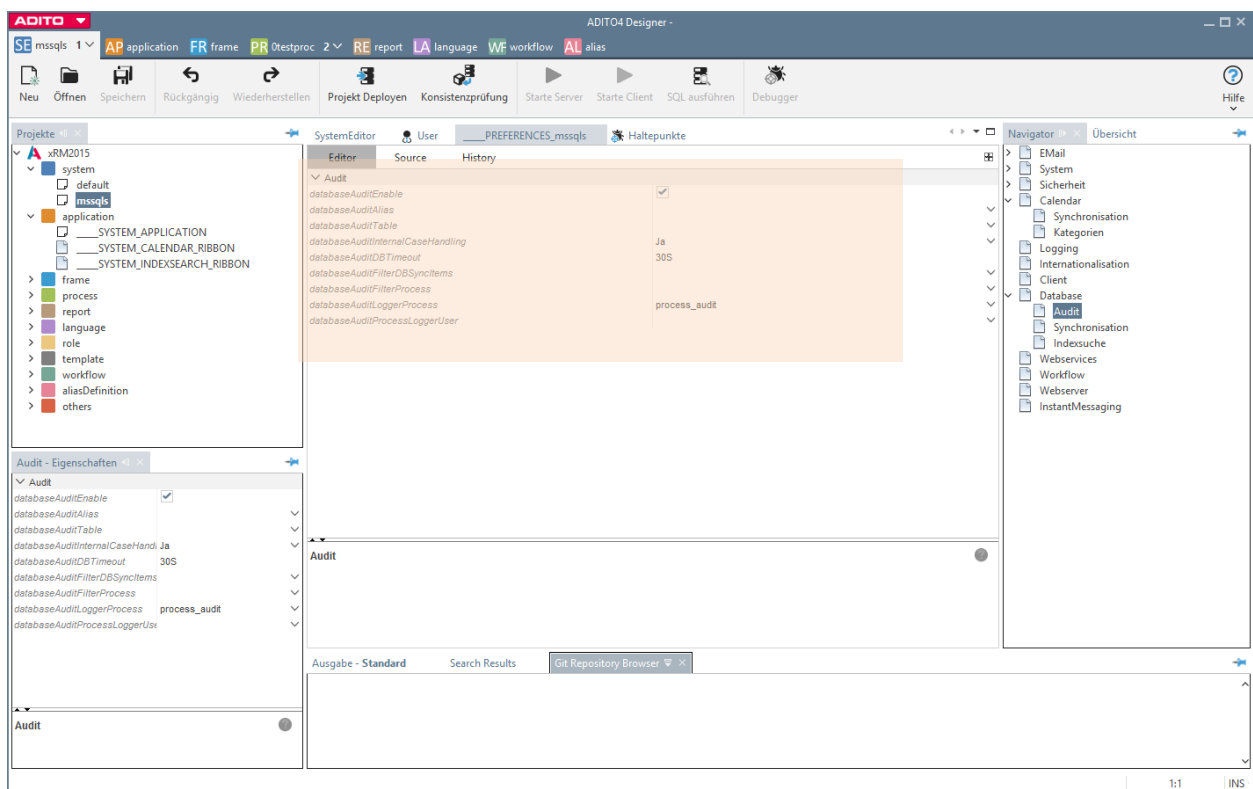


Abbildung 2: ADITO4 online

Eigenschaft	Beschreibung
<b>databaseAuditEnable</b>	Aktiviert die Auditing Funktion.
<b>databaseAuditAlias</b>	Der Datenbankalias, in welchem die Veränderungen im Datenbestand protokolliert werden sollen.
<b>databaseAuditTable</b>	Eine Tabelle <code>ASYS_AUDIT</code> , in welcher die Änderungen protokolliert werden. Kann im Sytemeditor-Editor nach dem Doppelklicken der gewünschten Datenbank und dem Öffnen des Baumes bis Tabellen über den Kontextmenüeintrag Systemtabellen anlegen angelegt werden.
<b>databaseAuditInternalCaseHandling</b>	Aktiviert das interne Case-Handling, welches dazu führt, dass in der Systemdatenbank auch Indizes verwendet werden können, was die Performance steigert. Steht dieser Wert auf „Nein“, wird die Abfrage auf die Systemdatenbank mit einem <code>UPPER()</code> -Befehl versehen.
<b>databaseAuditDBTimeout</b>	Der timeout, bei dem der ADITO-Server nicht mehr versucht, die Änderungen zu protokollieren.
<b>databaseAuditFilterDBSyncItems</b>	Aktiviert den Datenbankfilter für Synchronisations-Items. Das unterdrückt bei der Synchronisation das Schreiben der Datensätze, die aus dem jeweils anderen System kommen. Diese Einstellung ist am Slave sinnvoll, damit Änderungen, die vom Master kommen, nicht auch in die Audit-Schicht eingetragen werden.
<b>databaseAuditFilterProcess</b>	Gibt den Prozess an, der zur Filterung des Audit-Logging verwendet werden soll.
<b>databaseAuditLoggerProcess</b>	Gibt den Prozess an, der für das Audit-Logging verwendet werden soll.
<b>databaseAuditProcessLoggerUser</b>	Der Benutzer, der für den Audit-Logging-Process verwendet wird.
<b>databaseAuditGetOldValueLobs</b>	Wenn „Ja“, werden auch die alten Werte mitgegeben. Als Auswirkung

