

Invoice Update

Dieses Kapitel beschreibt das Vorgehen bei einem Invoice-Update. Es empfiehlt sich generell, dass das Update zunächst auf dem Testsystem des Kunden durchgeführt wird und der Kunde das System nach dem Update ausführlich testet. Erst nach Freigabe durch den Kunden sollte das Produktiv-System aktualisiert werden.

- [Test- und Produktivsystem](#)
- [Sicherung erstellen / Anpassungen separat sichern](#)
- [Documents-Update](#)
- [Documents Update-Ordner](#)
- [Documents XML Importe](#)
- [Update Datenbank-Tabellen](#)
- [Parameter importieren](#)
- [Documents Abgleich Mappentypen](#)
- [Individuelle Versions-Anpassungen / Wichtige Hinweise](#)

Test- und Produktivsystem

Generell wird beim Einsatz einer Invoice-Lösung dringend ein Testsystem empfohlen! Ein Update der Invoice-Version wird häufig mit einem Update der Documents-Version kombiniert und häufig sollen zusätzlich noch projektspezifische Kundenwünsche umgesetzt werden. Die Kombination dieser Updates und neuer Anforderungen kann schnell zeitaufwändig und komplex werden. Gerade bei größeren Versionssprüngen gebündelt mit Anpassungswünschen bietet es sich an, alle Anpassungen zunächst streßfrei auf einer Testumgebung zu testen.

Vor dem Update der Testumgebung sollte das Testsystem zum Produktivsystem abgeglichen werden. Im Optimalen Fall wird ein kompletter JEX-Export übertragen und auch die Documents-Ordner und die DEX-Datenbanken sollten übertragen werden. Im optimalen Fall werden alle Anpassungen immer zuerst auf dem Testsystem vorgenommen und getestet und werden dann erst auf das Prod übertragen. In dem Fall kann man sich den abgleich sparen. Wenn das Testsystem auf demselben Stand wie das Produktivsystem ist, können die Konfigurationen aus dem Testsystem später 1:1 wieder zurück auf das Prod gespielt werden.

Für das Invoice-Update wird in den aktuellen Versionen eine separate Update-Version bereitgestellt.

Es wird dringend empfohlen, dass das Update zunächst auf einem Testsystem durchzuführen!

Das Testsystem sollte vor dem Update zum produktiven System abgeglichen werden!

Die Anpassungen sollten im Anschluss vom Kunden getestet und freigegeben werden!

Alle Besonderheiten und Fallstricke sollten beim Update gut dokumentiert werden. Wenn projektspezifische Anpassungen an Dateien zurückgespielt werden, können für diese angepassten Dateien später für das update des produktiven Systems verwendet werden, damit die Arbeit nicht doppelt durchgeführt werden muss. Erst jetzt sollte das Update auf dem Produktiv-System durchgeführt werden.

Durch ein solches Vorgehen werden Risiken minimiert. Durch die Erfahrungen, die man im Testsystem sammelt und dadurch, dass man auf bereits fertig angepasste User-Exit-Dateien zurückgreifen kann, sollte sich die Downtime bei einer Umstellung des produktiven Systems deutlich verringern.

Sicherung erstellen / Anpassungen separat sichern

Vor dem Update sollte sicher gestellt werden, dass alle Benutzer vom System abgemeldet sind. Die einfachste Variante ist den Tomcat und den TableService zu stoppen.

Der aktuelle Stand sollte in jedem Fall gesichert werden. Das Invoice-Template enthält verschlüsselte Standard-Skripte und Standard-Konfigurationen, die nicht angepasst werden sollten. So lange man sich daran hält, sollte ein Update kein Problem darstellen. Trotzdem sollte möglichst viel manuell gesichert werden.

Ein komplettes Documents-Backup ist als Sicherung gut. Aber ein solches Backup hilft nichts, wenn die projektspezifischen Anpassungen vorab nicht dokumentiert oder separat gesichert wurden! Um kleinere Update-Fehler schnell beheben zu können, sollten viele kleinere Sicherungen zum Beispiel als XML-Export hilfreicher als ein Gesamt-Backup.

Alles was projektspezifisch hinzugefügt oder angepasst wurde, sollte wenn möglich einzeln als XML-Export gesichert werden. Das betrifft zum Beispiel die Mappentypen und Ordner. Bei den Anpassungen in UserExit-Skripten empfiehlt sich generell Visual Studio Code mit der Erweiterung "JavaScript Remote Debugger for JANUS Apps". Hierdurch sind automatisch alle Skript-Anpassungen gesichert.

Bei einem geplanten Invoice- plus Documents-Update bietet es sich an, einmal den kompletten Documents-Ordner zu sichern. Es ist sinnvoll bereits im Projekt alle Anpassungen direkt zu sichern und/oder zu notieren. Dann muss man die Stellen bei einem Update nicht mühselig identifizieren.

Zusammenfassung:

- Empfohlen: Gesamt-Sicherung (Documents-Backup und/oder Snapshot bei virtuellen Maschinen)
- Sicherung Mappentypen "Invoice" / "otrAccessProfile" / "otrUser"
- Angepasste Ordner / Outbars sichern und ggf. mit Screenshots dokumentieren
- Angepasste UserExit-Skripte sichern
- Sicherung Documents-Installations-Ordner

Documents-Update

Im Zuge eines Invoice-Updates sollte auch Documents auf eine stabile aktuelle Version gehoben werden. In der "**DEXPRO_CHANGELOG.md**" und der "**INVOICE_CHANGELOG.md**" wird zu jeder Version eine Empfehlung für die minimal zu verwendende Documents-Version gegeben. Diese Version wurde bei der Entwicklung verwendet und mit dieser Version wurde entsprechend getestet. Einige Neuerungen und Anpassungen könnten Funktionen verwenden, welche erst ab der empfohlenen Version zur Verfügung stehen. Die Changelogs enthalten zudem wichtige Informationen zu manuellen Änderungen.

Wenn ein Invoice-Update ohne ein Update auf die empfohlene Documents-Version durchgeführt wird, sollte dies zunächst auf einem Testsystem durchgeführt und im Anschluss ausreichend getestet werden!

Falls bereits aktuellere Documents-Updates existieren, sollten ggf. Erfahrungsberichte bei den Kollegen eingeholt werden. Es gibt vereinzelt immer wieder Documents-Versionen mit Fehlern! Die Verwendung aktuellerer Documents-Versionen ist in der Regel aber kein Problem.

Documents Update-Ordner

Zu den aktuellen Invoice-Versionen werden passende Update-Pakete bereitgestellt. Diese erleichtern das vorgehen bei Updates. Das Update-Paket enthält einen separaten Ordner "**Documents6**". Dieser Ordner enthält nur die für ein Update relevanten Ordner und Dateien.

