

# Redaktions- System 2.5

exportiert am 16-09-2022

# Inhaltsverzeichnis

<b>Rechtliche Hinweise</b> .....	<b>9</b>
<b>Was ist neu in Version 2.5</b> .....	<b>10</b>
<b>Einführung</b> .....	<b>11</b>
Umgebung des R-S .....	14
Die Benutzertypen .....	22
<b>Nutzer-Dokumentation</b> .....	<b>23</b>
Grundlagen .....	28
Status - Workflows und Versionierung .....	29
Status einer Objekt-Version setzen .....	37
Arbeitsbereiche .....	41
Ressource-Typen .....	42
Variablen .....	44
RichText.....	48
Binäre Ressourcen.....	52
Regeln.....	55
Physikalische Beileger .....	66
Logische Beileger .....	69
WhiteSpaces .....	72
Datenstrom-Variablen .....	78
Basisobjekte im R-S .....	82
Transaktions-Formulare .....	84
Schemas .....	87
Produktions-Pakete .....	90
Bibliotheken .....	93
Material .....	99
Docxworld-Links .....	102
Produktions-Paket Verknüpfungen .....	105
Erweiterungen .....	108
Erste Schritte .....	110
Anmeldung am R-S Client.....	111
Nutzeroberfläche und Navigation .....	113
Nutzeroberfläche des R-S .....	114
Bedienung des R-S.....	126
Dashboard.....	135
Arbeiten mit Transaktions-Formularen .....	136

Formular erstellen, ändern und löschen.....	138
Formular-Version .....	140
Formular-Version erstellen, ändern und löschen .....	141
Formular-Version kopieren .....	143
Status einer Formular-Version setzen.....	145
Registerkarten bei Formular-Versionen .....	147
Arbeiten mit Schemas .....	172
Schema erstellen, ändern und löschen .....	174
Schema-Version.....	176
Schema-Version erstellen, ändern und löschen .....	177
Schema-Version kopieren .....	179
Status einer Schema-Version setzen .....	181
Schema-Version exportieren .....	183
Schema-Version importieren.....	184
Registerkarten bei Schema-Versionen .....	186
Arbeiten mit Bibliotheken .....	219
Bibliotheken erstellen, ändern und löschen .....	221
Bibliotheks-Version.....	223
Bibliotheks-Version erstellen, ändern und löschen .....	225
Bibliotheks-Version kopieren .....	227
Status einer Bibliotheks-Version setzen .....	229
Bibliotheks-Version exportieren .....	232
Bibliotheks-Version importieren.....	233
Bibliotheks-Ressourcen .....	235
Ressourcenverwaltung .....	237
Ressourcen erstellen und ändern .....	239
Binäre Ressource erstellen und ändern .....	241
Variable erstellen und ändern .....	244
Regel erstellen und ändern .....	246
Physikalischen Beileger erstellen und ändern .....	256
Logischen Beileger erstellen und ändern.....	259
WhiteSpace erstellen und ändern .....	274
RichText erstellen und ändern.....	278
Erforderliche Ressource erstellen und ändern.....	283
Ressourcen löschen.....	286
Ressourcen duplizieren .....	288
Ressourcen zurücksetzen .....	290
Ressourcen exportieren.....	292
Ressourcen importieren .....	294
Ressourcen Zustände.....	296

Arbeiten mit Produktions-Paketen .....	297
Paket ändern.....	298
Registerkarte - Details in Produktions-Pakete .....	299
Registerkarte - Historie .....	300
Registerkarte - Vorschau in Produktions-Pakete .....	301
Registerkarte - Test-Zyklus .....	303
Registerkarte - Produktions-Zyklus .....	304
Registerkarte - Paket-Prüfung.....	305
Status eines Pakets setzen .....	306
Arbeiten mit Materialien.....	312
Material für physikalischen Beileger erstellen, ändern und löschen .....	314
Material für logischen Beileger erstellen, ändern und löschen .....	318
Arbeiten mit Docxworld-Links .....	322
Docxworld-Links erstellen, ändern und löschen .....	323
Arbeiten mit Produktions-Paket Verknüpfungen .....	327
Verknüpfung erstellen und ändern.....	328
Mehr zur Arbeit mit Objekten .....	332
Löschen von Objekten .....	333
Wechseln des Modus .....	335
EVAL-Anweisungen, Operatoren und Datentypen.....	337
Funktionen in EVAL-Anweisungen.....	341
INLINE-Sequenzen .....	393
b .....	395
cn .....	396
dn .....	397
f .....	398
fn.....	399
fs .....	400
i .....	401
n .....	402
normal .....	403
ol .....	404
plain.....	405
sl.....	406
t .....	407
u+ .....	408
ul .....	409
up.....	411
<b>Administrator-Dokumentation.....</b>	<b>412</b>

Administrator-Client UI .....	414
Anmeldung am R-S Admin-Client.....	415
Nutzeroberfläche des RS-Admin-Clients .....	417
Funktionen des Admin-Clients .....	418
Navigation - Nutzer .....	419
Navigation - Arbeitsbereiche .....	428
Schriftarten.....	446
Binärpakete.....	449
Administrator-Client Konsole .....	452
Installation .....	460
System-Voraussetzungen .....	462
R-S Client .....	463
R-S Server .....	464
Datenbanken .....	465
Datenbank - Derby.....	466
Datenbank - MySQL.....	467
Systemsicherheit .....	469
Verwendete Software .....	471
System-Berechtigungen .....	472
Arbeitsbereich-Administrator .....	473
Nutzer, Nutzer-Gruppen und Rechte .....	474
Nutzer verwalten .....	475
Rechte.....	477
Gruppen .....	483
System-Nutzer.....	485
Systembetrieb .....	486
Backup.....	487
Binär-Pakete .....	488
Aufbau des ZIP-Binär-Paketes.....	489
Aufbau der Binär-Paket Beschreibungs-Datei.....	490
Programme in Binär-Paketen .....	493
HTTP-Status-Codes durch Client- oder Server-Fehler im WEB-Browser .....	498
Restore .....	500
R-S Server Konfigurationsdatei.....	502
<b>Anleitungen .....</b>	<b>513</b>
Logische Beileger des R-S über SupplementWorkitem des ReportWriters .....	514
Optionale Beileger und Porto.....	516
TCI des ReportWriters .....	517
Konfigurationen Transaktions-Formulars .....	520

Konfiguration im R-S.....	521
Beispiel - optionale Beileger.....	524
Algorithmen .....	525
<b>Glossar .....</b>	<b>526</b>
A.....	526
B.....	527
C.....	528
D.....	528
E.....	530
F.....	531
G.....	531
H.....	532
I.....	532
J.....	533
K.....	533
L.....	533
M.....	534
N.....	534
O.....	535
P.....	535
Q.....	536
R.....	536
S.....	537
T.....	538
U.....	539
V.....	539
W.....	540
X.....	540
Y.....	540
Z.....	541
<b>Hinweise zur Online-Dokumentation .....</b>	<b>542</b>
Verwendete Symbole.....	543
<b>Weiterführende Informationen.....</b>	<b>545</b>
Sitemap .....	546
Rechtliche Hinweise.....	546
Was ist neu in Version 2.5 .....	546
Einführung .....	546
Nutzer-Dokumentation.....	546
Administrator-Dokumentation.....	549

<b>Anleitungen .....</b>	<b>550</b>
<b>Glossar .....</b>	<b>550</b>
<b>Hinweise zur Online-Dokumentation .....</b>	<b>550</b>
<b>Weiterführende Informationen.....</b>	<b>550</b>
<b>Download der Dokumentation.....</b>	<b>551</b>
<b>Online-Archiv dieses Produkts .....</b>	<b>552</b>
<b>Ergänzende Dokumentationen.....</b>	<b>553</b>
<b>Service und Support .....</b>	<b>554</b>

## **Herzlich Willkommen auf der Online-Dokumentationsseite des Redaktions-Systems !**

Hier finden Sie alle Informationen zur aktuellen Version des Redaktions-Systems (nachfolgend R-S genannt).

### **Erste Schritte**

- Neuerungen im R-S*
- Das R-S*
- Welcher Benutzertyp bin ich?*

### **Benutzertypen Dokumentation**

- Endbenutzer*
- R-S Administrator*
- System Administrator*

### **Archiv- und Dokumentationsbereich**

- Online-Archiv des R-S*
- Ergänzende Online Dokumentation*
- Download der Dokumentation*

### **Erweiterte Informationen**

- Verwendete Symbole*
- Service und Support*
- Sitemap*



Beachten Sie, dass Sie zur Darstellung der Online-Dokumentation eine Mindestauflösung von 1280 x 1024, idealerweise eine Auflösung von 1920 x 1200 benötigen. Die Werte können je nach Format des Bildschirms variieren.



### **Weitere Online-Dokumentationen**

Sie suchen Hilfe für ein anderes Produkt? [Hier](#) gelangen Sie zur Übersicht aller Online-Dokumentationen!

## Rechtliche Hinweise

Der Inhalt dieser Online-Dokumentation ist das geistige Eigentum der profiforms gmbh. Bei der Erstellung der Texte und Abbildungen dieser Online-Dokumentation wurde mit größter Sorgfalt vorgegangen. Trotzdem können Fehler nicht vollständig ausgeschlossen werden. Die profiforms gmbh übernimmt keinerlei Gewähr für die Aktualität, Korrektheit und Vollständigkeit der bereitgestellten Informationen.

Die profiforms gmbh behält sich das Recht vor, den Inhalt dieser Online-Dokumentation ohne vorherige Ankündigung zu verändern oder ergänzen und übernimmt keine Haftung für Fehler in dieser Online-Dokumentation oder daraus resultierende mögliche Schäden.

Für Verbesserungsvorschläge und Hinweise auf Fehler sind Herausgeber und Autoren dankbar.

Diese Software wird gebündelt mit den Schriftarten des DejaVu-Projekts ausgeliefert. DejaVu ist eine Sammlung von verschiedenen, unter freier Lizenz stehenden Schriftarten, die ihren Ursprung in der Schriftartensammlung Bitstream Vera haben. Die Schriftarten stehen unter dem Bitstream Vera Fonts Copyright und dem Arve Fonts Copyright, welche als Lizenzverträge mit installiert werden. DejaVu ist kein preislicher Bestandteil dieses Produkts. Profiforms übernimmt für diese Schriften weder Garantie noch Wartung. Alle Rechte bezüglich dieser Schriften liegen bei Bitstream und dem DejaVu-Projekt.

SAP, SAP R/3, SAPScript, SmartForms, BC-RDI, BC-XFP und andere sind eingetragene Warenzeichen der SAP AG, Walldorf.

Java ist eine eingetragene Marke der Oracle Corporation.

Adobe, Adobe Present, Adobe Central, Adobe Designer, PostScript, PDF, XDP und weitere Warenzeichen sind eingetragene Warenzeichen der Adobe Systems Incorporated.

Hewlett Packard, HP-PCL sind eingetragene Warenzeichen der Hewlett-Packard Company.

Unix ist ein Warenzeichen der Open Group.

Windows ist ein eingetragenes Warenzeichen der Microsoft Corporation.

TBarcode ist ein eingetragenes Warenzeichen der TEC-IT Datenverarbeitung GmbH.

Alle anderen Firmennamen und Produktbezeichnungen sind Warenzeichen oder eingetragene Warenzeichen der jeweiligen Firmen und unterliegen im Allgemeinen warenzeichen-, marken- und/oder patentrechtlichem Schutz.

## Was ist neu in Version 2.5

In der Version 2.5 wurden im Vergleich zur Version 2.4 folgende wesentliche Änderungen/Erweiterungen am R-S vorgenommen:

- Änderungen im R-S Admin-Client:
  - Umstellung von Adobe Flash Flex zu HTML5
  
- Änderungen im R-S Client:
  - Umstellung von Adobe Flash Flex zu HTML5
  - Überarbeitung des Look and Feel
  - Navigationsleiste für Übersichten
  - Öffnen der Objekte als einzelne Tabs
  - In Place Ressourcen-Editor
  - herunterladen von Entwicklungs-Paketen (Schema)
  - Performanceverbesserungen in Übersichten
  - Dashboard
  
- Änderungen im R-S Server:
  - Produktionspakete, welche keine aktive [Test]-Produktionspaketversion haben, verwenden nun automatisch, die aktive [Prod]-Produktionspaketversion

## Einführung

Die heutigen Business-Anforderungen bei der Erzeugung, Gestaltung und bei der Zustellungsform von Dokumenten haben sich in den letzten Jahren stark gewandelt. Wurden zum Beispiel Rechnungs-Dokumente noch Anfang des jetzigen Jahrtausends von vielen Unternehmen als Notwendigkeit angesehen den gesetzlichen Vorgaben zu entsprechen, ergibt sich heute ein ganz anderes Bild. Moderne und innovative Unternehmen sehen Business-Dokumente als Möglichkeit der exklusiven Kunden-Kommunikation und als Marketing-Instrument. Denn Business-Dokumente werden von Kunden nachweislich ganz anders und intensiver wahrgenommen als zum Beispiel Info-Massen-Briefe und Werbe-Mails. Um eine große Kunden-Aufmerksamkeit und Akzeptanz zu erreichen, werden Transaktions-Dokumente wie Rechnungen immer stärker im (Voll-)Farbdruck mit direkt auf den Kunden abgestimmten Werbebotschaften (WhiteSpaces) produziert. Diesem Trend kommen die in den Druckzentren verwendeten Druck- und Kuvertiertechniken entgegen. So lassen sich heute Transaktions-Dokumente im Vollfarb-Druck fast so kostengünstig und in solchen Stückzahlen produzieren, wie mit herkömmlicher schwarz/weiß - Laserdrucktechnik. TRANSPROMO ist ein Synonym für das organische Zusammenwachsen von TRANSaktions-Dokumenten angereichert um PROMotion/werblichen Inhalt.



Die Produktion von TRANSPROMO-Dokumenten wird jedoch in vielen Unternehmen kontrovers diskutiert und kommt deshalb oftmals nur schleppend und mit mäßigem Erfolg zum Einsatz. Dies liegt unter anderem daran, dass in den Unternehmen die Prozesse zur Dokument-Generierung aus der Historie typische IT-Prozesse sind, welche nach Richtlinien der IT und weniger nach Anforderungen aus Vertrieb und Marketing ausgerichtet sind.

### Gemeinsam den Output gestalten ?

profiforms

**Marketing**  
Frau Meier

**Vertrieb**  
Herr Völter

**EDV**  
Herr Ebert

⚠ Hinweis: Personen, Namen und Dokumente sind frei erfunden.

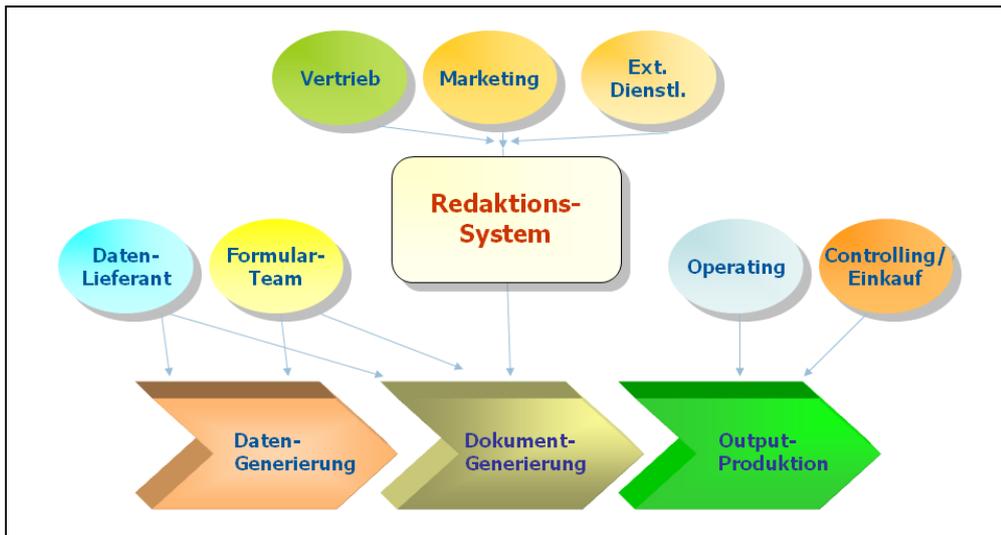
Die Dokument-Generierung im Transaktions-Druckbereich galt bisher als Domäne von technisch versierten Spezialisten, welche sich aus dem Bereich der Datenbesorgung aus ERP/CRM-Systemen, dem Formular-Design, dem Output/Archiv-Processing und vielen weiteren Unternehmensbereichen zusammensetzt. Ein über Jahrzehnte geübtes und filigranes Zusammenspiel ... welches nur bedingt auf TRANSPROMO-Anforderungen flexibel reagieren kann.