wird das Audit langsamer und  
verbraucht mehr Speicher.

## 3. Daten protokollieren

### 3.1. Einstellung der Protokollierung in ADITO online 3.1

Wurde nun Auditing aktiviert, so kann dieses für einzelne Tabellen verwendet werden. Welche Werte in welchen Tabellen protokolliert werden, stellen Sie im Alias-Editor, Reiter Sequences ein:

Alias: AO_DATEN		ID_AUDIT	
ID_TABLE	ID_COLUMN	ID_AUDIT	
ASYS_SYSTEM	ID	Aus	
ATTR	ATTRID	Standard	
ATTRLINK	ATTRLINKID	Standard	
ATTROBJECT	ATTROBJEC	Standard	
BANKACCOUNT	BANKACCOI	Aus	

Der Sequences-Reiter des Alias-Editors dient eigentlich dazu, die von ADITO online angestoßene ID-Generierung zu verwalten. Allerdings kann hier für jede Tabelle auch eingestellt werden, welche Spalten protokolliert werden sollen. Diese Einstellung wählen Sie im Reiter ID\_AUDIT. Die Möglichkeiten sind in Kapitel 3.3 beschrieben.

### 3.2. Einstellung der Protokollierung in ADITO4 online

Ab ADITO der Version 4.6 werden die Einstellungen zum Audit im aliasDefinition-Editor gepflegt. Markieren Sie dafür die Tabelle, für die die Protokolllierung eingeschaltet werden soll und stellen Sie in den Properties den `auditMode` ein. Die Möglichkeiten hierfür finden Sie im nächsten Kapitel.

Navigator		ORG - Properties	
>	NODUPLICATE	name	ORG
>	OBJECTRELATION	title	Organisationen
>	OFFER	description	Die Tabelle ORG enth... ..
>	OFFERITEM	dbName	
>	ORDERITEM	idColumn	ORGID
>	<b>ORG</b>	idGeneratorType	UUID
>	PERS	idGeneratorInterval	1
		auditMode	BLOB

Eine Übersicht, für welche Tabellen das Audit aktiviert wurde, erhalten Sie im Reiter Übersicht der Alias-Definition.



AO_DATEN ×			
Editor	Index	Übersicht	Source
name	idColumn	auditMode	
ABSENCE	ABSENCEID	Aus	▼
ACTIVITY	ACTIVITYID	Standard	▼
ACTIVITY_ACTION	ACTIVITY_ACTIONID	Standard	▼
ACTIVITY_ACTION_IN...	ACTIVITY_ACTION_IN...	Aus	▼
ACTIVITY_CONDITION	ACTIVITY_CONDITION...	Aus	▼
ACTIVITY_INSTANCE	ACTIVITY_INSTANCEID	Aus	▼
ACTIVITY_STEP	ACTIVITY_STEPID	Standard	▼
ADDRESS	ADDRESSID	BLOB	▼

### 3.3. Einstellungsmöglichkeiten der Protokollierung

Folgende Möglichkeiten stehen Ihnen bei der Protokollierung zur Verfügung:

Eigenschaft	Beschreibung
<b>Aus</b>	Keine Protokollierung in dieser Datenbank-Tabelle.
<b>Standard</b>	Veränderungen in allen Datentypen außer CLOB und BLOB werden protokolliert.
<b>CLOB</b>	Änderungen in allen Datentypen außer BLOB werden protokolliert.
<b>BLOB</b>	Änderungen in allen Datentypen dieser Tabelle werden protokolliert.