Vor dem Update sollten diese Ordner gesichert werden um projektspezifische Anpassungen zurückspielen zu können! Im Folgenden werden einige Informationen zu den Ordnern gegeben.

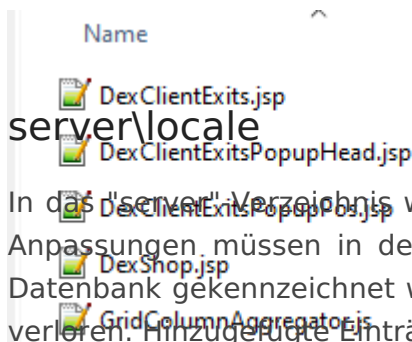
DEXPRO

Der Ordner enthält alle Unterordner und Dateien, die im Laufe der Entwicklung hinzugekommen sind bzw. Dateien, die im Laufe der Zeit angepasst wurden. Der Ordner kann über die bestehende Installation kopiert werden. Die Update-Variante enthält keine Konfigurations-Dateien. Trotzdem sollte der gesamte Documents-Ordner vorab gesichert werden.

Über den Ordner werden auch TableService-Konfigurationen geändert oder hinzugefügt. Der TableService Dienst muss im Laufe des Updates neu gestartet werden.
Es kann sein, dass einige WEB-Konfigurationen erst wieder korrekt angezeigt werden, wenn auch die Datenbank-Tabellen aktualisiert wurden

DEXPRO_ClientExits_

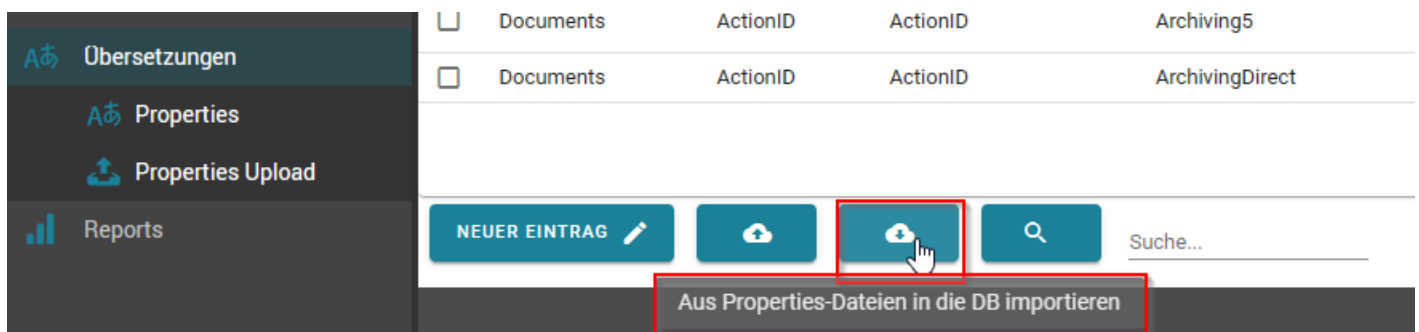
Der Unterstrich hinter dem Ordernamen soll verhindern, dass die Dateien einfach kopiert und ersetzt werden. Die hier befindlichen Dateien dürfen explizit projektspezifisch angepasst und erweitert werden und diese projektspezifischen Anpassungen sollen nicht wieder durch die Standard-Funktionen ersetzt werden. Bei Bedarf können die Dateien manuell abgeglichen werden - das ist aber nicht zwingend notwendig. Dabei sollten lediglich nicht vorhandene Funktionen hinzugefügt werden. Sollte das Update zusätzliche Dateien in dem Ordner enthalten, müssen diese Dateien kopiert werden!




In das "server"-Verzeichnis werden die aktuellen "**properties**"-Dateien kopiert. Projektspezifische Anpassungen müssen in der WEB-Konfiguration als "**projektspezifischen Anpassung**" in der Datenbank gekennzeichnet worden sein - andernfalls gehen diese Anpassungen bei einem Update verloren. Hinzugefügte Einträge bleiben in jedem Fall erhalten!

Nach dem Kopieren aller properties müssen die Daten einmal über die WEB-Konfiguration importiert und wieder exportiert werden!

Beim Hochladen der properties in die Datenbank werden die als projektspezifische Anpassung markierten Einträge nicht überschrieben und manuell hinzugefügte Einträge bleiben in der Datenbank ebenfalls bestehen. Wenn die properties wieder aus den Daten aus der Datenbank geschrieben wird, dann enthalten die neuen properties alle Neuerungen plus die projektspezifischen Anpassungen.



The screenshot shows a sidebar with navigation items: Übersetzungen, Properties, Properties Upload, and Reports. The main area contains a table with columns: Documents, ActionID, ActionID, and Archiving5. Below the table are four buttons: 'NEUER EINTRAG' (with a pencil icon), a cloud upload icon, a cloud upload icon with a plus sign (highlighted with a red box and a tooltip), and a search icon. The tooltip text is 'Aus Properties-Dateien in die DB importieren'.



The screenshot shows a 'Meldung' (Message) window with the title 'Logausgaben für Properties-Dateien'. It contains a table with the following data:

Zeitstempel	Status	Sprache	Log-Text
01.03.2023 17:38:26	INFO		function transferPropertiesFromDbToFile() - Template(ActionID_%lang%.properties)
01.03.2023 17:38:26	INFO	de	Export aus Tabelle 'lang_properties' (ActionID) => ActionID_de.properties
01.03.2023 17:38:26	INFO	de	Es wurde eine neue properties-Datei erstellt (C:\DEXPRO\Documents5\DEXPRO\Locale\Ac
01.03.2023 17:38:26	INFO	de	Die bestehende properties-Datei wurde verschoben und umbenannt (C:\DEXPRO\Documen

A red arrow points from the 'Properties Upload' button in the sidebar to the log entry.

Auch hier können die finalen properties-Dateien direkt für das Produktivsystem verwendet werden.

SQUEEZE

Der Squeeze-Ordner wurde in der Version 2.0.100 in den **DEXPRO**-Ordner verschoben. Der existierende SQUEEZE-Ordner kann nach einem erfolgreichen Update gelöscht werden.

Documents XML Importe

Das Update-Paket enthält nur wenige XML-Import-Dateien!

3. a) b) Update verschlüsselte Portal-Skripte

Die **verschlüsselten Skripte** können gesammelt als XML importiert werden. Hierfür werden passende XML-Import-Dateien bereitgestellt. Es wird jedoch nicht nur der Skript-Inhalt, sondern auch die Skript-Eigenschaften überschrieben und das wirkt sich vor allem auf die Job-Skripte aus!