⚠ Hinweis: Prozess nur beispielhaft dargestellt.

Und hier kommt das **Redaktions-System (R-S)** zum Einsatz.

Das R-S ist eine moderne, WEB-basierte Business-Applikation zur Anreicherung von Dokumenten mit Zielgruppen orientierten und personalisierten Daten aus dem Vertrieb und dem Marketing. Der Schwerpunkt des R-S liegt dabei auf der Entkopplung des technischen Prozesses der Dokument-Erstellung aus Rohdaten (Transaktions-Formular) und der Aufbringung von standardisierten Elementen des Corporate Designs oder individuellen Informationen wie personalisierter Werbung für den Dokument-Empfänger. D.h. mit der Einführung des R-S erhalten Vertrieb und Marketing die Möglichkeit, Business-Dokumente zu gestalten, ohne aktiv in den IT-Prozess der Generierung der Transaktions-Dokumente einzugreifen zu müssen.



#### Über das R-S können Marketing und Vertrieb u.a. folgende Business-Funktionen ausführen:

- Definition von grafischen oder textuellen Inhalten für vorgegebene Bereiche auf dem Transaktions-Dokument
- Definition einer Business-Logik für dynamische Inhalte auf der Basis von Excel-ähnlichen Entscheidungstabellen
- Festlegung von physikalischen Beilegern (z.B. Flyer aus dem Offset-Druck) pro Dokument/Kunde
- Definition von logischen Beilegern, welche im Druckprozess mit erzeugt werden und pro Dokument/Kunde inhaltlich variieren können (z.B. Gutscheine, AGB's)
- realitätsnahe Druckvorschau der Business-Dokumente inkl. der Elemente, welche durch das R-S eingesteuert werden
- Versionierung der Änderungen im R-S und Verwaltung der Zeitpunkte, wann welcher Inhalt in der Dokument-Produktion zum Einsatz kommt
- u.v.a.m

Weitergehende Informationen zu der Applikations-Umgebung des R-S und den beteiligten Prozessen finden Sie hier: **Umgebung des R-S** Sind Sie End-Nutzer des R-S, so ist die **Nutzer-Dokumentation** der richtige Einstiegspunkt, wenn Sie Ihre Kenntnisse um das R-S vertiefen wollen. Sind Sie mit der System-Administration oder dem Support für das R-S oder in anderer Hinsicht mit technischen Aspekten zum R-S betraut, so nutzen Sie bitte die **Administrator-Dokumentation** für weiterführende Informationen.

## Umgebung des R-S

In diesem Teil der Dokumentation werden Sie mit der Umgebung, in welcher das Redaktions-System (*in der Dokumentation "R-S"*) arbeitet, vertraut gemacht. Dabei lernen Sie weiterführende Begriffe kennen, mit welchen Sie während der Arbeit mit dem R-S in Berührung kommen können. Dies betrifft Personen, welche bestimmte Rollen im Prozess der Dokument-Generierung übernehmen sowie auch IT-Systeme, welche Daten/Konfigurationen an das R-S liefern oder aus dem R-S auslesen, um das Ergebnis der Arbeit im R-S weiterzuverarbeiten.

Zunächst wird vorgestellt, wie mit dem R-S aus Rohdaten produktionsbereite, individualisierte Dokumente generiert werden. Danach wird die Produktions-Umgebung des R-S detaillierter beschrieben und Sie sehen, in welcher Beziehung das R-S zu dem Formular-Design (Rohdaten), den Datensystemen, aus denen das R-S die dokumentspezifischen Daten erhält (z.B. Adressat, Kundename, Kundennummer etc.) und den dokumentgenerierenden Systemen (z.B. ReportWriter) steht. In einem dritten Abschnitt wird dann auf technischer Ebene gezeigt, wie das R-S in die Produktions- und Druckumgebung eingebettet wird und wie andere Komponenten des Produktionsprozesses mit dem R-S interagieren (wie Dokumente aus dem R-S in die Produktion eingeschleust werden).

### Funktionsweise des R-S

Im Redaktions-System werden Rohdokumente flexibel gestaltet. Dazu erhält das Redaktions-System ein sogenanntes Import-Paket, das alle diese Rohdokumente und weitere, für den Dokument-Erstellungsprozess im R-S relevante Daten, enthält. Wie dieses Import-Paket aufgebaut ist, können Sie [hier](#) einsehen.

Die Rohdokumente im Import-Paket enthalten Layout-Bereiche, also Bereiche auf den Dokumenten, die im R-S mit Bildern, Texten, Werten etc. statisch oder durch logische Abhängigkeiten aufgefüllt werden können. Inhalt kann im R-S nur an den dafür vorgesehenen Layout-Bereichen platziert werden. Die Layout-Bereiche können im R-S nicht verändert werden. Das R-S unterstützt allerdings WhiteSpaces, die flexibel freie Bereiche ("Weiße Bereiche") mit Inhalt (z.B. Werbung) füllen können. Außerdem erlaubt das R-S die Erstellung von Beilegern, die den Dokumenten beim Produktions-Prozess beigelegt oder direkt mit den Dokumenten erstellt werden. Hier sehen Sie, wie aus den Rohdaten des Import-Pakets durch das R-S für die weitere Produktion bereite Dokumentdaten generiert werden:

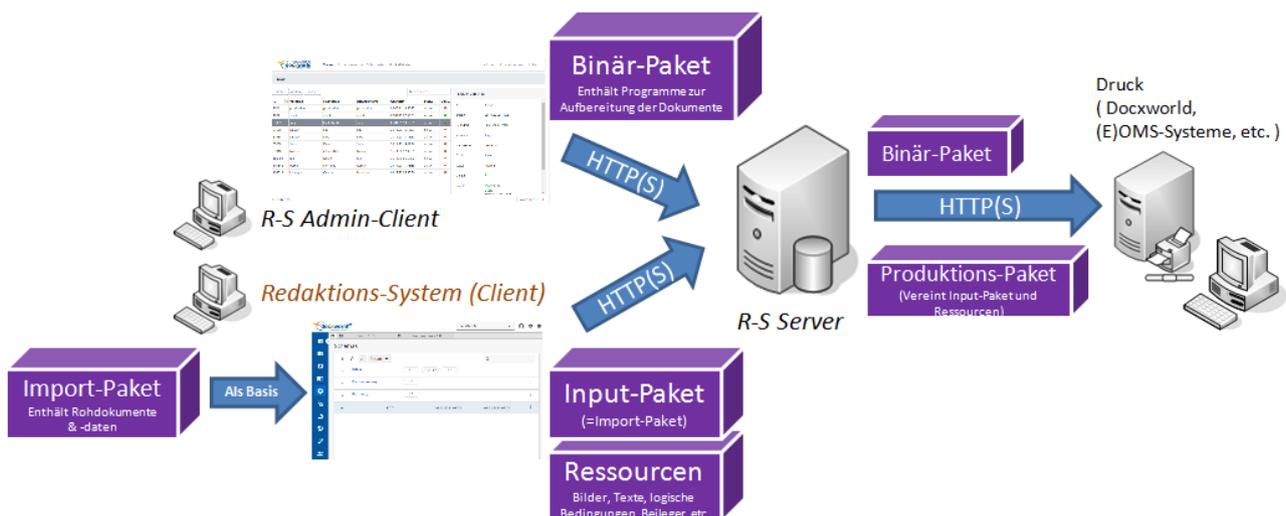


Abbildung A: Funktionsweise des R-S

Als Endnutzer arbeiten Sie mit dem Redaktions-System Client (in Abb. A orange markiert). Als Basis der Arbeit im R-S Client dienen die Rohdokumente und Daten aus dem Import-Paket. Import-Pakete werden durch Formular-Designer erstellt, zum Beispiel mit Hilfe des DeveloperClient. Die Rohdaten müssen also in Form eines Import-Paketes in den R-S Client importiert werden. Im R-S wird das Import-Paket durch das Basisobjekt Transaktions-Formular repräsentiert. Mehr dazu lernen Sie im Verlauf der Dokumentation. Nachdem Sie Ihre Arbeit im R-S Client beendet haben und die Dokumente fertig gestaltet sind, übergibt der R-S Client Folgendes an den R-S Server, welcher daraus ein für die Produktion bereites Paket (im R-S als Basisobjekt Produktions-Paket repräsentiert) generiert. Aus diesem Paket kann dann z.B. der ReportWriter druckfertige Dokumente erstellen. Das Produktions-Paket enthält folgende Komponenten:

- Das **Input-Paket** (= Daten des Transaktions-Formular)

Es enthält die Daten des Import-Paketes und damit die Rohdokumente: Das Grundlayout der Dokumente, Konfigurationen, Daten, Ressourcen und weitere Informationen. Die Rolle von Input-Paketen wird im Folgenden noch genauer erklärt.

- Die **Ressourcen** (= Daten des Schema)

Das Basisobjekt Schema enthält alle Daten (Bilder, Texte, Beileger, etc.), mit denen die Rohdokumente aus dem Input-Paket angereichert (gefüllt) werden (= Es enthält die komplette Gestaltung der Rohdokumente durch das R-S, also alles, was Sie im R-S erstellt haben.

Daraus wird durch den R-S Server das **Produktions-Paket** generiert, welches die beiden Komponenten enthält. Aus dem Produktions-Paket können die fertigen Dokumente generiert werden. Dies geschieht durch Programme wie z.B. den ReportWriter, die die Gestaltung der Rohdokumente auf diese anwenden. Diese Programme, Skripts und Daten, die dazu nötig sind, um aus dem Produktions-Paket fertige Dokumente zu erstellen und die Daten aus den Datensystemen (ERP-/CRM-/HOST-Systemen, wird im Folgenden noch erklärt) in die Dokumente einzufügen, sind im **Binär-Paket** enthalten (Binär-Pakete werden nicht durch den R-S Client, sondern mit Hilfe des R-S Admin-Client auf den R-S Server geladen). Der R-S Server gibt das Binär-Paket dann unverändert zusammen mit dem Produktions-Paket in die Druckumgebung weiter (*dazu später mehr*). Welches Binär-Paket ein Produktions-Paket benötigt (also welche Programme darin enthalten sein müssen und in welcher Version), ist im Import-Paket festgelegt. Fehlt das **benötigte** Binär-Paket, kann das Produktions-Paket nicht für den Druck aufbereitet werden.

## **Die R-S Umgebung**

Im letzten Abschnitt wurde gezeigt, wie mit dem R-S durch Rohdaten aus Import-Paketen und dem Binär-Paket fertige Dokumente erstellt werden. Hier wird nun genauer beschrieben, welche Rolle das R-S im Produktions-Prozess übernimmt. Vorab eine kurze Beschreibung der verwendeten Begriffe:

- **ERP-/CRM-/HOST-System (z.B. SAP-Systeme)**

Sie enthalten und verarbeiten die für Dokumente bestimmten Daten und stellen diese in Datenfeldern für den Druckprozess zur Verfügung. Im R-S können Daten aus diesen Systemen als **Datenstrom-Variablen** genutzt werden. Welche Datenstrom-Variablen aus dem ERP-/CRM-/HOST-System im R-S angesprochen und benutzt werden dürfen, wird durch den Formular-Designer im DeveloperClient festgelegt (Das Binär-Paket enthält deklariert also alle verfügbaren Datenstrom-Variablen). Im R-S können Sie dann mit den verfügbaren Datenfeldern arbeiten. Die Werte der Datenfelder werden während des Druckprozesses für jedes Dokument spezifisch aufgefüllt, da die Werte der Datenfelder von Dokument zu Dokument unterschiedlich sein können. Solche Daten können z.B. Kundename, Kundennummer, Rechnungsbetrag, Anschrift etc. sein. Mehr über Datenstrom-Variablen erfahren Sie in deren **Definition**.

- **DeveloperClient**

Im EDeveloperClient werden die Rohdokumente erstellt. Neben dem Layout der Dokumente werden hier auch die Bereiche festgelegt, in die Sie im R-S Inhalt einfügen können (im R-S *Erweiterungs-Punkte*), welche Daten aus dem ERP-/CRM-/HOST-System Sie im R-S verwenden können, welche Fonts im R-S benutzt werden dürfen und weitere Eigenschaften des Formulars, z.B. ob und welche Beileger erlaubt sind. Die durch den DeveloperClient generierten Daten werden im Import-Paket zusammengefasst.

- **ReportWriter (Binär-Paket)**

Der ReportWriter fügt Dokumente und Rohdaten zu fertigen PDF's zusammen, um sie für den Druckprozess aufzubereiten. Er fügt also das Resultat aus dem R-S mit den Rohdokumenten des Import-Pakets sowie den benötigten, in den Dokumenten verwendeten Daten aus dem Datensystem zusammen (Er füllt also die Werte der Datenstrom-Variablen). Die durch den ReportWriter erstellten PDF's können dann in den Druck geschickt werden. Der ReportWriter und alle nötigen Konfigurationen und weiteren Daten müssen im Binär-Paket enthalten sein. Alternativ ist auch die Verwendung anderer Software möglich.

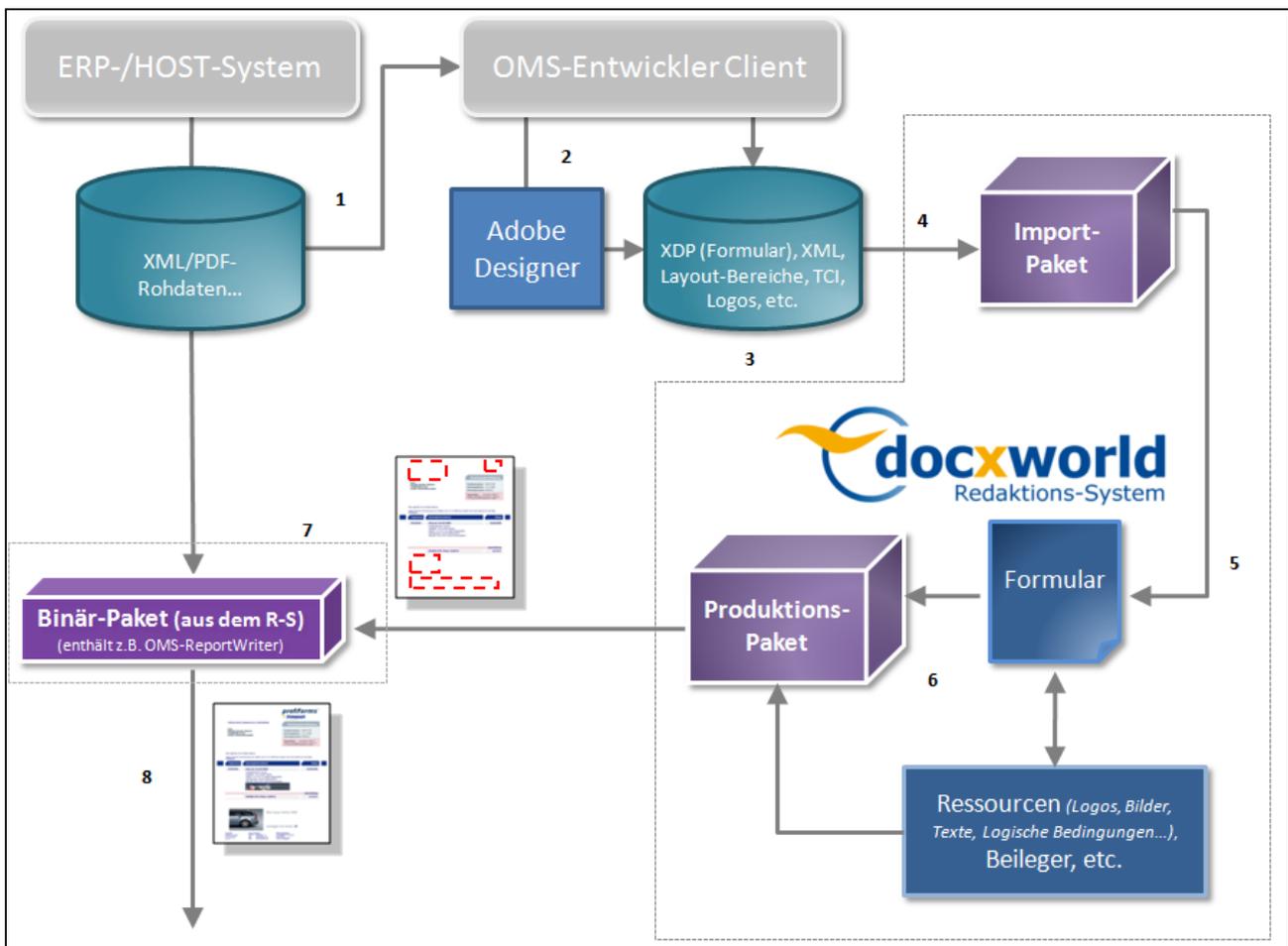


Abbildung B

Komponente des R-S / aus dem R-S (Wurde oben bereits thematisiert).