Wurde beispielsweise der Wert „Standard“ in der Tabelle `ORG` ausgewählt, so wird bei einer Neuanlage einer Firma in die Tabelle `ASYS_AUDIT` protokolliert. Die Protokollierung, welche z.B. für die Datensynchronisation `DBSync` verwendet wird, kann über folgenden SQL geprüft werden.



Details zur Datensynchronisation `DBSync` entnehmen Sie bitte dem ADITO-Informationsdokument `AID056-DE - Datensynchronisation DBSync`.

```
select * from asys_audit where atablename = 'ORG' order by logtime desc
```

## 4. Die Systemtabelle ASYS\_AUDIT

Die Systemtabelle ASYS\_AUDIT enthält folgende Spalten:

Spalte	Beschreibung
<b>ACOLUMNNAME</b>	Name der protokollierten Spalte.
<b>ALIAS</b>	Name des Alias, innerhalb dessen eine Änderung stattgefunden hat.
<b>ATABLENAME</b>	Name der Tabelle, innerhalb deren eine Änderung stattgefunden hat.
<b>ATIMESTAMP</b>	Datum der Protokollierung, wann die Transaktion durchgeführt wurde.
<b>AUSER</b>	Hier wird der Benutzer eingetragen, der die Aktion, die zum Audit geführt hat, angestoßen hat.
<b>DATASETID</b>	ID des Datensatzes, bei dem sich einzelne Spalten geändert haben.
<b>DATASETNEWVALUE_BLOB</b>	Protokollierter neuer Wert einer BLOB-Spalte.
<b>DATASETNEWVALUE_CLOB</b>	Protokollierter neuer Wert einer CLOB-Spalte.
<b>DATASETNEWVALUE_STRING</b>	Protokollierter neuer Wert einer beliebigen anderen Spalte außer CLOB und BLOB.
<b>LOGTIME</b>	Datum der Protokollierung, wann das Element in die AUDIT-Tabelle geschrieben wurde.
<b>LOGTYPE</b>	Logtyp, referenziert sich darauf was geloggt wird (NEWVALUE_STRING, _CLOB, _BLOB).
<b>LOGWRITERUID</b>	Die UID des Logwriters.
<b>SQLACTION</b>	Protokollierte Aktion.

### 4.1. SQLAction

In der Spalte SQLACTION wird angegeben, welche Aktion protokolliert wurde. Diese wird entweder vom System vorgegeben, oder kann auch per Hand in die Systemtabelle eingetragen werden, beispielsweise um bestimmte Aktionen für die Datensynchronisation durchzuführen.

#### 4.1.1.I (Insert) / System

Hier werden alle Daten der Spalten protokolliert, die durch einen Insert verursacht wurden.

#### 4.1.2.U (Update) / System

Hier werden alle Daten protokolliert, die bei einem Update verändert wurden.

#### 4.1.3.D (Delete) / System

Hier wird in der Spalte `DATASETID` die ID des Datensatzes protokolliert, welcher gelöscht wurde.

#### 4.1.4.X (Unbekanntes Statement) / System

Wird ein Statement ohne Verwendung der ADITO eigenen Methoden `db.inserts`, `db.insertData`, `db.updates`, `db.updateData`, `db.deletes` oder `db.deleteData` in die Datenbank geschrieben (z.B. über

```
db.cell("update org set orgname = 'irgendwas'");
```

oder ähnliche Befehle) eingetragen, so wird das Statement protokolliert und als `SQLACTION X` eingetragen.

#### 4.1.5.P (Process) / User

Diese `SQLAction` wird von Hand in die Tabelle `ASYS_AUDIT` eingetragen. Hiermit kann ein JDito-Prozess angegeben werden, der über die Einschränkungen der `ASYS_AUDIT` Protokollierung bestimmte Prozesse abgleicht.

#### 4.1.6.C (Change) / User

Auch diese `SQLAction` wird von Hand in die Tabelle eingetragen. Über `Change` können alle Datensätze synchronisiert werden, die eine bestimmte Bedingung erfüllen, unabhängig davon, ob bei diesen Datensätzen eine `SQLACTION I, U` oder `D` vorliegt. Über die Spalte `LOGTYPE` kann durch eintragen der Werte, wie sie aus `util.AUDIT_CONDITION_JDITO` und `util.AUDIT_CONDITION_PLAIN` zurückgegeben werden.

#### 4.1.7.T (Transaction) / User

Ähnlich wie `C`, nur dass hier keine Daten gelöscht bzw. synchronisiert werden können, sondern lediglich transferiert.



Detaillierte Informationen zu den `SQLActions P, C` und `T` können Sie im ADITO-Informationsdokument `AID056-DE - Datensynchronisation DBSync` nachlesen.