Die Einstellungen an den konfigurierten **Job-Skripten** sollten vor dem Update notiert werden, denn die Einstellungen werden durch das Update überschrieben!

Welche Skripte laufen als Job und zu welchen Zeiten bzw. in welchem Zeitintervall sind diese Jobs eingeplant?

DEXPRO_CHANGELOG.md

INVOICE_CHANGELOG.md

Update unverschlüsselte Portal-Skripte

Die **UserExit**-Skripte dürfen beliebig projektspezifisch angepasst und erweitert werden. Aus diesem Grund sollten die angepassten Skripte regelmäßig gesichert werden. Durch einen erneuten XML-Import würden alle projektspezifischen Anpassungen verloren gehen! Aus diesem Grund sollten die UserExit-Skripte nur überschrieben werden, wenn garantiert keine Anpassungen vorgenommen wurden.

Der Ordner "**XML SingleExports**" enthält alle **UserExit**-Skripte als Einzel-XML-Export und diese werden pro Skript-Kategorie in einem separaten Unter-Ordner abgelegt. Die noch nicht existierenden Skripte aus den Ordner "**Dexpro_UserExit**" und "**Invoice_UserExit**" sollten importiert werden. Alle weiteren Ordner sind optional.

Die bereits existierenden UserExit-Skripte müssen nicht zwingend angepasst werden!

In den Versionen wurden einige Standard-UserExit-Funktionen sinnvoll erweitert. Bei den folgende Anpassungen wird eine Übernahme empfohlen:

- **EXPRO__UserExit_TechActionLib**

In der aktuellen Version enthalten die Funktionen einen Parameter "**log**". Bei den beim Kunden verwendeten Funktionen sollte der Parameter ergänzt werden. Vorhandene `util.out()` sollten durch `log.info()` bzw. `log.err()` ersetzt werden. Das erleichtert die Supportfälle.

In der Funktion **ue_Archiving()** wird vor der Archivierung die sqz-Datei vom Squeeze-

Register entfernt und das Register wird ausgeblendet. Nach der Archivierung wird der **ArchivKey** der Mappe in die Tabelle **Invoice_Posting_Head** geschrieben. Diese Parts sollten - wenn noch nicht vorhanden - in die bestehende Funktion übernommen werden.

Für das Update im Produktivsystem können direkt die angepassten UserExit-Skripte inklusive aller Anpassungen aus dem Testsystem exportiert werden. Über die Export-Klasse "**PortalScript**" lassen sich alle Skripte auf einmal exportieren. alternativ kann auch eine gesamte Scripting-Kategorie exportiert werden.

4. Update Workflow

Der Workflow muss nur importiert werden, wenn sich die Versionsnummer geändert hat. Nach dem Update des Mappentypen muss der aktuelle Workflow als Standard-Workflow hinterlegt werden. So lange wie einige Mappen noch ältere Workflow-Versionen verwenden, müssen diese Workflows unter "Zulässige Workflows" gelistet sein!

Wenn die Documents-Lizenz keine freie Workflow-Lizenz enthält muss der alte Workflow deaktiviert werden. Die bestehenden Mappen laufen trotzdem mit dem alten Workflow weiter. Es können nur keine neuen Workflows mit der alten Workflow-Version gestartet werden. Neu angelegte Mappen starten mit der neuen Version.

7. Update Outbar Administration

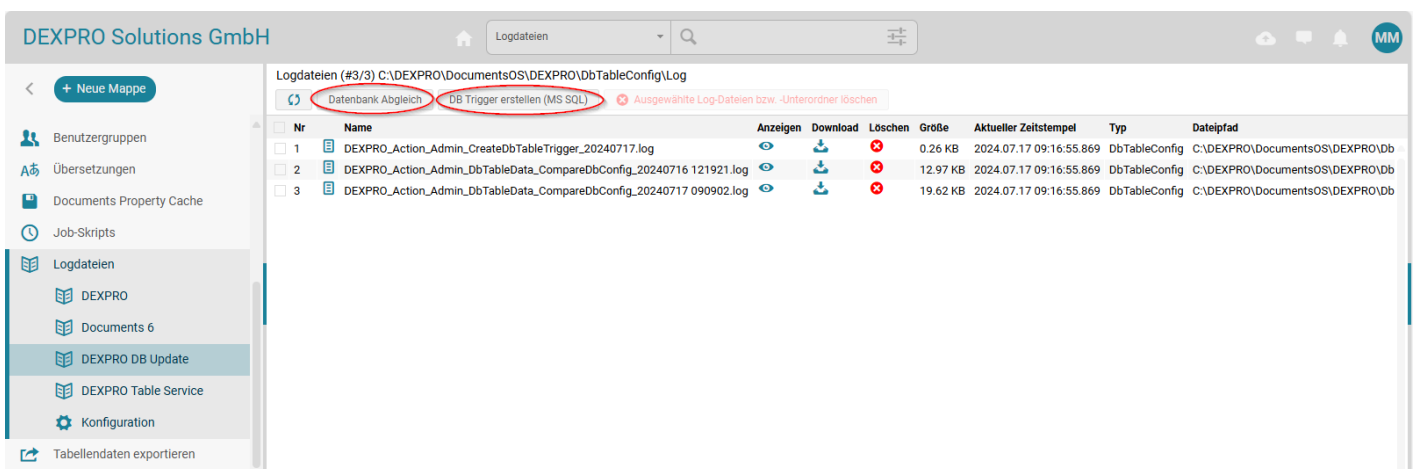
Bei den öffentlichen Ordnern muss zwischen den Outbars **Invoice** und den Administrations-Outbars unterschieden werden. Die Ordner auf der Outbar Invoice wurden bereits im Projekt passend für den Kunden ein- und ausgeblendet und gegebenenfalls wurden projektspezifische Ordner hinzugefügt; benutzerdefinierte Aktionen an Ordnern hinzugefügt und Zugriffsberechtigungen gesetzt. Die komplette Konfiguration entspricht in der Regel bereits genau dem, was der Kunde sehen möchte und durch ein Update auf die neueste Invoice-Version würden größtenteils nicht benötigte Ordner importiert. Hier ist die Empfehlung nur sinnvolle und vom Kunden geforderte Neuerungen manuell aus dem aktuellen Standard zu übernehmen.

Bei der **Administrations**-Outbar werden hingegen häufig nur wenige bis gar keine Anpassungen vorgenommen. Hier empfiehlt es sich die Update-XML zu importierten. Dabei werden bestehende Outbars und Ordner überschrieben und neue Ordner werden hinzugefügt. Ggf. müssen im Anschluss nicht benötigte Admin-Outbars deaktiviert werden und einzelne Ordner müssen wieder passend ein- oder ausgeblendet bzw. an die korrekte Stelle verschoben werden.