In Abb. B sehen Sie, wie das R-S in den Dokument-Erstellungsprozess eingebettet ist, von den Rohdaten des ERP-/CRM-/HOST-Systems bis zum fertigen Dokument.

**Der typische Ablauf des in Abb. B beschriebenen, kompletten Dokument-Erstellungs-Prozesses:**

Nummer	Beschreibung
1	Das ERP-/CRM- oder HOST-System migriert seine Datenstruktur /XML-Rohdaten) in den DeveloperClient. Dort kann der Formular-Entwickler bei der Formularentwicklung darauf zugreifen, sieht, welche Datenfelder im System zur Verfügung stehen und kann steuern, welche davon wie im R-S verwendet werden dürfen.
2	Mit dem in den DeveloperClient integrierten Adobe Designer wird das Layout des Formulars erstellt (XDP).

Nummer	Beschreibung
3	Neben dem Layout der Dokumente (2) legt der Formular-Designer noch andere Eigenschaften fest, z.B. die Layout-Bereiche im Formular, die durch das R-S gefüllt werden dürfen, welche Daten aus dem Datenstrom des ERP-/CRM-/HOST-Systems im R-S verwendet werden dürfen (später im R-S die Datenstrom-Variablen) und weitere Konfigurationen.
4	Aus den Dateien, Formularen und Konfigurationen wird ein Import-Paket für das R-S erstellt, das alle relevanten Daten beinhaltet.
5	Das Import-Paket bildet die Basis für das Transaktions-Formular des R-S, das dann im R-S gestaltet wird. Die festgelegten Layout-Bereiche werden durch Ressourcen gefüllt, Beileger werden definiert, etc. Die vom Formular-Designer deklarierten Datenstrom-Variablen werden dazu genutzt, um im R-S Daten aus dem ERP-/CRM-/HOST-System zu verwenden. Ihre Werte werden später durch den ReportWriter (7) in das Dokument eingefügt.
6	Nachdem die Bearbeitung der Rohdaten im R-S abgeschlossen ist wird ein Produktions-Paket erstellt, das die gestalteten Dokumente mit den durch das R-S zugewiesenen Ressource enthält und aus dem druckfertige Dokumente generiert werden können.
7	Durch den ReportWriter aus dem Binär-Paket werden aus dem Produktions-Paket und den Daten aus dem Datenstrom des ERP-/CRM-/HOST-Systems die fertigen Dokumente erstellt: Die Daten aus dem Datenstrom, die im R-S für die Dokumentgestaltung benutzt wurden (Datenstrom-Variablen), werden durch den ReportWriter eingefügt und die Informationen aus dem Produktions-Paket werden zu fertigen Dokumenten verarbeitet.
8	Das fertige Dokument wird in die Produktion geschickt (z.B. zum Spooler, siehe <a href="#">unten</a> )

**Wie aus den R-S Daten in der Praxis Dokumente generiert werden**

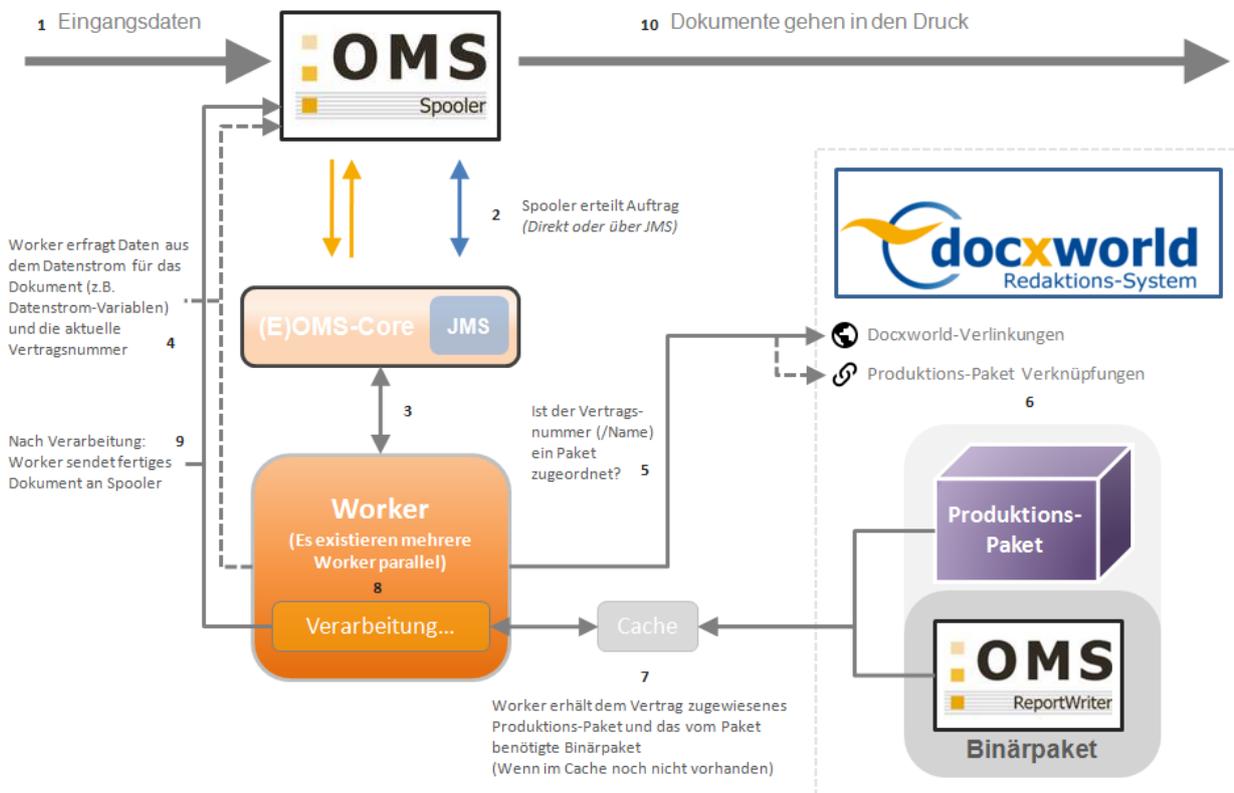


Abbildung C

Bis jetzt haben Sie gesehen, wie durch das R-S aus den Rohdaten des Formular-Designers druckfertige Dokumente erstellt werden. Hier sehen Sie nun vertieft, wie in der Realität durch Nutzung des **Spoolers**, des **ReportWriters** und von **docxworld** die Dokumente aus dem R-S in den Produktionsprozess (und Druckprozess) eingeführt werden und welche technischen Abläufe nötig sind. Abb. C erweitert die Prozesse aus Abb. A + B um die technische Anbindung an den Druckprozess. In Abb. C ist das Schema bei Verwendung des (E)OMS-Core-Systems dargestellt. Das (E)OMS-Core-System ermöglicht es, Aufträge des Spoolers auf mehrere Recheneinheiten (sog. Worker) aufzuteilen ("verteilte Systeme") und beschleunigt so den Generierungsprozess. Momentan unterstützt nur das EOMS-Core-System eine Anbindung an das Redaktions-System.

**Die einzelnen Ablaufschritte in Abb. C sind in chronologischer Reihenfolge nummeriert:**

Nummer	Beschreibung
1	Der Spooler empfängt die für das Dokument bestimmten Daten (Die Daten aus dem ERP-/CRM-/HOST-System aus Abb. B ) auf einer seiner Schnittstellen (LPD, SAP, O2O, OMS.REMOTE oder aus seinem Scan-Verzeichnis).

Nummer	Beschreibung
2	Der Spooler erteilt dem (E)OMS-Core-System den Verarbeitungs-Auftrag für die Dokumente. Dies geschieht standardmäßig über JMS Messaging, Spooler und der (E)OMS-Core können aber auch direkt kommunizieren.
3	Das (E)OMS-Core-System verteilt die Aufträge auf mehrere Worker (Maschinen). In Abb. C ist nur 1 Worker dargestellt, da jeder Worker gleich arbeitet. Jeder Worker verarbeitet dann die ihm zugewiesenen Dokument-Erstellungs-Aufträge.
4	Der Worker erfragt die zum Formular gehörenden Daten aus dem Datenstrom (inkl. der Docxworld-Vertragsnummer des Auftrags bzw. dem Namen des Dokuments) vom Spooler. Die Docxworld-Vertragsnummer / der Name ist nötig, um im R-S gestaltete Dokumente eindeutig einem Datenstrom und einer Produktion zuweisen zu können.
5	Der Worker erfragt vom R-S, ob der Vertragsnummer / dem Namen ein Produktions-Paket im R-S zugeordnet ist. Zuweisungen im R-S werden durch Docxworld-Links und Produktions-Paket Verknüpfungen realisiert, die Sie im Laufe der Dokumentation kennen lernen werden. Produktions-Pakete haben Sie bereits oben kennengelernt.
6	Existiert im R-S ein zugeordnetes Produktions-Paket, übergibt das R-S das Paket und das für das Paket benötigte Binär-Paket an den Worker. Binär-Pakete, die die für die Dokument-Generierung aus Rohdaten notwendigen Programme enthält, haben Sie bereits oben kennengelernt. Andernfalls wird der Prozess für dieses Dokument abgebrochen.
7	Sind Produktions-Paket oder Binär-Paket schon im Cache des Workers vorhanden, wird das Produktions-Paket bzw. das Binär-Paket nicht aus dem R-S geladen, sondern es werden die Pakete aus dem Cache benutzt, um eine schnellere Ausführung zu erzielen. Ansonsten werden Produktions-Paket / Binär-Paket erst in den Cache geladen.
8	<p>Die Programme aus dem Binärpaket (in der Regel ReportWriter) verarbeiten die Daten aus dem Datenstrom des Spoolers, die der Worker in #4 erhalten hat und das Produktions-Paket zu fertigen Dokumenten (Einfügen der Daten aus dem Datenstrom in das Dokument des Produktions-Pakets). Dieser Prozess geschieht im Cache des Workers.</p> <p>Vorsicht: Wenn das Produktions-Paket einem Datenstrom (Vertrag) zugeordnet ist, der nicht für dieses Dokument bestimmt ist, kommt es zu unerwünschtem Verhalten, da der Datenstrom nicht die erwarteten Daten enthält. Achten Sie deshalb darauf, in Docxworld-Links immer die richtige Vertragsnummer anzugeben, die auch zum gewünschten Datenstrom gehört.</p>
9	Nachdem das Dokument fertig generiert wurde, sendet der Worker das fertige Dokument direkt an den Spooler zurück.
10	Der Spooler erfragt in regelmäßigen Abständen vom (E)OMS-Core-System, ob die Dokument-Generierung abgeschlossen ist. Wenn sie das ist, schickt der Spooler die empfangenen Dokumente in den Druck (z.B. docxworld).

## Die Personen, die an den Prozessen beteiligt sind und ihr Aufgabenfeld:

- **ERP-Entwickler**

Ist zuständig für die Verwaltung des ERP-/CRM-/HOST-Systems und die Aufbereitung der durch das System bereitgestellten Daten. Außerdem ist er bei der Formularerstellung mit dem Formular-Team beteiligt, um eine effiziente Nutzung der Daten aus dem System im Formular zu gewährleisten.

- **Formular-Team**

Das Formular-Team arbeitet mit dem Formular-Designer DeveloperClient und Adobe Designer), entwirft und konzipiert damit das Rohformular und die Daten (Ressourcen, Logo, Hintergründe, andere Dateien), die später im R-S gestaltet und genutzt werden. Außerdem konfigurieren sie die Eigenschaften des Formulars und bereiten die Datenstruktur aus dem ERP-/CRM-/HOST-System für die Verwendung im R-S auf.

- **Vertrieb, Marketing, etc.**

Vertrieb und Marketing sind die Hauptnutzer des R-S. Sie gestalten die durch das Formular-Team vorgegebenen Rohdokumente und füllen sie mit Inhalt, entwerfen Beileger, etc. QA-Teams überprüfen die Qualität der Enddokumente.

- **Druck-Operating und Produktion**

Sie sind für den weiteren Produktions- Prozess verantwortlich, kümmern sich um Druck, Kuvvertierung und Versand der Dokumente. Außerdem sind sie für die Produktion über docxworld oder Produkte zuständig.

Eine Sammlung aller Begriffe, die Sie für die Verwendung des Redaktions-Systems kennen sollten finden Sie im [Glossar](#).

## Die Benutzertypen

Im R-S werden drei verschiedene Benutzertypen mit unterschiedlichen Aufgaben unterschieden:

- Endbenutzer
- R-S-Administrator
- System-Administrator

### Welcher Benutzertyp sind Sie?

#### **Endbenutzer**

Der Endbenutzer hat die Aufgabe die Software via WEB-Browser zu bedienen und Transaktions-Formulare, Schemas, Produktions-Pakete, Materialien, Beileger etc. zu verwalten. Einstellungen und Veränderungen in den Systemeinstellungen des R-S System sind Ihm nicht zugänglich. Ein Endbenutzer hat eingeschränkte Zugriffsrechte.

[Zur Dokumentationsseite](#)

#### **R-S-Administrator**

Der R-S-Administrator arbeitet an den Einstellungen des R-S wie zum Beispiel den Nutzer-Rechten für Endbenutzer oder der Kopplung des R-S mit externen Systemen. R-S-Administratoren stellen das Programm auf die Umgebung ein und optimieren seine Abläufe. Ein R-S-Administrator besitzt Zugriff auf nahezu alle Funktionen der Software.

[Zur Dokumentationsseite](#)

#### **System-Administrator**

Der System-Administrator ist der Administrator des Systems, auf dem das R-S installiert wird. Er kümmert sich ausschließlich um die Systemeinstellungen, wie zum Beispiel Auswahl und Freigabe der Netzwerk-Ports sowie der Installation und Pflege (Backup). Ein System-Administrator besitzt die Kontrolle über alle Systemeinstellungen der R-S Software.

[Zur Dokumentationsseite](#)

## Nutzer-Dokumentation

In der Nutzer-Dokumentation lernen Sie den Aufbau des R-S kennen. Außerdem werden Sie hier angeleitet, wie Sie mit den einzelnen R-S Objekten arbeiten können. Als Administrator beachten Sie bitte die **Administrator-Dokumentation**.

Nachfolgend sehen Sie eine Übersicht über die in der Nutzer-Dokumentation behandelten Themen. Klicken Sie auf eine Überschrift, um zu dem jeweiligen Kapitel zu gelangen. Beachten Sie, dass in dieser Übersicht nur die wichtigsten Kapitel aufgeführt sind. Um Schritt für Schritt durch alle Seiten der Dokumentation zu navigieren, können Sie die Links am Ende jeder Seite benutzen, die Sie zur nächsten oder vorherigen Seite der Dokumentation führen.

 Die Nutzer-Dokumentation beginnt mit den **Grundlagen** des R-S. Zum Einstieg empfehlen wir unbedingt, damit zu beginnen. Eine Einführung in die technische Arbeitsweise und die Umgebung des R-S finden Sie **hier**.

### 1. Grundlagen

- **Grundlagen über den R-S Workflow**
  - Beschreibt den Status-Workflow der Objekte und die Versionierung im R-S.
- **Was sind Arbeitsbereiche?**
  - Arbeitsbereiche bilden unabhängige, abgetrennte Arbeitsumgebungen.
- **Die Ressourcen-Typen im R-S**
  - Erklärt die verschiedenen Ressourcen, die im R-S zur Speicherung von Daten, Inhalten und Informationen dienen.
- **Die Basisobjekte im R-S**
  - Erklärt die verschiedenen Basisobjekte im R-S, die Arbeit mit Dokumenten im R-S ermöglichen.
- **Erweiterungen**
  - Mit Erweiterungen werden Ressourcen in Dokumente eingebunden und dadurch mit Inhalt gefüllt.

### 2. Erste Schritte (Bedienung und Navigation)

- **Die Top-Navigation**
  - Bildet den Teil der Nutzeroberfläche des R-S, der grundlegende Konfigurationsmöglichkeiten enthält.

- **Der Detail-Bereich**
  - Enthält Basis-Objekte und ermöglicht die Arbeit mit ihnen. Dies ist der Bereich im Layout des R-S, in dem Sie arbeiten.
- **Die Hilfe-Systeme des R-S**
  - Beschreibt, welche integrierten Hilfe-System das R-S zur Verfügung stellt.

### **3. Dashboard**

- **Aufbau des Dashboards**
  - Zeigt einen Überblick der letzten Aktivitäten, welche im R-S durchgeführt wurden.