## 5. Protokollieren ohne Datenbank

Sie können Auditing auch verwenden, ohne die Systemdatenbank `ASYS_AUDIT` und das daraus resultierende hohe Datenaufkommen zu aktivieren. Dazu muss zwar Auditing auch für die einzelnen Tabellen aktiviert sein (somit werden auch Daten in die Datenbank geschrieben), aber die Audit-Tabelle kann auf Wunsch auch bereinigt werden.

Dazu benötigen Sie einen Prozess `process_audit`, dieser muss in den Prozess-Editor des ADITO Designers eingebunden werden. Hier stehen einige lokale Variablen zur Verfügung.

Variable	Beschreibung
<code>\$local.action</code>	I / U / D für die jeweilige SQLAction.
<code>\$local.user</code>	Benutzer, welcher die Änderung ausführt.
<code>\$local.timestamp</code>	Zeitpunkt der Änderung.
<code>\$local.alias</code>	Alias der Änderung.
<code>\$local.table</code>	Tabelle des betroffenen Datensatzes.
<code>\$local.idvalue</code>	ID-Wert des betreffenden Datensatzes.
<code>\$local.columns</code>	Betroffene Spalten (Array).
<code>\$local.values</code>	Neue Werte (Array).
<code>\$local.oldvalues</code>	Alte Werte (Array).



Process und Change sollten hier nicht verwendet werden.

### 5.1. Beispiel

Folgender Code könnte als `process_audit` zur Verfügung stehen:

```
import ("system.logging");
import ("system.vars");

logging.log("Audit");
logging.log(" " + vars.getString("$local.action"));
logging.log(" " + vars.getString("$local.user"));
logging.log(" " + vars.getString("$local.timestamp"));
logging.log(" " + vars.getString("$local.alias"));
logging.log(" " + vars.getString("$local.table"));
logging.log(" " + vars.getString("$local.idvalue"));
```

```
var xx = vars.get("$local.columns");
try
{
if (xx != null && xx != undefined)
    xx = xx.join("-");
}
catch(ex)
{
}
logging.log(" " + xx);

var xx = vars.get("$local.values");
try
{
if (xx != null && xx != undefined)
    xx = xx.join("-");
}
catch(ex)
{
}
logging.log(" " + xx);

var xx = vars.get("$local.oldvalues");
try
{
if (xx != null && xx != undefined)
    xx = xx.join("-");
}
catch(ex)
{
}
logging.log(" " + xx);
```

Bei einer Veränderung wird dann folgende Ausgabe in die Output-Konsole (z.B. bei Start des Servers über `server.bat`) geschrieben:

```
Audit
```

U

Admin

1329474637822

AO\_DATEN

ORG

ebece025-cc5c-4169-a4bc-b8c130fc1fb1

ORGNAME-DATE\_EDIT

M&M Haider und Wagnermaier GmbH-1329474633000

M&M Haider und Wagnermaier AG-1329470942000

## 6. Datenfilterprotokoll

Sollen nicht alle Daten, sondern nur bestimmte Daten durch den Audit-Filter gejagt werden, kann ein Prozess mit dem Namen `process_auditfilter` angelegt werden. In ihm stehen dieselben Variablen wie in `process_audit` zur Verfügung. Dieser Prozess muss in den Preferences in den Eigenschaften `databaseAuditFilterProcess` eingestellt werden, der Name des Prozesses kann grundsätzlich frei gewählt werden, der vorgegebene Name wird aber empfohlen.

Variable	Beschreibung
<code>\$local.action</code>	I / U / D für die jeweilige <code>SQLAction</code> .
<code>\$local.user</code>	Benutzer, welcher die Änderung ausführt.
<code>\$local.timestamp</code>	Zeitpunkt der Änderung.
<code>\$local.alias</code>	Alias der Änderung.
<code>\$local.table</code>	Tabelle des betroffenen Datensatzes.
<code>\$local.idvalue</code>	ID-Wert des betreffenden Datensatzes.
<code>\$local.columns</code>	Betroffene Spalten (Array).
<code>\$local.values</code>	Neue Werte (Array).
<code>\$local.oldvalues</code>	Alte Werte (Array).



Process und Change sollten hier nicht verwendet werden.

Wird in diesem Prozess nichts oder null zurückgegeben, werden keine Daten gefiltert.

Der Rückgabewert dieses Prozesses ist eine Map in der Form

```
map[spaltenname] = Wert
```

angegeben, z.B.

```
var map = [];
mal[columnname] = value;

result.object(map);
```