Update Datenbank-Tabellen

Abgleich-Skript

Auf der "**Administrations**"-Outbar befindet sich auf der Sub-Outbar "**Allgemein**" der Ordner "**Logdateien**" mit dem Unterordner "**DEXPRO DB Update**". Hierüber wird das Skript "**DEXPRO_Action_Admin_DbTableData_CompareDbConfig**" ausgeführt. Alternativ kann das Skript auch über den Manager ausgeführt werden. Die zugehörige Log-Datei wird nach einer Aktualisierung der Ansicht direkt im Ordner angezeigt. Das Skript benötigt etwas Zeit und ggf. muss die Ansicht erneut aktualisiert werden. Über das Augen-Symbol kann die Logdatei direkt angesehen werden. Abweichungen werden als Fehler **rot** markiert.



Das Portal-Skript liest die aktuell beim Kunden vorhandene Datenbank aus und sucht nach fehlenden Tabellen und Tabellen-Spalten und legt diese an. Zudem werden alle Tabellen-Spalten auf Datentyp, Default-Werten und der Einstellung ob NULL-Werte zulässig sind überprüft. Zusätzlich werden Trigger und Primary- sowie Unique-Keys überprüft. Fehlende Datenbank-Trigger unter MS SQL können über das Skript "**DEXPRO_Action_Admin_CreateDbTableTrigger_All**" hinzugefügt werden.

Das Skript kann viele Abweichungen wie fehlende Spalten oder Tabellen direkt beheben. Einige Abweichungen wie Veränderungen bei einem Unique-Key müssen manuell behoben werden. Hinweise hierzu folgen weiter unten auf dieser Seite.

Anpassungen an einer Tabellen-Spalte schlagen jedoch fehl, wenn die Spalte in einem Unique-Key verwendet werden. Unique-Keys werden aktuell nicht automatisch erstellt oder angepasst. Somit kann es sein, dass manuell nachgearbeitet werden muss.

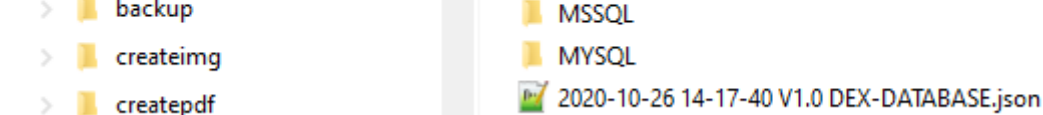
Wenn das Skript auch nach mehrfacher Ausführung Abweichungen nicht automatisch auflösen kann, müssen die Anpassungen manuell hergestellt werden. Das Skript kann nach jeder Anpassung erneut ausgeführt werden. Der Vorgang kann wiederholt werden, bis die Log-Datei

keine weiteren Abweichungen entdeckt. Hilfestellungen zu den Fehlern werden weiter unten gegeben.

Technischer Hintergrund

Für das Update der Datenbanken enthält der "**DEXPRO**"-Ordner den Unterordner "**DbTableConfig**". Dieser enthält wiederum einen "Log-Ordner" und die Ordner "**MSSQL**" und "**MYSQL**" sowie eine JSON-Datei "**DEX-DATABASE**" mit der Versionsnummer und einem Zeitstempel im Namen. Mit jedem Update wird eine neue JSON-Datei ausgeliefert. Die alten Dateien sollten aus dem Ordner entfernt werden, damit garantiert die neueste Version verwendet wird! Die Datei enthält die komplette Datenbank-Struktur im Soll-Zustand.

Die Ordner "**MSSQL**" und "**MYSQL**" enthalten wiederum Unterordner für die beiden Datenbanken "**DEX_MasterData**" und "**DEX_Workflow**", in welchen pro Tabelle einzelne SQL-Dateien mit den CREATE-Befehlen der Tabellen enthalten sind.



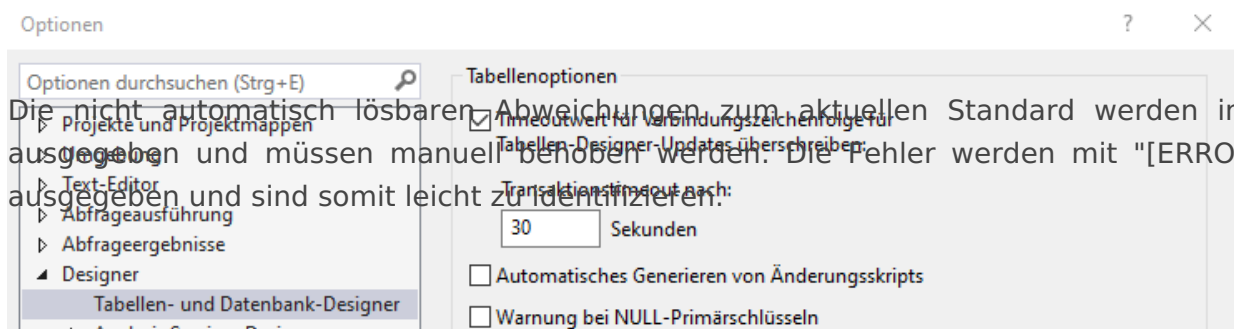
Datenbank-Berechtigungen

Der ausführende Benutzer muss Datenbank-Tabellen hinzufügen können und er muss bestehende Tabellen ändern können!

In jedem Fall muss ein Zugriff mit entsprechenden Rechten auf die beiden Datenbanken gegeben sein! Bei MS SQL zum Beispiel über ein Management Studio.

Manuelle Anpassungen an den Datenbanken

Auch wenn viele Anpassungen automatisch über das folgende Skript ausgeführt werden kann es sein, dass manuelle Anpassungen an bestehenden Tabellen vorgenommen werden müssen. Bei MS SQL ist dies in der Voreinstellung häufig deaktiviert. Die Einstellung "Speichern von Änderungen verhindern, die die Neuerstellung der Tabelle erfordern" muss hierfür aktiviert werden. Die Einstellung befindet sich unter "Extras" -> "Optionen" im Abschnitt "Designer" -> "Tabellen- und Datenbank-Designer". Die Checkbox muss wie im Screenshot deaktiviert sein.



Die nicht automatisch lösbaren Abweichungen zum aktuellen Standard werden im Skript-Log ausgegeben und müssen manuell behoben werden. Die Fehler werden mit "[ERROR]" markiert ausgegeben und sind somit leicht zu identifizieren.

Datentyp-Abweichung korrigieren

Ein abweichender Datentyp wird im Log als Fehler angezeigt - muss aber nicht zwingend ein Fehler sein. Besonders in den Tabellen "Invoice_Posting_Head" und "Invoice_Posting_Pos" kann es sein, dass projektspezifisch bewusst Anpassungen auf sehr spezifische Typen vorgenommen wurden.