### **4. Arbeiten mit Transaktions-Formularen**

- **Was sind Transaktions-Formulare?**
  - Sie beinhalten die Rohdokumente und Konfigurationen des Formulars, das im R-S gestaltet und bearbeitet werden soll.
- **Arbeiten mit Transaktions-Formularen**
  - Zeigt, wie Transaktions-Formulare in der Praxis verwaltet, bearbeitet und angewandt werden.
- **Arbeiten mit Transaktions-Formular-Versionen**
  - Im R-S arbeiten Sie immer mit Versionen eines Basisobjekts: Hier werden Formular-Versionen thematisiert.
- **Eigenschaften einer Transaktions-Formular-Version**
  - Zeigt, welche Inhalte und Eigenschaften eine Formular-Version besitzt und wie Sie sie bearbeiten können.

### **5. Schemas**

- **Was sind Schemas?**
  - Sie ermöglichen die Gestaltung der Rohdokumente des Formulars durch Ressourcen und die Zusteuerung von Beilegern / WhiteSpaces.
- **Arbeiten mit Schemas**
  - Zeigt, wie Schemas in der Praxis verwaltet, bearbeitet und angewandt werden.
- **Arbeiten mit Schema-Versionen**
  - Im R-S arbeiten Sie immer mit Versionen eines Basisobjekts: Hier werden Schema-Versionen thematisiert.
- **Eigenschaften einer Schema-Version**
  - Zeigt, welche Inhalte und Eigenschaften eine Schema-Version besitzt und wie Sie sie bearbeiten können.

## 6. Produktions-Pakete

- **Was sind Produktions-Pakete?**
  - Sie bündeln Formular und Schema in einem, für den weiteren externen Produktionsprozess geeigneten, Paket (Endresultat).
- **Mit Produktions-Paketen arbeiten**
  - Zeigt, wie Produktions-Pakete in der Praxis verwaltet, bearbeitet und für die Produktion bereit gemacht werden.
- **Den Status eines Produktions-Pakets setzen**
  - Zeigt, wie Sie den Status eines Produktions-Pakets setzen (v.a., wie Sie Pakete produktionsbereit machen).

## 7. Material

- **Was sind physikalische Beileger?**
  - Erklärt die Funktionsweise physikalischer Beileger.
- **Was sind logische Beileger?**
  - Erklärt die Funktionsweise logischer Beileger.
- **Materialien**
  - Materialien definieren die physischen Eigenschaften eines Beilegers.
- **Physikalische Beileger erstellen**
  - Zeigt, wie physikalische Beileger erstellt werden.
- **Logische Beileger erstellen**
  - Zeigt, wie logische Beileger erstellt werden.
- **Material für physikalische Beileger erstellen**
  - Zeigt, wie Material für physikalische Beileger erstellt werden.
- **Material für logische Beileger erstellen**
  - Zeigt, wie Material für logische Beileger erstellt werden.

## 8. Ressourcen

- **Ressourcenverwaltung**
  - In Schema- und Bibliotheks-Versionen können Sie Ressourcen erstellen und verwalten.

- **Mit Erweiterungen arbeiten**
  - Zeigt, wie Ressourcen, Beileger und WhiteSpaces per Erweiterung in das Dokument eingefügt / dem Dokument beigelegt werden.

## 9. Bibliotheken

- **Was sind Bibliotheken?**
  - Eine Bibliothek ist eine Sammlung von Ressourcen, die Sie in einer Schema-Version nutzen können.
- **Arbeiten mit Bibliotheken**
  - Zeigt, wie Bibliotheken in der Praxis verwaltet, bearbeitet und angewandt werden.
- **Eine Bibliothek an ein Schema binden**
  - Zeigt, wie Sie eine Bibliothek in eine Schema-Version einbinden, um sie zu nutzen.

## 10. Docxworld-Links

- **Was sind Docxworld-Links?**
  - Sie machen Produktions-Pakete für externe Komponenten verfügbar und somit für den weiteren Produktionsprozess.
- **Arbeiten mit Docxworld-Links**
  - Zeigt, wie Docxworld-Links in der Praxis verwaltet, bearbeitet und angewandt werden.

## 11. Produktions-Paket Verknüpfungen

- **Was sind Produktions-Paket Verknüpfungen?**
  - Sie machen Produktions-Pakete für externe Komponenten verfügbar und somit für den weiteren Produktionsprozess.
- **Arbeiten mit Produktions-Paket Verknüpfungen**
  - Zeigt, wie Produktions-Paket Verknüpfungen in der Praxis verwaltet, bearbeitet und angewandt werden.

## 12. Arbeit mit Objekten

- **Was sind EVAL-Anweisungen?**
  - Sie ermöglichen dynamische Einbindung von Ressourcen und dynamische Berechnungen.
- **Funktionen**
  - Das R-S stellt diverse vorgefertigte Funktionen zur Verfügung, die Sie in EVAL-Anweisungen nutzen können.
- **Inline-Sequenzen**
  - Sie ermöglichen die Textformatierung in einer Variablen.
- **Datentypen**
  - Das R-S unterstützt diverse Datentypen, z.B. Integer für Numerische Werte und Strings für Zeichenketten.

## Grundlagen

Um mit dem R-S erfolgreich arbeiten zu können, sollten Sie eine genaue Kenntnis über die Arbeitsweise des R-S besitzen und wissen, wie der Dokumenterstellung-Prozess im R-S realisiert wird. Deshalb werden in diesem Kapitel zunächst die theoretischen Grundlagen des R-S vermittelt. Wenn Sie sich als Endnutzer auch darüber informieren möchten, wie das R-S mit anderen Komponenten im Produktions-Prozess zusammenarbeitet, erfahren Sie mehr darüber im **Abschnitt über die Umgebung des R-S**.

- Sie erhalten zunächst eine Einführung in den **Workflow des R-S**, also den Arbeitsablauf innerhalb des R-S.

Hier lernen Sie die verschiedenen **Status** innerhalb des Workflows und das Konzept der **Versionierung** kennen.

- Im R-S existieren des Weiteren sogenannte **Arbeitsbereiche**.

Jeder Nutzer hat einen/mehrere ihm zugewiesene Arbeitsbereich(e), in denen er arbeiten kann. Es wird beschrieben was genau ein Arbeitsbereich ist und wozu er dient.

- Im R-S werden Daten und Informationen in **Ressourcen** gespeichert.

Es gibt unterschiedliche Ressourcen für unterschiedliche Arten von Daten sowie Ressourcen, die das Arbeiten mit Datenströmen und Belegern ermöglichen.

- Als nächstes lernen Sie die **Basisobjekte** des R-S kennen:

Sie sind die Hauptelemente des R-S und modellieren den Dokument-Erstellungsprozess.

- Zuletzt werden **Erweiterungen** von Schemas vorgestellt:

Mit ihnen füllen Sie Dokumente mit Ressourcen und fügen Beileger an Dokumente an.

## Status - Workflows und Versionierung

Bevor Sie mit dem R-S arbeiten ist es wichtig, dass Sie die Konzepte hinter dem R-S verstehen. 2 Grundlegende Aspekte der Arbeitsweise des R-S sind dabei die **Versionierung** und der **R-S Workflow**. Es ist für das Arbeiten im R-S essenziell, dass Sie mit diesen Eigenschaften des R-S vertraut sind, da sie ein wichtiger Bestandteil des R-S Arbeitsablaufes sind.

### Versionierung im R-S

Im R-S gibt es Basisobjekte. Dies sind die grundlegenden Elemente, mit denen Sie arbeiten werden. Folgende Basisobjekte sind der Versionierung unterworfen:



(Dies sind gleichzeitig auch die Basisobjekte, die dem R-S Status Workflow unterworfen sind. Dazu mehr im nächsten Abschnitt)

Dies sind die vier elementaren Basisobjekte im R-S, die der Versionierung unterworfen sind:

Basisobjekt	Beschreibung
<b>Formulare</b>	Sie beinhalten die Rohdokumente und stellen zu den Dokumenten gehörende Datenströme (Informationen) zur Verfügung.
<b>Schemas</b>	Schemas ermöglichen die Gestaltung, Anpassung und Individualisierung eines Formulars.
<b>Bibliothek</b>	Bibliotheken beinhalten Ressourcen, die Sie in einem Schema verwenden können (z.B. Bilder).
<b>Produktions-Pakete</b>	Produktions-Pakete bündeln Formular und Schema in einem Paket, das außerhalb des R-S verwendet werden kann.

Mehr über die Objekte lernen Sie in einem späteren Abschnitt. Zunächst konzentrieren wir uns auf die Versionierung: Versionierung bedeutet, dass Sie von einem Basisobjekt, das Sie erstellt haben, mehrere Versionen anlegen können (und mindestens eine anlegen müssen), mit denen Sie dann arbeiten. Die Versionen sind dabei voneinander unabhängig, sie haben nur die Zugehörigkeit zum selben Basisobjekt gemeinsam. Im R-S wird also nicht direkt mit den Basisobjekten Formular, Schema, etc. gearbeitet, sondern mit ihren Versionen.

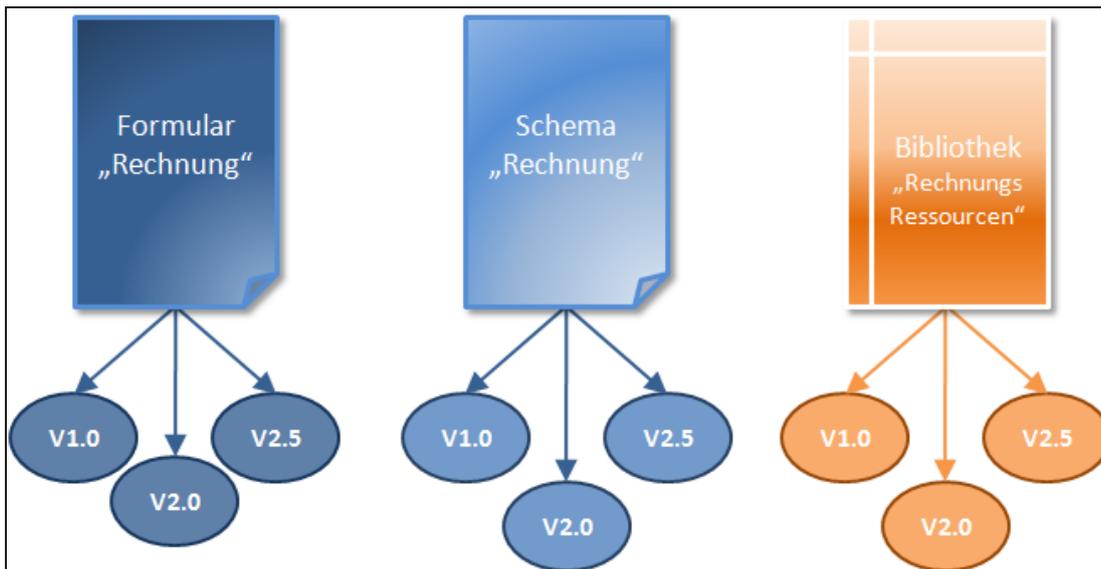


Abbildung B

Alle Eigenschaften und Inhalte, die ein Objekt enthalten kann, werden in den Versionen der Objekte abgelegt. In Abb. B existieren also folgende nutzbare Objekte

Das Formular "Rechnung v1.0"

Das Formular "Rechnung v2.0"

Das Formular "Rechnung v2.5"

Das Schema "Rechnung v1.0"

Das Schema "Rechnung v2.0"

Das Schema "Rechnung v2.5"

Die Bibliothek "Rechnungs-Ressourcen v1.0"

Die Bibliothek "Rechnungs-Ressourcen v2.0"

Die Bibliothek "Rechnungs-Ressourcen v2.5"

Von einem angelegten Basisobjekt muss also mindestens eine Version existieren, damit Sie damit arbeiten können, es können aber auch mehrere Versionen eines Objekts zur gleichen Zeit existieren.

**i** Bei Produktions-Paketen existiert keine Versionierung in diesem Sinne, da Produktions-Pakete die Version der Schema-Version übernehmen: Für die Schema-Version 1.5 des Schemas Rechnung wird ein Produktions-Paket für die Version 1.5 des Schemas Rechnung generiert.

**Die Struktur des R-S mit Versionierung in der Praxis:**

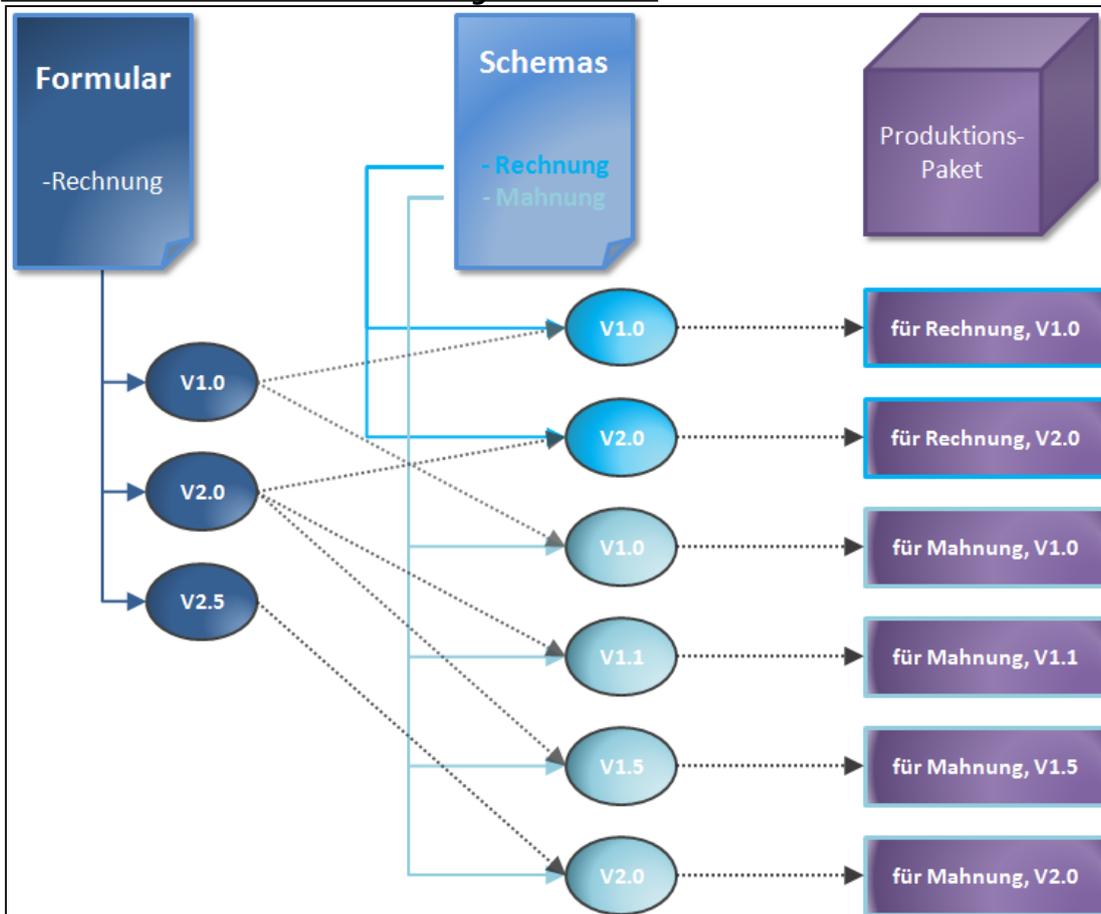


Abbildung C

Wie Sie sehen, verläuft der komplette R-S Prozess auf Basis der Versionen der Formulare und Schemas, während die Objekte Formular Rechnung, Schema Rechnung und Schema Mahnung als "Ordner" für ihre Versionen dienen. (Schemas brauchen immer ein Formular, das durch sie gestaltet wird, weswegen in Abb. C jeder Schema-Version eine Formular-Version zugewiesen wird). Zur Vereinfachung fehlen in der Darstellung die Bibliotheken. Für sie verläuft die Versionierung aber analog.

**Einige Vorteile der Versionierung:**

- + Übersichtliche Verwaltung von Objekten möglich.
- + Versionierung und systematische Aktualisierung von Objekte möglich.
- + Änderungen können in einer neuen Version implementiert werden, die alte Version bleibt erhalten.
- + Alte Versionen bleiben solange in Gebrauch, bis sichergestellt wurde, dass die neue Version fehlerfrei ist.

 Versionen müssen nicht zwingend Versionsnummern als Namen haben. Sie können einer Version jeden beliebigen Namen geben. In den meisten Fällen empfiehlt es sich jedoch aufgrund der besseren Übersichtlichkeit, Versionen zu Nummerieren.

### Status-Workflow im R-S

**Versionen** eines Basisobjekts (Formulare, Schemas, Bibliotheken und Produktions-Pakete) besitzen im R-S einen **Status**. Versionen haben einen Lebens-Zyklus, dieser wird durch ihren Status signalisiert. Der Status gibt also den aktuellen Zustand der Version an. Die meisten Statusänderungen werden durch den Benutzer durchgeführt. Es gibt aber auch temporäre, durch das R-S eingeleitete Status. Der Status eines Objekt bestimmt, welche Aktionen mit diesem Objekt durchgeführt werden dürfen, z.B. ob ein Objekt geändert oder verwendet werden darf.

Im R-S gibt es folgende Status:

Status	Erklärung	Objekte
EDIT	Das Objekt befindet sich in der Erstellungs-Phase und wird noch bearbeitet.	
TEST	Das Objekt befindet sich in der Test-Phase. Die Bearbeitung des Objekts ist abgeschlossen.	 
ACCEPTED	Das Objekt wurde getestet und zur Nutzung freigegeben.	 
REJECTED	Das Objekt wurde zurückgewiesen, z.B. während der Test-Phase und muss überarbeitet werden.	 
DEPRECATED	Das Objekt wurde schon zur Nutzung freigegeben, ist jetzt aber veraltet und sollte nicht mehr benutzt werden.	 
NEW	Das Produktions-Paket wurde neu angelegt (Übergangszustat).	
SCHEDULED	Das Produktions-Paket befindet sich in der Warteschlange zur Generierung (Übergangszustat).	
GENERATING	Das Produktions-Paket wird gerade generiert (Übergangszustat).	
GENERATED	Das Produktions-Paket wurde fertig generiert.	
PRODUCTION	Das Produktions-Paket wurde freigegeben und kann jetzt in der Produktion verwendet werden.	
ERROR	Bei dem Produktions-Paket ist ein Fehler aufgetreten.	

- Status gibt es bei Produktions-Paketen.
- Status gibt es bei Formularen, Schemas und Bibliotheken.

⚠

- Im **REJECTED**-Status kann ein Produktions-Paket zwar bearbeitet, aber nicht wieder verwendet werden.
- Im **EDIT**- und **TEST**-Status können Formulare und Bibliotheken verwendet werden, für die Produktion müssen sie aber **ACCEPTED**-Status haben.

### Workflows der Objekte im R-S

Im R-S werden 3 Abschnitte im Lebens-Zyklus einer Objektversion unterschieden:

- **Design-Zyklus** (Bearbeitung des Objekts)
- **Test-Zyklus** (Bearbeitung abgeschlossen, Objekt wird getestet)
- **Produktions-Zyklus** (Tests abgeschlossen, Objekt ist freigegeben)

Zwischen Design- und Test-Zyklus kann eine Objektversion hin und her wechseln, hat eine Version aber einmal den Produktions-Zyklus erreicht, kann nicht mehr in den Test- oder Design-Zyklus gewechselt werden. Hier sehen Sie den kompletten **Statusworkflow des R-S**:

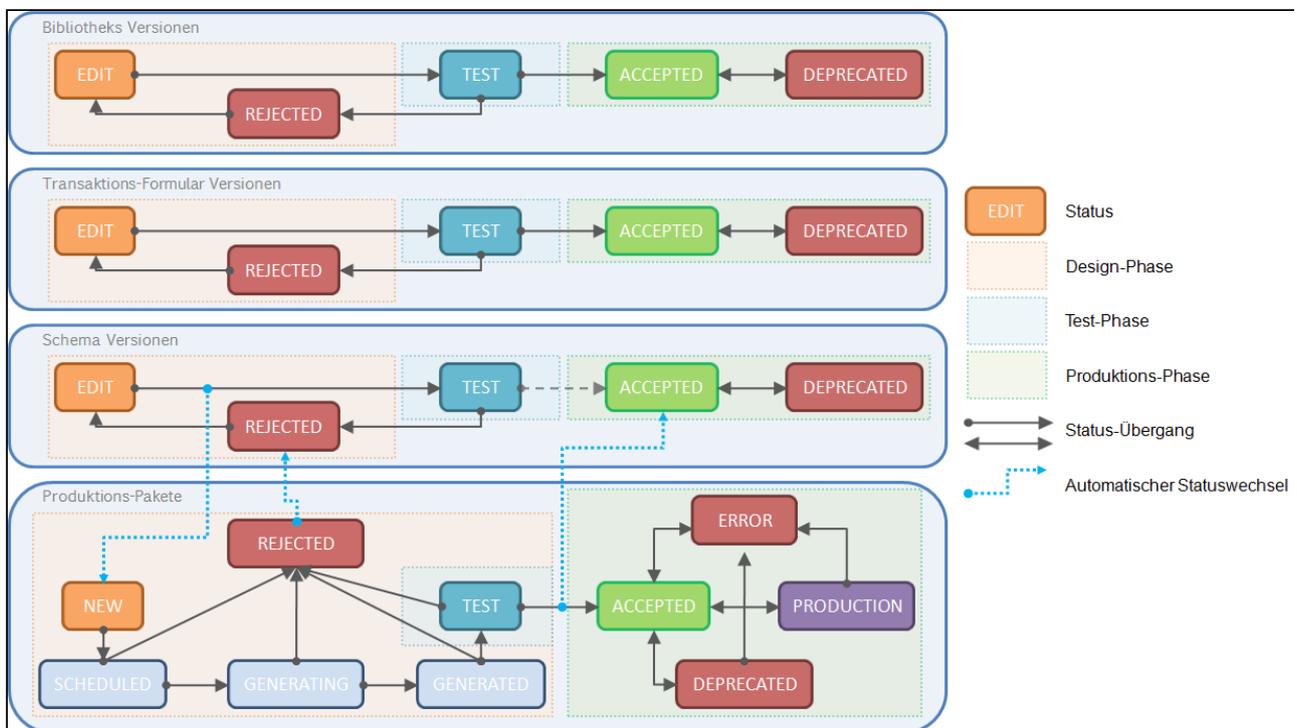


Abbildung C

**Für Transaktions-Formulare, Schemas und Bibliotheken verläuft der Status-Workflow gleich:**

Phase	Beschreibung
Bearbeitungs-Phase	<p>Nach der Erstellung haben Formular-Versionen, Schema-Versionen und Bibliotheks-Versionen den Status <b>EDIT</b>. Nur im <b>EDIT</b>-Status können Sie Objekte bearbeiten und verändern. Wechseln Sie zum <b>TEST</b>-Status, wenn Sie das Objekt fertig bearbeitet haben. Wird ein Objekt während der Test-Phase zurückgewiesen und erhält den Status <b>REJECTED</b>, kann es in diesem Status nicht bearbeitet werden. Vom <b>REJECTED</b>-Status ist aber nur der Übergang zum <b>EDIT</b>-Status möglich, in dem Sie das Objekt dann wieder bearbeiten und die Beanstandungen, die sich in der Test-Phase ergaben, korrigieren können.</p>
Test-Phase	<p>In der Test-Phase prüfen Sie, ob das Objekt für den weiteren Produktions-Prozess geeignet ist. In der Test-Phase können Sie das Objekt nicht bearbeiten. Wenn Sie das Objekt ausreichend getestet haben und es den Anforderungen genügt, weisen Sie dem Objekt den <b>ACCEPTED</b>-Status zu. Wenn das Objekt den Anforderungen nicht genügt, weisen Sie das Objekt durch Wechsel in den <b>REJECTED</b>-Status zurück. Das Objekt kann dann in den <b>EDIT</b>-Status zurückversetzt werden, um nachträgliche Änderungen vorzunehmen.</p> <p><b>⚠ Schema-Versionen</b> können Sie nicht manuell den <b>ACCEPTED</b>-Status zuweisen. Dies geschieht automatisch, wenn das zugehörige Produktions-Paket den <b>ACCEPTED</b>-Status erreicht. (siehe <a href="#">hier</a>)</p>
Produktions-Phase	<p>Hat ein Objekt den <b>ACCEPTED</b>-Status erreicht, ist es für den weiteren Produktions-Prozess freigegeben. Erst dann ist es für die finale Verwendung zugelassen. Um ein Produktions-Paket von einem Schema erstellen zu können, müssen alle beteiligten Objekte den <b>ACCEPTED</b>-Status haben. Um ein Objekt als veraltet zu markieren, können Sie es, nachdem es den <b>ACCEPTED</b>-Status erreicht hat, in den <b>DEPRECATED</b>-Status versetzen. Das Objekt ist allerdings weiterhin zur Verwendung freigegeben, es sollte aber nicht mehr benutzt werden. Sie können ein als <b>DEPRECATED</b> markiertes Objekt auch wieder in den <b>ACCEPTED</b>-Status zurückversetzen.</p>

**! WICHTIG**

Sie können in einer Schema-Version zwar eine Formular-Version verwenden, die als Status nicht **ACCEPTED** (oder **DEPRECATED**) hat, die Schema-Version kann aber erst dann in den **TEST**-Status wechseln, wenn die Formular-Version den **ACCEPTED**-Status erreicht hat. Ebenso können Sie in einer Schema-Version zwar Bibliotheks-Versionen, die als Status nicht **ACCEPTED** (oder **DEPRECATED**) haben, benutzen, die Schema-Version kann aber erst dann in den **TEST**-Status wechseln, wenn alle Bibliotheks-Versionen, die die Schema-Version nutzt, den **ACCEPTED**-Status erreicht haben.

**Status-Workflow bei Produktions-Paketen:**

Phase	Beschreibung
Bearbeitungs-Phase	Produktions-Pakete können nicht manuell erstellt werden, sondern sie werden automatisch für eine Schema-Version angelegt, wenn diese vom <b>EDIT</b> -Status in den <b>TEST</b> -Status wechselt (Vgl. "Automatischer Statuswechsel", <i>Abb. C</i> ). Ein Produktions-Paket hat dann erst den <b>NEW</b> -Status, das Paket durchläuft aber automatisch einen Generierungs-Prozess: Konnte das Paket erfolgreich generiert werden, nimmt es nach der Generierung den <b>GENERATED</b> -Status an. Tritt beim Generierungs-Prozess ein Fehler auf, geht das Produktions-Paket automatisch in den <b>REJECTED</b> -Status über. Befindet sich ein Produktions-Paket im <b>REJECTED</b> -Status, wird die in ihm enthaltene Schema-Version automatisch auch in den <b>REJECTED</b> -Status versetzt (Vgl. "Automatischer Statuswechsel", <i>Abb. C</i> ). Ein Produktions-Paket, das in den <b>REJECTED</b> -Status versetzt wurde, kann nicht mehr genutzt werden und wird unbrauchbar. Für eine Schema-Version wird aber jedes Mal, wenn sie vom <b>EDIT</b> -Status in den <b>TEST</b> -Status wechselt, ein neues Produktions-Paket angelegt. Wird ein Produktions-Paket also zurückgewiesen und seine Schema-Version erhält den <b>REJECTED</b> -Status, können Sie ein neues Produktions-Paket für die Schema-Version generieren lassen, indem Sie die Schema-Version in den <b>EDIT</b> - und dann in den <b>TEST</b> -Status versetzen. Vom <b>GENERATED</b> -Status können Sie in den <b>TEST</b> -Status wechseln, um das Produktions-Paket in die Test-Phase zu versetzen. Von beiden Status können Sie manuell in den <b>REJECTED</b> -Status wechseln, z.B. wenn während der Test-Phase Beanstandungen auftreten.
Test-Phase	In der Test-Phase prüfen Sie, ob das Produktions-Paket für die Produktion geeignet ist, oder ob noch Änderungen an den Komponenten nötig sind. Getestet werden Produktions-Pakete vor allem durch <b>Dokumentvorschauen</b> mit Hilfe von Testdaten und indem Sie das Produktions-Paket in die <b>Testproduktion</b> schicken. Wenn Sie das Produktions-Paket kontrolliert haben und es den Anforderungen genügt, versetzen Sie das Paket in den <b>ACCEPTED</b> -Status. Wechselt ein Produktions-Paket in den <b>ACCEPTED</b> -Status, erhält die zugehörige Schema-Version automatisch auch den <b>ACCEPTED</b> -Status (Vgl. "Automatischer Statuswechsel", <i>Abb. C</i> ). Wenn das Produktions-Paket den Anforderungen nicht genügt, machen Sie das Paket mit Hilfe des <b>REJECTED</b> -Status unbrauchbar.

Phase	Beschreibung
Produktions-Phase	<p>Hat ein Produktions-Paket den <b>ACCEPTED</b>-Status erreicht, können Sie es produktionsbereit machen (<b>PRODUCTION</b>-Status). Im <b>PRODUCTION</b>-Status sind Produktions-Pakete für die Produktion freigegeben und können im externen Produktions-Prozess verwendet werden. Von einem Schema kann immer nur ein Produktions-Paket den <b>PRODUCTION</b>-Status besitzen, da es nur ein Paket pro Schema geben darf, das von der Produktion "abgeholt" wird. Lesen Sie dazu den Abschnitt über <b>Produktions-Pakete</b> und den Abschnitt über die <b>Umgebung des R-S</b>. Wie andere Objekte auch markieren Sie ein Produktions-Paket durch den <b>DEPRECATED</b>-Status als veraltet. Den <b>ERROR</b>-Status nimmt ein Produktions-Paket bei Fehlern an. Sie können so ein Produktions-Paket auch manuell als fehlerhaft markieren. Von allen 3 Status aus können Sie auch wieder in den <b>ACCEPTED</b>-Status zurück wechseln. Da von einem Schema immer nur ein Produktions-Paket mit <b>PRODUCTION</b>-Status existieren darf, wird ein Produktions-Paket mit <b>PRODUCTION</b>-Status automatisch in den <b>ACCEPTED</b>-Status zurückversetzt, wenn ein anderes Paket des gleichen Schemas den <b>PRODUCTION</b>-Status zugewiesen bekommt.</p>

## Status einer Objekt-Version setzen

Wie bereits im vorigen **Abschnitt** erläutert, besitzen Formular-Versionen, Schema-Versionen, Produktions-Pakete und Bibliotheks-Versionen einen Status im R-S Workflow. Hier soll nun erklärt werden, wie Sie den Status eines Objekts ändern können (Falls Sie dazu **berechtigt** sind.). Hier sehen Sie die Oberfläche, mit der Sie für Schema-Versionen, Formular-Versionen, Bibliotheks-Versionen und Produktions-Pakete einen Status setzen können. Die Vorgehensweise wird für jedes Basisobjekt im jeweiligen Kapitel noch einmal ausführlich behandelt.

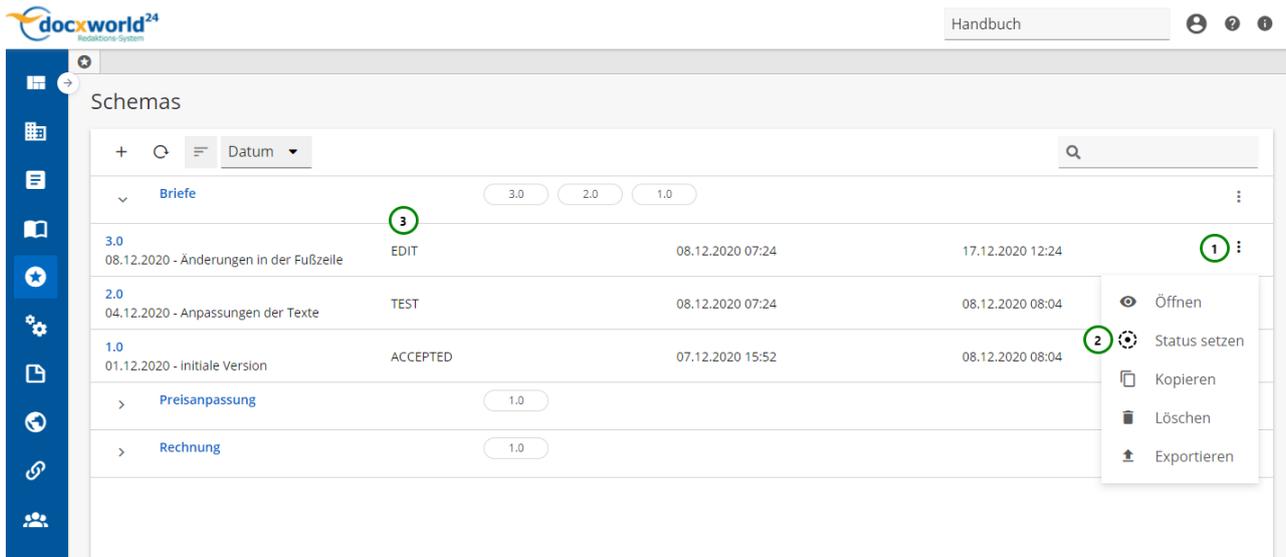


Abbildung A

Mit **Abb. A (1)** öffnen Sie das Kontextmenü der ausgewählten Schema-Version. Wählen Sie hier **Status setzen (2)**.