Im folgenden Beispiel wurde aus einer "int"-Spalte eine "bigint"-Spalte. Ohne die oben beschriebene Designer-Konfiguration würde beim Speichern ein Fehler erscheinen, da die Tabelle beim Speichern neu erstellt werden muss.

[ERROR] [fields pos definition][ID] divergent data type .. standard:bigint != custom:int

Microsoft SQL Server Management Studio

Key-Abweichung korrigieren

Wie bereits erwähnt werden Abweichungen bei Unique-Schlüsseln nicht automatisch korrigiert. Bei einer MySQL Installation über eine MariaDB können die Unique-Keys genau so wie die Primärschlüssel direkt über die Tabellen-Konfiguration konfiguriert werden.

Spaltenname	Datentyp	NULL-Werte...
ID	bigint	
Licence	varchar(50)	
TemplateFile	varchar(50)	

Im Microsoft SQL Server Management Studio muss die Tabelle zunächst im Entwurfs-Modus geöffnet werden. In der Tabellen-Ansicht kann aber nur der Primärschlüssel für die Tabelle festgelegt werden. Durch das Öffnen der Tabelle erscheinend in der oberen Leiste der Reiter "Tabellen-Designer". Hier befindet sich der Eintrag "Indizes/Schlüssel...".

In dem Pop-Up werden die Primärschlüssel und die eindeutigen Schlüssel definiert.

Im folgenden Screenshot ist eine Fehlerausgabe im Log zu sehen, bei der in der Tabelle "fields_pos_definition" ein bestehender Unique-Schlüssel um die Spalte "TemplateFile" erweitert wurde. Der technische Name des Unique-Keys ist nicht relevant, denn das Skript sucht nach einem Schlüssel mit denselben Spalten unabhängig vom Namen.

Tabellen-Designer

Primärschlüssel entfernen

Eigenschaften für Primärschlüssel/eindeutiger Schlüssel oder Index.

PK Fields_Pos_Definition 1
Unique_Fields_Pos_Definition

▼ (Allgemein)

```
[2020.11.04 14:46:53][ERROR] [fields_pos_definition] function compareTableKeys( standardKeysObj, customKeysObj, fields_pos_definition ) .. missing specific key 'licence_techfieldname'
```

```
26 [2020.11.04 14:46:53][ERROR] [fields_pos_definition] function compareTableKeys( standardKeysObj, customKeysObj, fields_pos_definition ) .. missing specific key 'licence_techfieldname'
```

```
27 .. standardKeysObj[uniqueKeys]: {
```

```
28 {
```

```
29   "desc": "Licence_TechFieldName",
```

```
30   "columns": {
```

```
31     "Licence",
```

```
32     "TechFieldName",
```

```
33     "TemplateFile"
```

```
34   }
```

```
35 }
```

```
36 }
```

```
37 .. customKeysObj[uniqueKeys]: {
```

```
38 {
```

```
39   "desc": "Unique_Fields_Pos_Definition",
```

```
40   "columns": {
```

```
41     "Licence",
```

```
42     "TechFieldName"
```

```
43   }
```

```
44 }
```

```
45 }
```

Wenn der Unique-Key noch nicht existieren sollte, dann muss zunächst ein neuer Schlüssel über den Button "Hinzufügen" erzeugt werden. Der Schlüssel muss vom Typen "Eindeutiger Schlüssel" sein. Wenn ein Schlüssel existiert aber nicht gefunden werden konnte, dann wurde evtl. ein falscher Schlüssel zugeordnet.

Unter "Identität" sollte ein sprechender Name für den Schlüssel angegeben werden. Bei Neuanlage beginnt der vordefinierte Name mit "IX", da als Typ "Index" voreingestellt ist. Aber auch wenn man

den Typen auf "Eindeutiger Schlüssel" ändert, wird das Präfix nicht angepasst. In unserer Auslieferung beginnen die Unique-Keys bei MS SQL mit "Unique", damit man den Typen des Schlüssels direkt über den Namen ableiten kann.

Unter "Spalten" können die Tabellen-Spalten zugeordnet werden. In diesem Beispiel muss die Spalte "TemplateFile" hinzugefügt werden. Das Pop-Up muss geschlossen werden. Das Speichern erfolgt im Anschluss durch das Speichern der Tabelle über das Disketten-Symbol.

DESKTOP-10PUD4K\S...ds_Pos_Definition

Spaltenname	Datentyp	NULL-Werte...
ID	bigint	<input type="checkbox"/>
TemplateFile	nvarchar(50)	<input type="checkbox"/>

Null-Werte (nicht) zulassen

Indizes/Schlüssel

Primärschlüssel/eindeutiger Schlüssel oder Index (ausgewählt):
PK_Fields_Pos_Definition_1
DateFormat (nvarchar(50))

Eigenschaften für Primärschlüssel/eindeutiger Schlüssel oder Index
ist eindeutig: Ja
Spalten: Licence (ASC), TechFieldName (ASC)

Index: Spalten

Geben Sie die Spalten und die Sortierreihenfolge für diesen Index an:

Spaltenname	Sortierreihenfolge
TechFieldName	Aufsteigend
TemplateFile	Aufsteigend