In der Status-Spalte **Abb. A (3)** wird Ihnen der aktuelle Status der Objekte angezeigt. Wählen Sie das Objekt aus, für das Sie einen Status setzen möchten, öffnen Sie das **⋮** Kontextmenü (1) und klicken Sie anschließend auf **Status setzen (2)**. Je nachdem, bei welchem Basisobjekt Sie den Status ändern möchten, öffnet sich folgender Dialog:

The image shows a dialog box titled "Status setzen" with a close button (X) in the top right corner. The dialog is divided into two main sections. The first section is labeled "Aktueller Status" and contains a text input field with the value "EDIT", which is highlighted by a green circle with the number "1". The second section is labeled "Neuer Status" and contains a dropdown menu with the value "TEST" and a downward arrow, also highlighted by a green circle with the number "2". At the bottom right of the dialog, there are two buttons: "Abbrechen" and "Ok".

Abbildung B

In Abb. B(1) sehen Sie den aktuellen Status des Objektes und können mit (2) den neuen Status auswählen. Eine Ausnahme bildet der Statuswechsel vom **ACCEPTED**- in den **PRODUCTION**-Status bei Produktions-Paketen. In diesem Fall erscheint folgender Dialog:

## Status setzen ×

Aktueller Status ①

Neuer Status ②

Beschreibung ③

Zeitgesteuerte Produktivsetzung  ④

Zeitpunkt Produktivsetzung ⑤

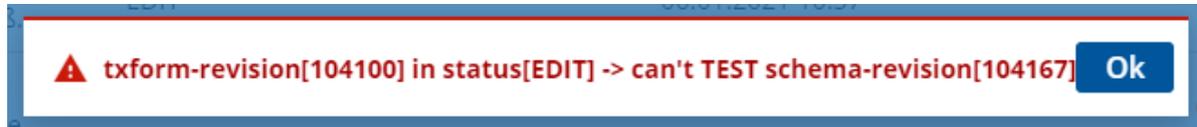
Zeitpunkt ist nicht bestimmt

Abbrechen

Abbildung C

Grundsätzlich funktioniert der Statuswechsel wie in [Abb. B](#), Sie müssen allerdings auch den **Zeitpunkt der Produktivsetzung** angeben [Abb. C \(4\)](#), also ab wann das Paket den **PRODUCTION**-Status haben soll. Dazu können Sie neben der sofortigen Produktivsetzung (Das Paket wird sofort als produktives Paket gekennzeichnet und ab sofort wird für dieses Schema immer dieses Paket von der Produktion benutzt.) auch einen Zeitpunkt (5) für die Produktivsetzung angeben. Setzen Sie dazu den Haken in (4) auf aktiv. Das Paket wird dann zum angegebenen Zeitpunkt in den **PRODUCTION**-Status versetzt. Das Paket, das bisher **PRODUCTION**-Status hatte, bekommt dann **ACCEPTED**-Status. Lesen Sie dazu den [Abschnitt über Produktions-Pakete](#).

⚠ Wenn Sie eine **Schema-Version** vom **EDIT**- in den **TEST**-Status versetzen wird geprüft, ob das Transaktions-Formular und alle eingebundenen Bibliotheken **ACCEPTED**-Status erreicht haben. Andernfalls werden Ihnen folgende Fehlermeldungen angezeigt:



Die Schema-Version kann erst in den **TEST**-Status versetzt werden, wenn sich alle eingebundenen Komponenten im **ACCEPTED**-Status befinden.

## Arbeitsbereiche

Arbeitsbereiche dienen im R-S dazu, voneinander unabhängige, abgetrennte Arbeitsumgebungen zu schaffen. Ein Arbeitsbereich enthält seine eigenen R-S Objekte, auf die von anderen Arbeitsumgebungen nicht zugegriffen werden kann. Elemente, die Sie in einem Arbeitsbereich erstellen, existieren also auch nur in diesem Arbeitsbereich und alle R-S Abläufe innerhalb eines Arbeitsbereichs haben keine Auswirkungen auf andere Arbeitsbereiche. Dies ermöglicht eine Aufteilung und Trennung der Arbeitsprozesse in mehrere Arbeitsbereiche, sodass sich Dokumente unterschiedlicher Produktionen nicht vermischen. Wenn Sie mit dem R-S arbeiten, tun Sie dies immer innerhalb eines Arbeitsbereichs. Es muss daher mindestens ein Arbeitsbereich existieren. R-S Nutzer haben nur Zugriff auf die ihnen zugewiesenen Arbeitsbereiche.

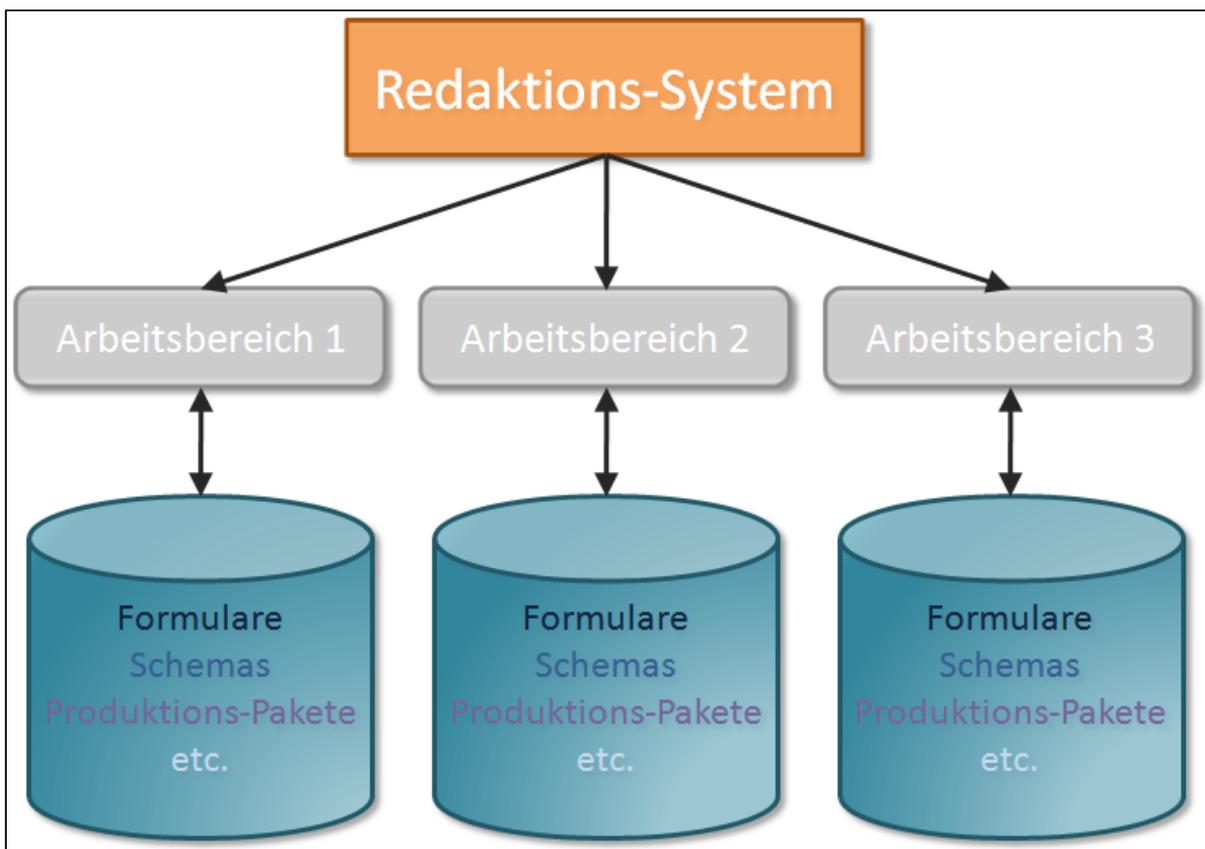


Abbildung A

Es ist nicht möglich, Objekte direkt zwischen den Arbeitsbereich auszutauschen oder aus einem Arbeitsbereich auf fremde Objekte zuzugreifen. Durch **Import/Export** von R-S Objekten und Ressourcen ist es aber möglich, ein Objekt aus einem Arbeitsbereich zu exportieren und im anderen Arbeitsbereich wieder zu importieren. Wie Sie Objekte exportieren und importieren, lernen Sie in den jeweiligen Objekt-Kapiteln.

Wenn Sie sich im R-S mit Ihrem Nutzernamen **anmelden**, wird Ihnen eine Liste der Ihnen zugewiesenen Arbeitsbereiche angezeigt. Wählen Sie daraus den Arbeitsbereich aus, in dem Sie arbeiten möchten. Sie können den Arbeitsbereich auch später im R-S noch wechseln. Sind Sie nur einem Arbeitsbereich zugewiesen, werden Sie dort automatisch angemeldet. Sind Ihnen keine Arbeitsbereiche zugewiesen, können Sie nicht mit dem R-S arbeiten. Wenden Sie sich dann bitte an Ihren System-Administrator. Momentan ist es für Endkunden noch nicht möglich, Arbeitsbereiche direkt über den Admin-Client zu verwalten.

## Ressource-Typen

Ressourcen sind die elementaren Objekte im R-S, um Dokumente zu gestalten. In ihnen speichern Sie Daten und Informationen und machen diese im Schema verfügbar (Schemas werden später behandelt). Es kann sich dabei um R-S interne Daten handeln, z.B. um geschriebenen Text in einem RichText oder einen Wert in einer Variablen, es ist aber auch möglich, externe Daten, z.B. Bilder und Dateien mit Ressourcen verfügbar zu machen. Im R-S gibt es je nach Zweck und möglichem Inhalt verschiedene Ressourcen.

### Klassische Ressourcen zur Speicherung von Daten:

-  **Variablen:** Sie speichern Werte und Text.
-  **Binaries:** Sie speichern Bilder und Dateien.
-  **RichText:** Sie ermöglichen die Erstellung von formatiertem Text.

### Ressourcen zum Arbeiten mit Beilegern und WhiteSpaces:

-  **Physikalische Beileger:** Definieren einen physikalischen Beileger für das Schema.
-  **Logische Beileger:** Definieren einen logischen Beileger (und sein Design) für das Schema.
-  **WhiteSpaces:** Ermöglichen das dynamische Füllen freier Flächen im Dokument mit Inhalt.

### Dynamische Ressourcen:

-  **Regeln:** Kontrollstrukturen, die abhängig von Bedingungen unterschiedliche Werte annehmen können.

### Ressourcen zum Arbeiten mit Datenströmen:

-  **Datenstrom-Variablen:** Durch Formular definierte Datenfelder, die Werte aus Datensystemen verfügbar machen.

Daneben gibt es noch die **!** **Erforderliche Ressourcen**. Sie ermöglichen das Arbeiten mit externen Ressourcen in Bibliotheken. Erforderliche Ressourcen enthalten keinen eigenen Inhalt, es handelt sich dabei um eine Ressource-Deklaration, um in einer Bibliothek Ressourcen nutzbar zu machen, die erst beim Einbinden der Bibliothek zur Verfügung stehen werden. Erforderliche Ressourcen werden in diesem Kapitel nicht behandelt, lesen Sie dazu den **Abschnitt über Bibliotheken** .

## Variablen

Variablen sind die grundlegenden Ressourcen im R-S. Es handelt sich dabei um einfache Platzhalter für benutzerdefinierte Werte. Das Symbol für Variablen im R-S ist **X**.

Der Wert einer Variablen im R-S wird als **STRING** (schlicht eine Zeichenkette) eingegeben. Je nach Form des STRINGS wird aber zwischen verschiedenen Datentypen unterschieden. Enthält die Variable z. B. nur Ziffern, kann Sie als **INTEGER** (Zahl), **NUMBER** (Zahl), **REAL** (Kommazahl) oder auch **BOOLEAN** (Wahrheitswert) behandelt werden. Eine Variable ist grundsätzlich immer eine Textvariable. Speichern Sie allerdings in einer Variablen nur Ziffern, können Sie mit der Variablen auch als Wert weiterarbeiten und sie beispielsweise in Rechenoperationen verwenden. Speichern Sie in einer Variablen nur eine 1 oder 0, können Sie die Variable als boolesche Variable verwenden. Die verschiedenen Datentypen im R-S zeichnen sich durch ihre Form aus. Sinnvolle Datentypen für Variablen sind:

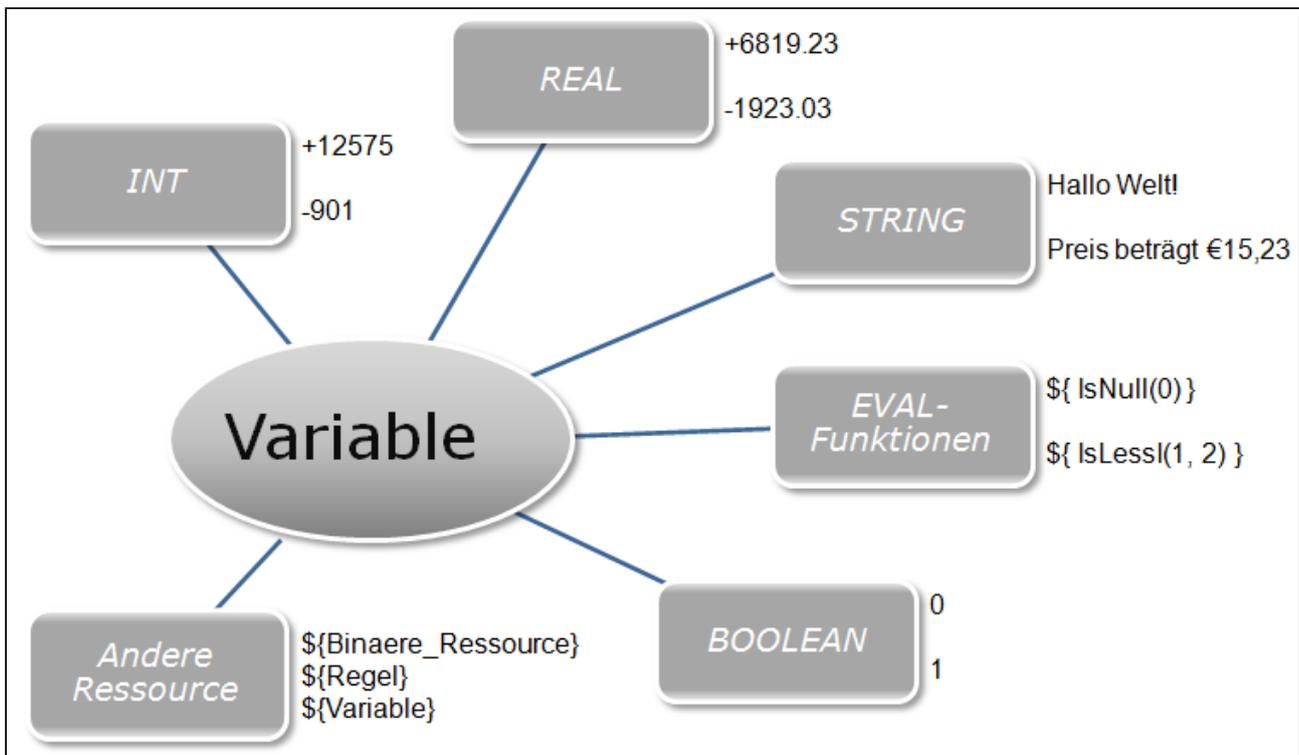


Abbildung A

Datentyp	Beschreibung	Wertebereich
STRING	Zeichenfolge (Text). Der Inhalt einer Variable ist immer auch ein String.	Beliebig viele beliebige Zeichen
INTEGER / NUMBER	Ganzzahliger numerischer Wert mit positivem oder negativem Vorzeichen. Das Vorzeichen kann vor oder nach dem ganzzahligen numerischen Wert stehen.	-2147483648 bis +2147483648

Datentyp	Beschreibung	Wertebereich
FLOAT	Numerischer Wert mit Nachkommastellen mit positivem oder negativem Vorzeichen. Das Vorzeichen kann vor oder nach dem ganzzahligen Wert stehen.	-3,4E-4932 bis +3,4E-4932
BOOLEAN	Schaltvariable zum Speichern von 2 Werten: Wahr oder Falsch	true / false bzw. yes / no bzw. 1 / 0
DATETIME	Datumsformat nach ISO 8601	Datum der Form yyyy-mm-dd, z. B.: 2013-04-08
RESOURCE NAME	Name einer R-S Ressource, z. B. einer anderen Variable	Name einer zulässigen und verfügbaren R-S Ressource
OBJECT NAME	Name eines R-S Objekts, z. B. eines physikalischen Beileger	Name eines verfügbaren R-S Objekts

Neben den Typen in der Tabelle existieren im R-S noch weitere Datentypen. Weitere Informationen darüber finden Sie [hier](#).