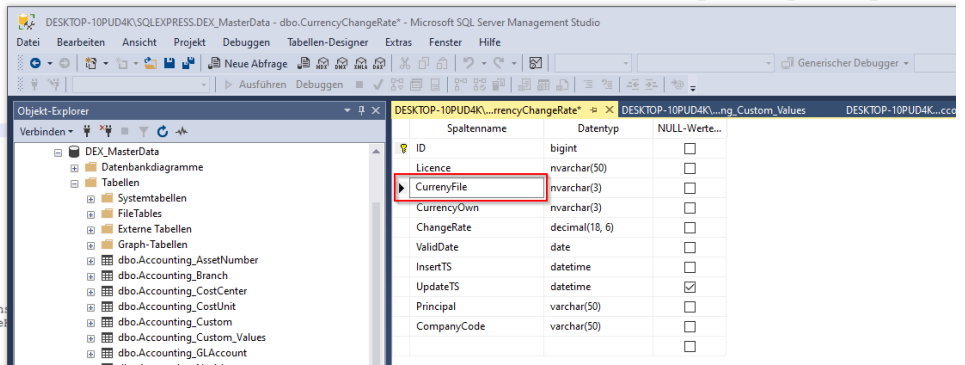
```
77 [2020.11.04 15:02:55][INFO] [lang_activelanguages][DateFormat] divergent 'IS NULLABLE' .. standard:NO != custom:YES
78 [2020.11.04 15:02:55][ERROR] [lang_activelanguages][DateFormat] could not alter nullable: [INFO] db_TableObj(DEX_Workflow, lang_activelanguages)
79 [INFO][MSSQL] USE Dex_Workflow SELECT TABLE_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE (TABLE_CATALOG='Dex_Workflow') AND (TABLE_NAME='lang_activelanguages')
80 [INFO] function alterNullable({
81   "COLUMN_NAME": "DateFormat",
82   "ORDINAL_POSITION": 7,
83   "DATA_TYPE": "varchar",
84   "TYPE_INTERNAL": "varchar",
85   "COLUMN_DEFAULT": "'dd.mm.yyyy'",
86   "IS_NULLABLE": "NO",
87   "CHARACTER_MAXIMUM_LENGTH": 50,
88   "NUMERIC_PRECISION": null,
89   "NUMERIC_SCALE": null
90 }) .. this.UseSQLType(MSSQL)
91 [ERROR] Fehler bei der Ausführung eines SQL-Statements
92 SQL error in 'ALTER TABLE Dex_Workflow.dbo.lang_activelanguages ALTER COLUMN DateFormat nvarchar(50) NOT NULL':
93 ReturnCode: -1
94 SqlState: 37000
95 NativeError: 5074
96 ErrorMsg: [Microsoft][SQL Server Native Client 11.0][SQL Server]Objekt 'DF_Lang_ActiveLanguages_DateFormat' ist von Spalte 'DateFormat' abhängig.
```

Die Spalte muss manuell über die Tabellen-Konfiguration geändert werden. Wenn NULL-Werte nicht mehr zugelassen werden, dann darf es in der Tabelle keinen Eintrag mit NULL geben. Das sollte in solchen Fällen immer gegeben sein, denn sonst würde die Spalte nicht entsprechend umgestellt werden. Sollte es dennoch Einträge mit NULL geben, muss ggf. manuell angepasst werden.

Spalten hinzufügen

In wenigen Fällen können Spalten nicht automatisch hinzugefügt werden. Im folgenden Beispiel war ein Schreibfehler bei einem Spaltennamen enthalten, wodurch gleichzeitig eine Abweichung beim Unique-Key entdeckt wird. Durch die Korrektur des Spaltennamens werden automatisch 2 Fehler behoben.

```
153 [2020.11.04 15:02:55][ERROR] [currencychangerate] could not add standard column 'CurrencyFile': [INFO] db_TableObj(Dex_MasterData, currencychangerate)
154 [INFO][MSSQL] USE Dex_MasterData SELECT TABLE_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE (TABLE_CATALOG='Dex_MasterData') AND (TABLE_NAME='currencychangerate')
155 [INFO] function addColumn(
156   "COLUMN_NAME": "CurrencyFile",
157   "ORDINAL_POSITION": 5,
158   "DATA_TYPE": "nvarchar",
159   "TYPE_INTERNAL": "nvarchar",
160   "COLUMN_DEFAULT": null,
161   "IS_NULLABLE": "NO",
162   "CHARACTER_MAXIMUM_LENGTH": 3,
163   "NUMERIC_PRECISION": null,
164   "NUMERIC_SCALE": null
165 ) .. this.UseSQLType(MSSQL)
166 [ERROR] Fehler bei der Ausführung eines SQL-Statements
167 SQL error in 'ALTER TABLE Dex_MasterData.dbo.currencychangerate ADD CurrencyFile nvarchar(3) NOT NULL':
168 Error while retrieving Error Data
169 [INFO] function addColumn() .. return false
170
171 [2020.11.04 15:02:55][ERROR] [currencychangerate] function compareTableKeys( standardKeysObj, customKeysObj, currencychangerate ) .. missing specific key 'licence_currencyfile_currencyown_validate'
172 .. standardKeysObj[uniqueKeys]: {
173   {
174     "desc": "Licence_CurrencyFile_CurrencyOwn_ValidDate",
175     "columns": [
176       "CurrencyFile",
177       "CurrencyOwn",
178       "Licence",
179       "ValidDate"
180     ]
181   }
182 }
183 .. customKeysObj[uniqueKeys]: {
184   {
185     "desc": "Unique_CurrencyChangeRate",
186     "columns": [
187       "CurrencyOwn",
188       "CurrencyFile",
189       "Licence",
190       "ValidDate"
191     ]
192   }
193 }
194 [2020.11.04 15:02:55][INFO] [customer] function compareColumns
195 [2020.11.04 15:02:55][ERROR] [customer] function compareTableKeys
196 .. standardKeysObj[uniqueKeys]: {
197   {
198     "desc": "Licence_Principal_CompanyCode_CustomerID"
```