Sie können den Text in Variablen auch formatieren, indem Sie **Inline-Sequenzen** verwenden. Variablen sollten jedoch in erster Linie zur Speicherung von Werten dienen, nicht zur Darstellung größerer Texte. Verwenden Sie **RichTexte**, um einfacher formatierten Text zu erstellen.

 Enthalten Variablen gemischte Datentypen, z. B. Ziffern und Buchstaben, werden Sie einfach als STRING (Text) behandelt.

### **EVAL-Anweisungen und dynamische Einbindung innerhalb von Variablen**

Neben der Möglichkeit, den Inhalt einer Variable statisch durch einen festen Wert zu definieren, können Sie innerhalb von Variablen auch EVAL-Anweisung verwenden. Mit Hilfe von EVAL-Anweisungen können Sie andere Ressourcen und dynamische Werte in eine Ressource einbinden. EVAL-Anweisungen werden immer mit **`\${** eingeleitet und durch **}** geschlossen. In den nachfolgenden Abschnitten über die anderen Ressourc-Typen lernen Sie noch detaillierter, wie man mit EVAL-Anweisungen arbeitet. Lesen Sie aber unbedingt auch den [Abschnitt über EVAL-Anweisungen](#).

### Einbinden anderer Ressourcen

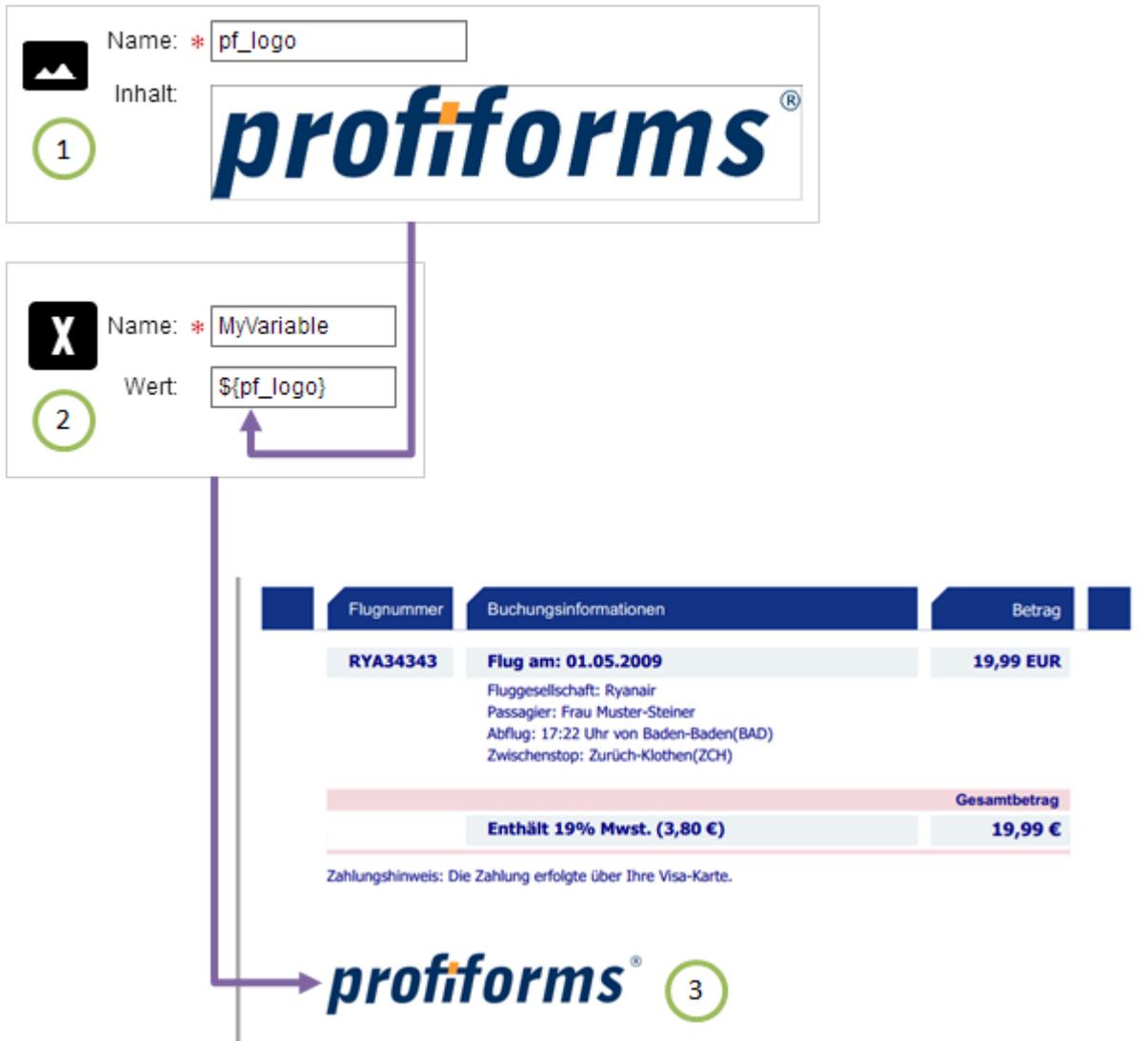


Abbildung B

In Abb. B wird gezeigt, dass in Variablen auch die dynamische Einbindung anderer Ressourcen erlaubt ist. Der Wert der **Variablen** `MyVariable` ist `${pf_logo}`. Durch diese dynamische Einbindung enthält die Variable einen Verweis auf die Ressource `pf_logo`. (Binäre Ressourcen werden in einem späteren Abschnitt vorgestellt, hier enthält die Ressource ein Bild). Wird die Variable dann in das Dokument eingefügt, führt das R-S die EVAL-Anweisung aus und fügt automatisch die Ressource, auf die verwiesen wurde, ein Abb. B (3).

**i** Die Variable `MyVariable` enthält nicht die Ressource `pf_logo`, sondern einen Verweis auf die Ressource: `${pf_logo}`. Bei Aufruf der Variablen wird dann der Verweis zurückgegeben.

⚠ Beachten Sie bitte, dass in einer Variable vor einer EVAL-Anweisung, die eine **binäre** Ressource dynamisch einbindet, kein Text (auch keine Leerzeichen oder Zeilenumbrüche) stehen darf. Text nach der Anweisung wird ignoriert. Dies gilt jedoch nicht für die dynamische Einbindung von z. B. anderen Variablen oder für andere EVAL-Anweisungen. Diese können mit Text kombiniert werden.

⚠ Sollen Variablen bei der Ausgabe von Dokumenten verwendet werden (z. B. als Feldinhalte von Dokumenten), so ist darauf zu achten, dass Variablen bei ihrer Verwendung aufgelöst werden und damit eine Typ-Prüfung der Variablen stattfindet. Verweist der Inhalt einer Variable zum Beispiel auf eine binäre Ressource (z. B. Logo) und kann der Feld-Typ auf dem Dokument nur Texte aufnehmen, so wird auf dem Dokument keine binäre Ressource ausgegeben, sondern im o.g. Beispiel der Name der binären Ressource.  
Bitte verwenden Sie insbesondere Ressourcen vom Typ RichText nicht innerhalb von Variablen zur Ausgabe auf Dokumenten.

### Dynamische Werte

Neben der dynamischen Einbindung von Ressourcen können Sie in einer Variable (und auch in anderen Ressourcen) mit Hilfe von EVAL-Anweisungen auch Werte dynamisch berechnen. Der Wert wird dann erst berechnet, wenn die Variable verwendet wird.

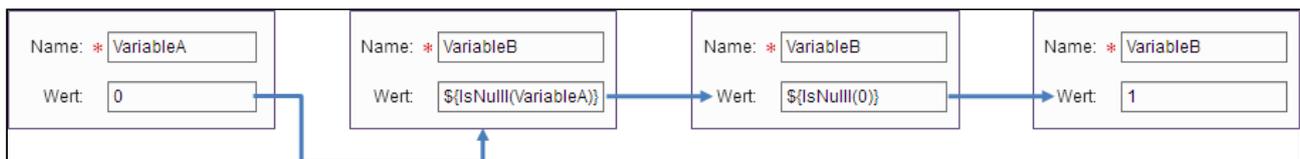


Abbildung C

Der Wert von VariableB hängt von VariableA ab. In diesem Beispiel wird VariableB immer den Wert 1 annehmen, da VariableA den statischen Wert 0 hat. Würde VariableB allerdings in der EVAL-Anweisung z. B. eine **Regel** oder eine **Datenstrom-Variable** verwenden, so wäre auch der Wert von VariableB volatil, da der Eingangswert aus einer Regel oder einer Datenstrom-Variablen sich von Dokument zu Dokument ändern kann. Sie benutzen hier übrigens auch wieder dynamische Einbindung, indem Sie die Variable VariableA einbinden. Sie betten diese Ressource dann in eine EVAL-Anweisung ein, um einen Wert dynamisch (abhängig von VariableA) berechnen zu lassen bzw. eine Operation mit dieser Ressource durchzuführen. IsNull() ist eine vorgefertigte Funktion. Im R-S gibt es eine Reihe dieser vordefinierten Funktionen. Eine Liste der Funktionen finden Sie [hier](#). Grundsätzlich können Sie dynamische Einbindungen und EVAL-Funktionen unbegrenzt tief verschachteln. Achten Sie aber darauf, dass die Ressource, die Sie verwenden, den erwarteten Datentyp hat: `${IsNull("01.01.2013")}` ist kein gültiger Ausdruck.

## RichText

Ein RichText erlaubt die Editor gestützte Speicherung und Darstellung von formatiertem Text. Im Gegensatz zu Variablen, in denen Sie Text nur durch **Inline-Sequenzen** formatieren können, eignen sich RichTexte deshalb wesentlich besser für Texte, die später formatiert im Dokument dargestellt werden sollen. Da RichTexte allerdings intern immer mit Formatierungsinformationen abgespeichert werden und sie nicht wie Variablen in Datentypen umgewandelt werden, können Sie RichTexte in EVAL-Anweisungen nicht so verwenden wie Variablen. RichTexte sollten daher wirklich nur für Texte genutzt werden. Das Symbol für RichTexte im R-S ist .

### Formatierungsmöglichkeiten in einem RichText

Im Gegensatz zu Variablen stehen Ihnen bei RichTexten folgende Möglichkeiten zur Verfügung: (Wie Sie den RichText-Editor benutzen, lernen Sie [hier](#))

#### Schriftart und Schriftgröße

Arial

11.5

Sie können innerhalb eines RichText jede verfügbare Schriftart verwenden. Beachten Sie, dass in einem RichText nur die Schriftarten verfügbar sind, die das Transaktions-Formular vorgibt.

#### Schriftauszeichnungen

*I* **B** U

Sie können Schrift **fett**, *kursiv* und unterstrichen darstellen (oder Kombinationen davon). Die verfügbaren Schriftauszeichnungen werden ebenfalls vom Transaktions-Formular vorgegeben.

#### Schriftfarbe

A

Sie können entweder aus der vorgegeben Farbpalette eine Schriftfarbe wählen oder sie manuell durch einen hexadezimalen Farbcode angeben.

#### Textausrichtung

Sie haben 4 Auswahlmöglichkeiten zur Textausrichtung: Linksbündig, zentriert, rechtsbündig oder Blocksatz. Die Standardeinstellung ist linksbündig.

### EVAL-Anweisungen und dynamische Einbindung innerhalb von RichTexten

Sie können in RichTexten, wie in allen anderen Ressourcen mit Textinhalt, EVAL-Anweisungen und dynamische Einbindung verwenden. Da RichTexte für Textabschnitte in Dokumenten verwendet werden, werden oft **Datenstrom-Variablen** dynamisch eingebunden, um den Text zu personalisieren (Datenstrom-Variablen sind Werte, die das Transaktions-Formular definiert und die das Datensystem liefert. In ihnen werden die

Informationen zum aktuellen Formular mitgeliefert):

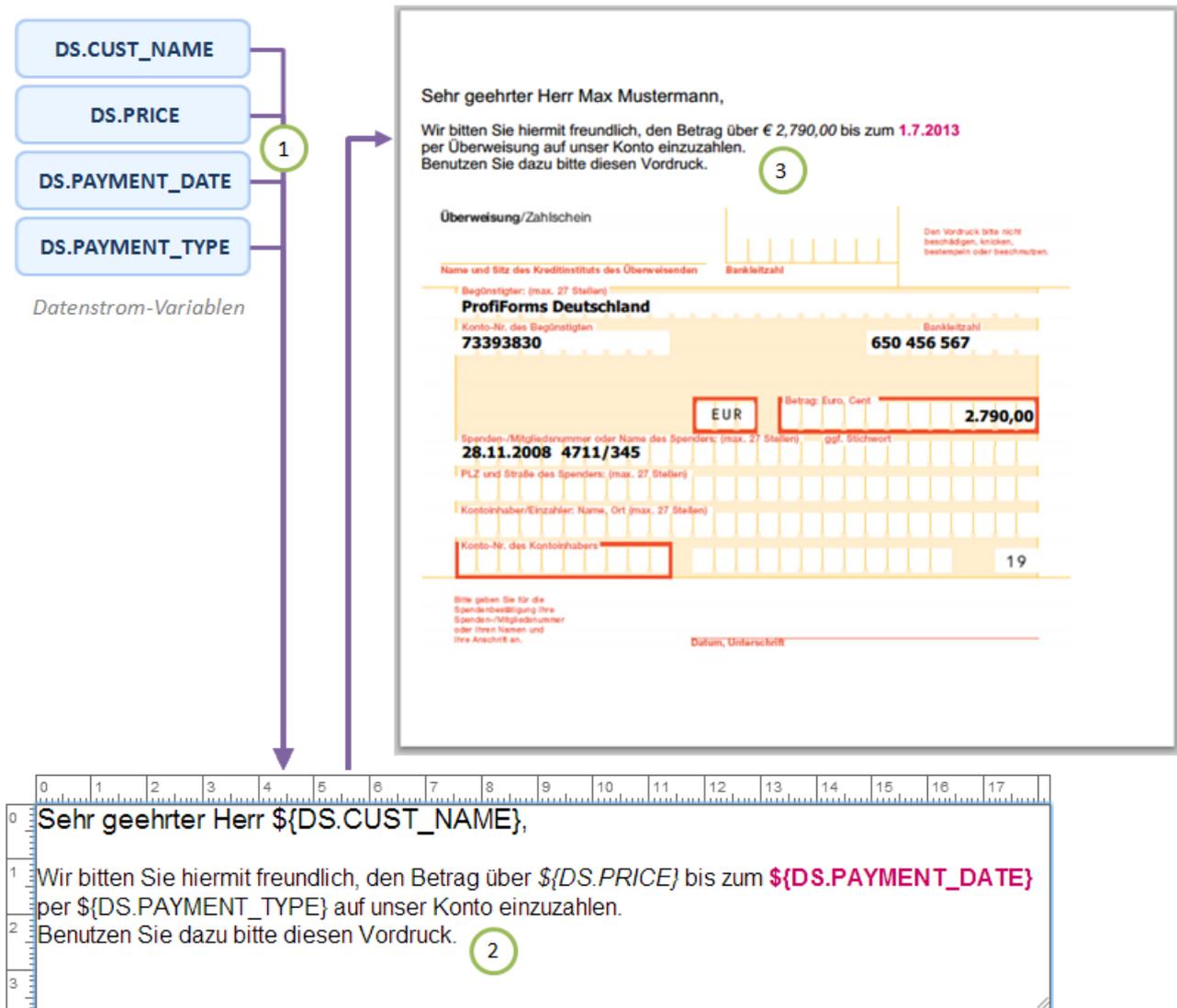


Abbildung A

In *Abb. A* wird die dynamische Einbindung von **Datenstrom-Variablen** in einen RichText gezeigt. Das Transaktions-Formular stellt 4 Datenstrom-Variablen bereit *Abb. A* (1) die in den RichText eingebunden werden, um den Text zu personalisieren (2). Wie Sie sehen, können Sie eingebundene Ressourcen direkt im Textfluss des RichText mit formatieren, indem Sie die Formatierung auf die gesamte EVAL-Anweisung anwenden. So wird z.B. DS.PRICE in (3) kursiv und DS.PAYMENT\_DATE rot dargestellt. Selbstverständlich lassen sich nicht nur Datenstrom-Variablen, sondern auch andere R-S Ressourcen in RichTexte einbinden.