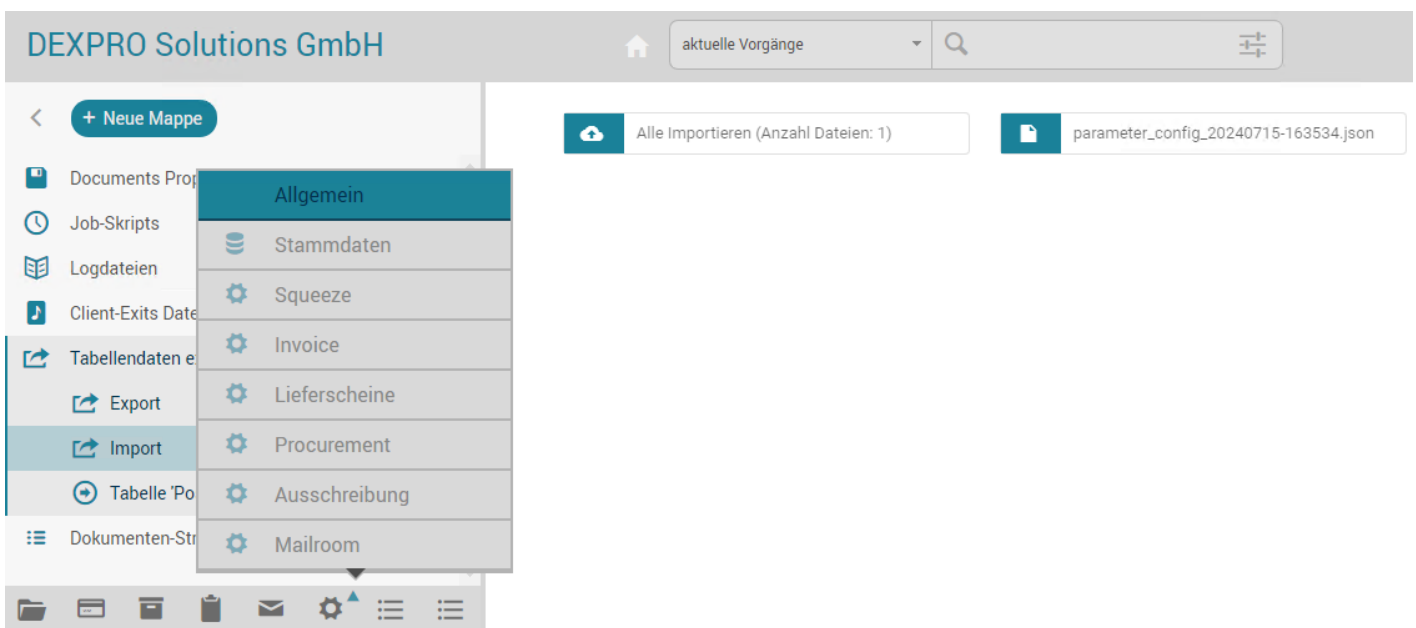


Nachdem alle Anpassungen vorgenommen wurden sollte das Skript erneut ausgeführt werden und das Log sollte erneut kontrolliert werden. Erst wenn keine relevanten Abweichungen mehr entdeckt werden, sind die Anpassungen abgeschlossen.

Parameter importieren

Bei einem Update muss lediglich die Konfiguration für die Parameter überschrieben werden. Die "**parameter_config*.json**" enthält alle Parameter-Definitionen. Die Datei kann über die WEB-Administration "**Tabellendaten exportieren**" -> "**Import**" importiert werden. Weitere Konfigurationen sollen nicht importiert werden!

Durch den Import wird alle Tabellendaten gelöscht und neu geschrieben und projektspezifische Parameter gehen verloren und müssen manuell wieder angelegt werden!



Documents Abgleich

Mappentypen

otrAccessProfile / otrUser

Die Mappentypen **otrAccessProfile** und **otrUser** werden für gewöhnlich nicht angepasst. Diese Mappentypen werden u. a. auch im Contract verwendet. Sollte ebenfalls ein Contract installiert sein, dann sollten die Mappentypen nicht überschrieben werden! Andernfalls können die Mappentypen über die Update-XML überschrieben werden. Projektspezifische Anpassungen müssen manuell wieder hergestellt werden.

Invoice

Am Mappentypen **Invoice** finden im Projekt viele Anpassungen angepasst. Felder werden hinzugefügt; die Reihenfolge der Felder wird angepasst und die Feldeigenschaften werden verändert; Felder werden den Documenten-Registern hinzugefügt oder vom Register entfernt; Trefferlisten und Suchmasken werden für den Kunden optimiert; benutzerdefinierte Aktionen werden hinzugefügt. Ein Mappentyp-Update würde viele dieser Anpassungen wieder auf den Standard zurücksetzen und alle Einstellungen müssten sehr zeitaufwändig wieder hergestellt werden.

Von einem Mappentyp-Update wird dringend abgeraten!

Es werden immer mal wieder neue Felder am Standard-Mappentypen hinzugefügt. In der Regel werden die neuen Felder nicht zwingend benötigt und müssen nicht zwingend hinzugefügt werden - hier muss im Einzelfall die Notwendigkeit abgewogen werden.

Im Laufe der Entwicklung hinzugefügte Felder sind zum Beispiel:

- **CostCenter / CostCenterDesc**
Kostenstelle im Kopf inklusive Pop-Up und Autocomplete-Funktion.
- **HR_PROJECT / ProjectRef / ProjectIDInt**
Verknüpfung zum Mappentypen "ProjectFile". Wird nur benötigt, wenn Rechnungen zu Projekt-Akten verknüpft werden sollen.
- **PaymentAmount / PaymentID / PaymentList**
Der Zahlungs-Informationen werden generell nur benötigt, wenn im Workflow nach der Buchung auf die Zahlung gewartet wird.
- **HR_SQUEEZE / SqueezeStatus / SqueezeID**

Wird für den Support bei zur Kommunikation mit Squeeze benötigt. Die Felder sollten ergänzt werden, wenn Sie noch nicht vorhanden sind.

- **HR_BREEZE / BreezeID / CoreID / NavisionMetadata / NavisionDimensionData**

Die Felder werden zur Kommunikation mit Navision BC365 benötigt.

- **AutoValidation**

Die Checkbox kann in den Workflow-Regeln verwendet werden um die Validierung zu überspringen, wenn Squeeze bereits alle benötigten Daten erkannt hat. Die zugehörigen Funktion "**docFile.checkAutoValidation()**" wird in der "DEXPRO__UserExit_InvoiceLib" ausgeliefert.

- **GoodsReceiptID / OrderConfirmationID**

Der Invoice-Mappentype kann auch für Wareneingänge oder Auftragsbestätigungen verwendet werden. Durch das Ein- und Ausblenden von Feldern über die Feldkonfiguration und der Einrichtung neuer Workflows können auch andere Belegtypen abgebildet werden.

Die benutzerdefinierten Aktionen können wie die Feldliste abgeglichen werden. Hier sollten alle nicht vorhandenen benutzerdefinierten Aktionen ergänzt werden. Neue benutzerdefinierte Aktionen werden standardmäßig ausgeblendet, solange sie nicht explizit im Skript "**Invoice__UserExit_DF_ShowUserDefinedActions**" definiert werden.

Neue Mappeneigenschaften können - müssen aber nicht zwingend übernommen werden. Bei älteren Versionen wurde bei den Eigenschaften "**ArchiveMonitorAsPdf**" und "**ArchiveStatusAsPdf**" der Wert "**true**" gesetzt - es muss aber zwingend die "**1**" sein. Dies sollte korrigiert werden.

Wenn nach den Anpassungen ein "**Mappen ändern**" durchgeführt werden muss, dann werden alle durch den Workflow angepassten Feldeigenschaften auf die Standardwerte zurückgesetzt. In diesem Fall muss der JOB "**Invoice_JOB_SetFieldConfig**" ausgeführt werden. Dieser führt für alle Invoice-Akten die Aktion "Feldkonfiguration laden" aus.

Individuelle Versions- Anpassungen / Wichtige Hinweise

Bei einigen Versions-Sprüngen müssen bestimmte zusätzliche Anpassungen vorgenommen werden bzw. müssen wichtige Hinweise berücksichtigt werden. Auf dieser Seite werden die wichtigsten Informationen zu den Versionen mitgeliefert. Infos zu allen Änderungen befinden sich wie immer im Changelog.

2.0.400

Beim Aussteuern werden die Vorgänge nicht mehr direkt archiviert und gelöscht. Nach der Archivierung werden die Vorgänge nicht mehr in den Papierkorb verschoben, sondern gelöscht.

Das alte Verhalten kann über das Skript **DEXPRO_Action_Disqualify_ArchiveAndDelete** und den Parameter **Disqualify_DeleteOptions** eingestellt werden. Genauere Informationen befinden sich im ChangeLog.

2.0.400

Das Skript **DEXPRO_UserExit_Escalation** wurde in **Invoice_UserExit_EscalationLib** umbenannt. Bei einem Update muss der Skript-Inhalt manuell kopiert werden und das alte Skript muss gelöscht werden.

2.0.200 / Documents6

Die Version wurde mit Documents6 getestet.

2.0.200

Nach einem Update auf Documents6 kann es sein, dass der **Squeeze-Viewer** geblockt wird. Hierfür muss die globale Eigenschaft "**csp-config**" vom Typen "**csp-config**" gesetzt werden. Im Anschluss muss der Tomcat neu gestartet werden - bzw. kann "**&cacheEvent=clearAll**" angewendet werden.

```
{
```

```
"routes": {
  "documents": {
    "source-hosts": ["https://my1.custom.example.de", "https://my2.custom.example.de"]
  },
  "login": {
    "source-hosts": []
  }
}
}
```

2.0.200

Bei einem Update auf **Documents6** muss in der **dbConn.json** der Eintrag "**Documents5**" in "**Documents6**" umbenannt werden und der zugehörige Eintrag "**database**" muss von "**documents5**" auf "**documents6**" umgestellt werden! Bei einer Neuinstallation wird die dbConn.json bereits mit dem "Documents6"-Eintrag ausgeliefert. Wenn eine Installation unter Documents5 erfolgt, muss der Eintrag wieder auf "5" umgestellt werden.

2.0.200

Einige **Table-Service-Konfigurationen** wurden umbenannt und wären bei einem Update doppelt vorhanden.

Dadurch lässt sich der TableService-Dienst nicht mehr starten.

Der Inhalt des folgenden Ordners muss einmal gelöscht werden und die Dateien für diese Ordner müssen aus dem Update nochmal übertragen werden:

...\DEXPRO\TableService\data\table-definitions\Workflow

2.0.200

Falls der Umzug des Squeeze-Ordners noch nicht durchgeführt wurde (Version 2.0.100), müssen die folgenden Dateien aus dem Install-Verzeichnis manuell kopiert werden:

...\DEXPRO\Vue\TableService.js

...\DEXPRO\Squeeze\SqueezeConfig.js

2.0.100

Der **Squeeze-Ordner** wurde in den DEXPRO-Ordner verschoben. Zudem ist ein **Vue-Ordner** hinzugekommen.

Die Anbindung an den TableService wurde technisch von der Squeeze-Konfiguration getrennt. Die Pfadanpassungen müssen in der documents.xml im Tomcat-Ordner übernommen werden!

Siehe hierzu die aktuelle Installations-Anleitung.

Der alte Squeeze-Ordner kann nach erfolgreichem Update gelöscht werden.

2.0.100

Durch den Umzug muss die **documents.xml** im Tomcat-Ordner angepasst werden. Hierzu Bitte die Vorlage aus dem Install anschauen.

Bei einer Umstellung auf den **Tomcat9** muss die Datei im Ordner "**tomcat9\conf\Catalina\localhost**" abgelegt werden.

2.0.100

Die **GridColumnAggregator.js** wurde in den DEXPRO_ClientExits - Ordner verschoben.

Die bestehende Datei im Ordner ...**DEXPRO\ScriptExits\DexLibs** muss manuell gelöscht werden!

2.0.100

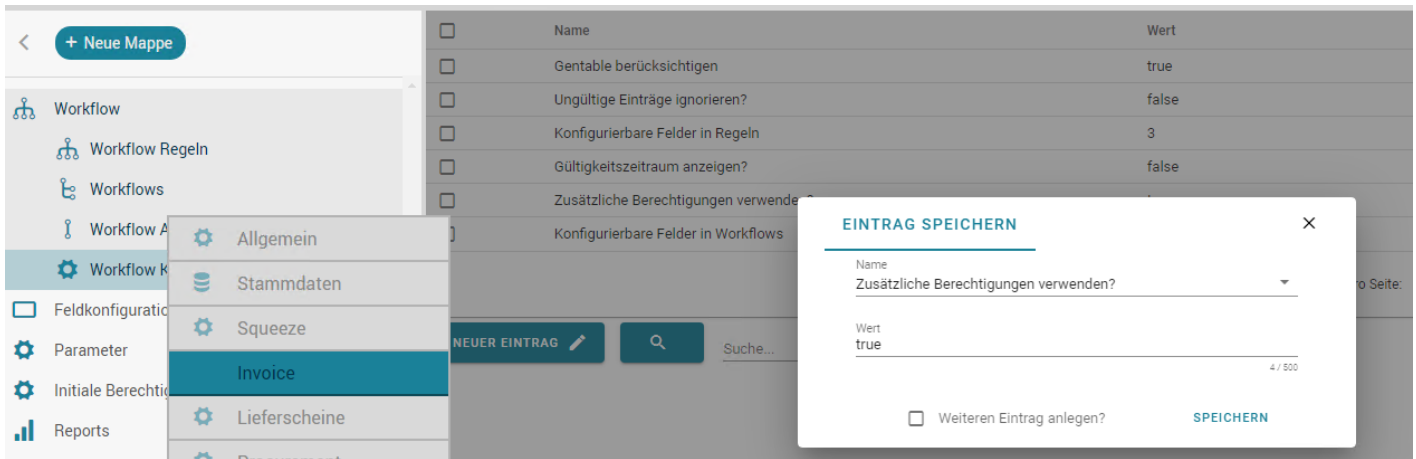
In **Documents6** sind clientseitige Skript-Ausführungen nur möglich, wenn die Skripte explizit dafür freigegeben sind.

An den entsprechenden Standard-Skripten wurde die Eigenschaft '**clientExecutable**' mit dem Wert '**true**' gesetzt.

Bei den User-Exit-Skripten und den projektspezifischen Skripten muss die Eigenschaft manuell ergänzt werden.

1.1.300

Die Workflow-Regeln können manuell um die Spalte "**Zusätzliche Berechtigungen verwenden?**" erweitert werden.



1.1.200

Aktivierung/Umstellung auf **Squeeze Viewer 2.0** möglich!

1.1.015

Ab dieser Version muss Documents zwingend in der Version **#2311** oder höher vorliegen!

1.1.009

Das Schreiben in die Datenbank kann über den Parameter "**GentableSuppressWritingDataIntoDb**" unterdrückt werden.

1.1.005

Workflow Version "**WorkflowRules-7**"

1.0.300

Für das Modul wird eine **DEXPRO-Lizenz** benötigt!