### **Interne Darstellung von RichTexten**

Das R-S implementiert RichText Ressourcen nach dem XFA-Standard 2.0. Weitere Informationen finden Sie hier: [W3C XFA-Spezifikation](#).

#### **Der Ausdruck**

```
${ IsNullI(Eins) }
```

Würde 1 ergeben, obwohl der Inhalt des RichText "1" ist. Grund ist, dass der RichText nicht in einen Integer mit dem Wert 1 konvertiert wird und der Wert deshalb undefiniert ist ( = 0 ). Wäre die Ressource *Eins* eine Variable, so wäre das Ergebnis 0, da Variablen in Integer konvertiert werden können. Sie können RichTexte also in EVAL-Anweisungen nicht auf dieselbe Weise nutzen wie Variablen. Natürlich können Sie aber RichTexte dynamisch in eine andere Ressource einbinden, z.B. in eine Variablen mit dem Ausdruck:

```
${ NameDesRichText }
```

Die Formatierung des RichText bleibt dabei erhalten.

### **Hinweise zur praktischen Anwendung und bekannte Einschränkungen**

Im Folgenden sollen einige Hinweise zur praktischen Anwendung von RichText-Ressourcen gegeben werden. Dabei sollen nicht nur Möglichkeiten, sondern auch bekannte Einschränkungen bei der Nutzung von RichText-Ressourcen gegeben werden.

#### **Tabulatoren**

- haben einen festen Abstand (können nicht verändert werden).
- sind immer linksbündig.
- können nur in linksbündig ausgerichteten Absätzen verwendet werden (nicht rechtsbündig und nicht zentriert).

### Einbinden von Ressourcen und Ausdrücken über \${...}

- es können Ressourcen gestaffelt eingebunden werden (z.B. wird über eine Regel auf Inhalte von Variablen zugegriffen).
- **INLINE-Sequenzen** werden im RichText nicht ausgewertet, ebenfalls nicht über eingebundene Ressourcen.
- besitzen eingebundene Ressourcen eine Formatierung über RichText, so bekommt diese Formatierung Vorrang vor der Formatierung im übergeordneten RichText.
- besitzen eingebundene Ressourcen KEINE Formatierung, so wird die Formatierung des übergeordneten RichText angewendet.



#### Weitere bekannte Einschränkungen

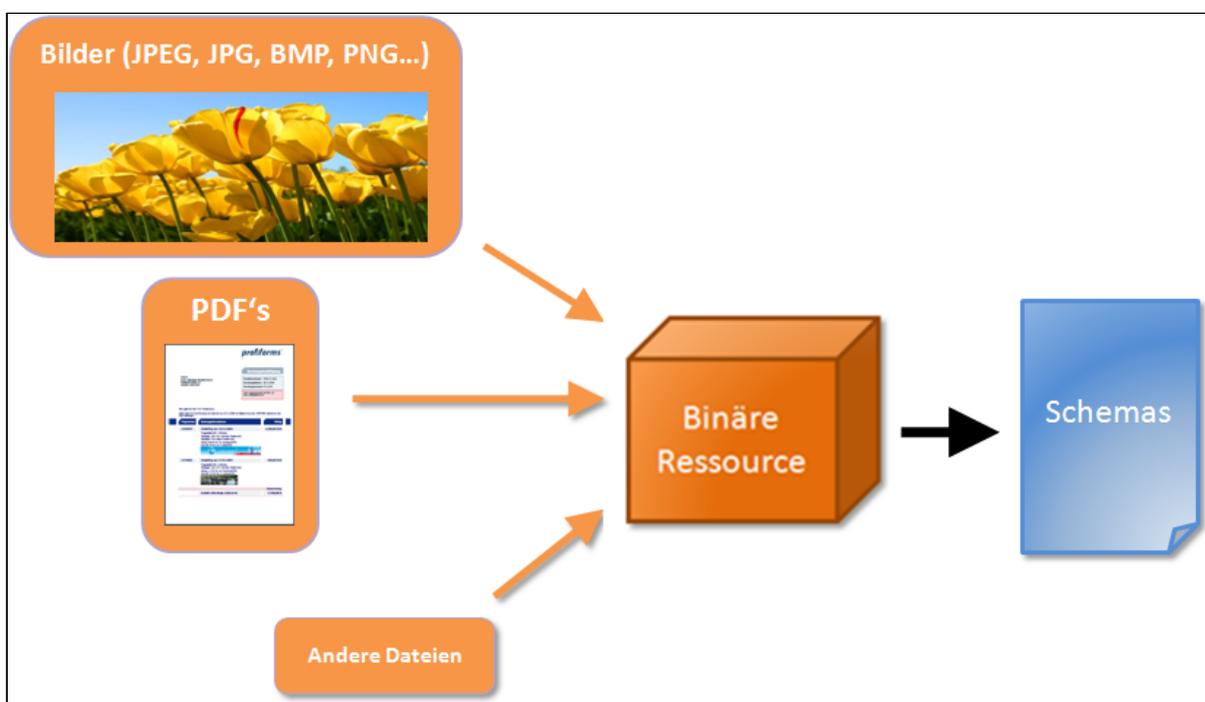
- Sie können in einen RichText keine binären Ressourcen einbinden.
- Werden über Ausdrücke \${...} Ressourcen vom Typ RichText eingebunden, so werden aus dem eingebundenen/incluierten RichText-Ressourcen alle Container-Elemente wie zum Beispiel Absatzformatierungen und Zeilenumbrüche entfernt!

## Binäre Ressourcen

Binäre Ressourcen ermöglichen es Ihnen, externe Dateien im R-S verfügbar zu machen. Binäre Ressourcen repräsentieren also eine im R-S gespeicherte, externe Datei. Die Hauptaufgabe von binären Ressourcen besteht in der Darstellung von Grafiken: Da Sie im R-S keine Grafiken direkt erstellen können, wie sie es z.B. für Texte können, müssen Sie fertige Grafiken in Ihr Projekt einbinden. Dies machen Sie mit binären Ressourcen. Das Symbol für binäre Ressourcen im R-S ist .

### Inhalte von binären Ressourcen

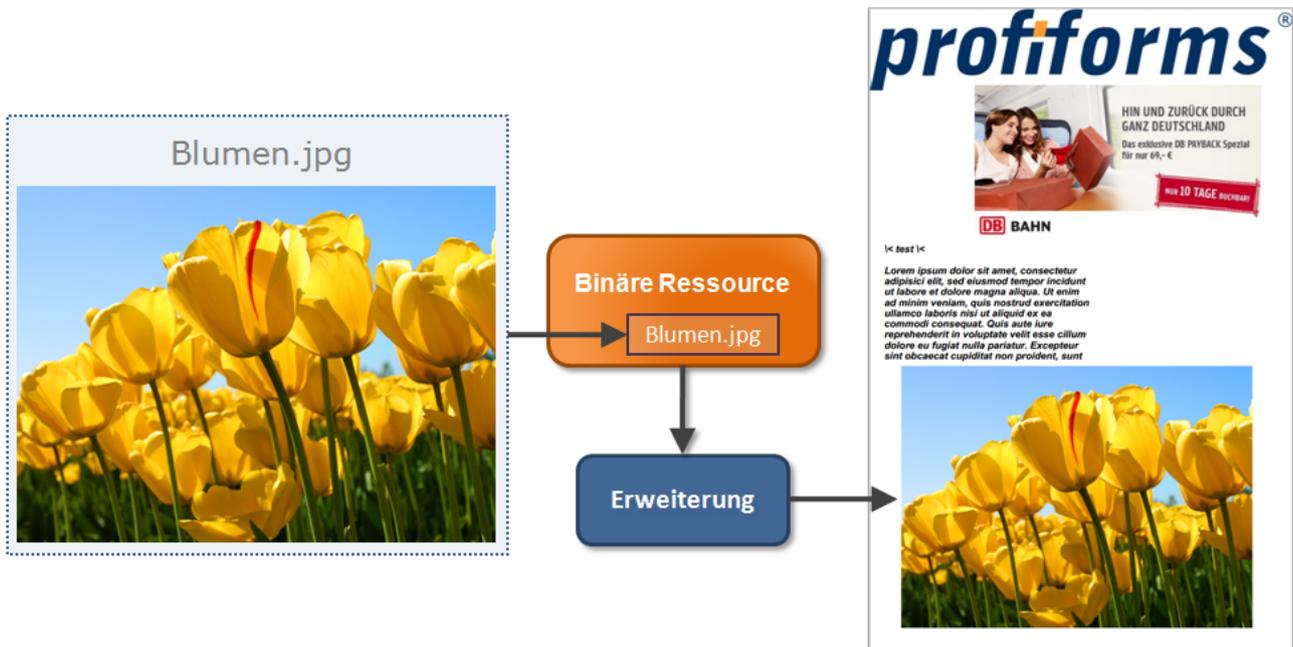
Mit binären Ressourcen können Sie grundsätzlich beliebige Dateien einbinden. Welche Dateierweiterungen für den Import erlaubt sind, ist in der **R-S Server Konfigurationsdatei** festgelegt. Dort sind auch andere Eigenschaften wie die Maximalgröße definiert. Im Auslieferungszustand des R-S sind das die Dateien, die der ReportWriter verarbeiten kann. Standardmäßig sind das: Bilder in Form von *.bmp*, *.tiff*, *.jpeg*, *.jpg*, *.png*, *.gif* Dateien sowie *.xdp*'s und *.pdf*'s.



Je nachdem, welche Software Sie zur Dokumenterstellung verwenden, können auch andere Dateien verarbeitungsfähig sein. Wenden Sie sich an Ihren R-S Administrator, wenn Sie sich nicht sicher sind, welche Dateien verwendet werden können. Versuchen Sie, eine Datei mit einer nicht erlaubten Endung hochzuladen, bricht das R-S mit einer Fehlermeldung ab. Dateien werden außerdem einer Datei- und Anti-Virenprüfung unterzogen. Die Größe der Grafik in einer binären Ressource wird bei Einbindung in das Dokument automatisch an den verfügbaren Platz angepasst. Eine Datei, die Sie in eine binäre Ressource laden, wird in die Ressource kopiert. Es besteht also kein Verweis zur Datei und selbst wenn Sie die Datei löschen oder verändern, bleibt die Binäre Ressource unverändert.

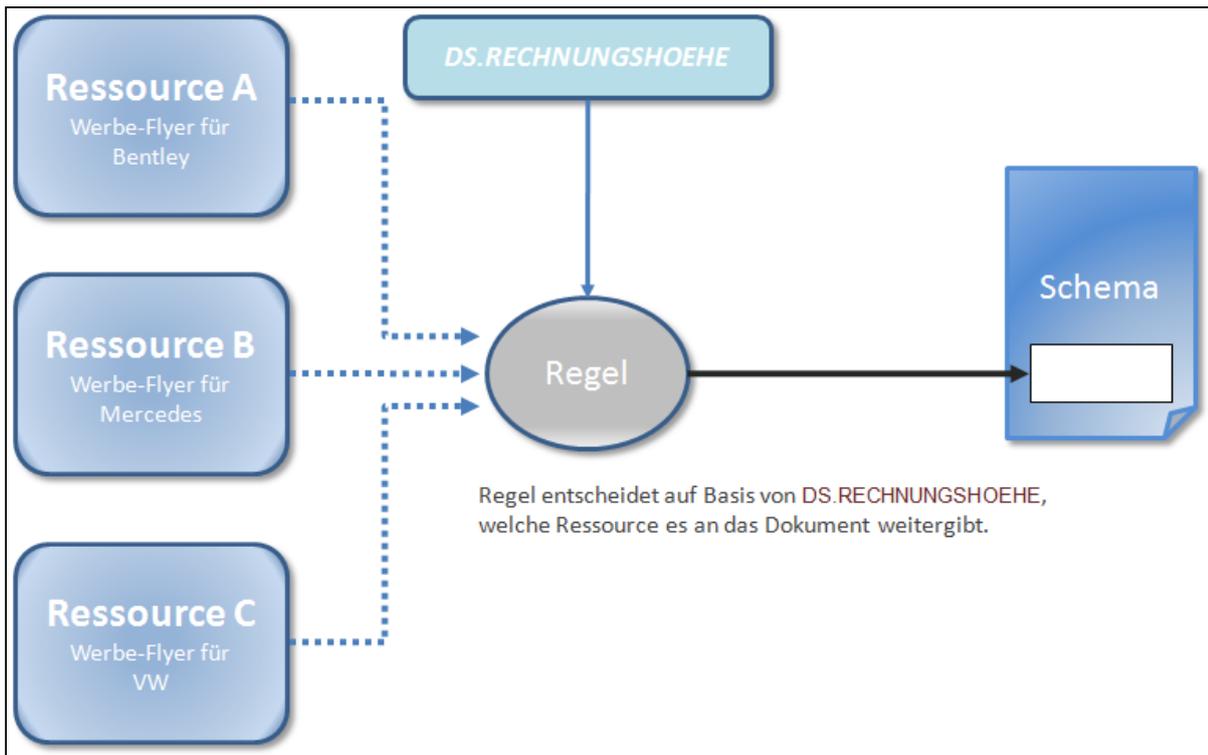
### Verwendung von Binären Ressourcen

Wie bei jedem Ressource-Typ besteht die einfachste Art, eine Ressource in ein Dokument einzufügen darin, die Ressource direkt und statisch einzubinden:



(Erweiterungen werden benutzt, um Ressourcen in das Dokument einzufügen. Sie werden später vorgestellt.)

Binäre Ressourcen werden aber auch oft dynamisch eingebunden, um das Dokument zu individualisieren. Ein Anwendungsgebiet davon ist personalisierte Werbung. Haben Sie zum Beispiel 3 verschiedene Werbe-Flyer und Sie möchten jeweils den Flyer auf dem Dokument platzieren, der am besten zum Kunden passt, können Sie die Ressourcen nicht direkt in das Dokument einbinden. Sie laden die Flyer in 3 binäre Ressourcen. Dann muss für jeden Kunden individuell entschieden werden, welcher Flyer (welche binäre Ressource) in das Dokument eingebunden werden soll. Um dieses Verhalten zu implementieren, werden Regeln und dynamische Einbindung genutzt. Regeln haben Sie noch nicht kennengelernt, dynamische Einbindung wurde aber schon in den vorigen Abschnitten vorgestellt. Sie sollten sich zunächst nur mit dem Konzept vertraut machen:



Hier ist es z.B. denkbar, die Ressource aufgrund einer **Datenstrom-Variablen** (Datenstrom-Variablen sind Werte, die vom Formular geliefert werden) zu bestimmen (hier: DS.RECHNUNGSHOEHE). Bei einer hohen Rechnungssumme wird Werbung für ein Luxuswagen in das Dokument eingefügt, bei geringer Rechnungssumme Werbung für einen Mittelklassewagen. Beispiele zur Einbindung von binären Ressourcen mit Hilfe von Regeln finden Sie im **nächsten Abschnitt**.

## Regeln

Regeln sind Kontrollstrukturen, die abhängig von Bedingungen unterschiedliche Werte annehmen können. Eine Regel enthält also eine oder mehrere Bedingungen und eine oder mehrere Auswahlmöglichkeiten. Die Auswahlmöglichkeit, die alle Bedingungen erfüllt, liefert den Rückgabewert der Regel.

Dazu gibt es in einer Regel Spalten und Zeilen: Die Spalten sind die Bedingungen der Regel. Jede Spalte enthält einen Wert, die Bedingung dieser Spalte. Die Zeilen sind die alternativen Auswahlmöglichkeiten für die Regel. Jede Zeile gibt pro Spalte einen Wert an. Die Zeile, in der in jeder Spalte der Zeilenwert die Bedingung der Spalte erfüllt, ist die gültige Auswahl für die Regel und ihr Rückgabewert der Wert der Regel. Dieser kann einen beliebigen Datentyp haben (*Datentypen*: Siehe **Variable**). Eine Regel kann auch eine EVAL-Anweisung / einen Verweis auf eine Ressource zurückgeben. Das Symbol für Regeln im R-S ist .

### Aufbau einer Regel im R-S

#	Erstzulassung	Modell	Preis	Will_verkaufen	PS	Rückgabewert
1	01.01.2013 DATE	RS5 STRING	75.500 REAL	TRUE BOOLEAN	450 NUMBER	
1	01.01.2012	A5	38.900	FALSE	150	\$(Angebot1)
2	01.01.2011	S5	55.400	TRUE	333	\$(Angebot2)
3	01.01.2013	RS5	75.500	TRUE	450	\$(Angebot3)
4	*	*	*	*	*	Es wurde leider kein passendes Angebot gefunden

Abbildung A

In *Abb. A* sehen Sie den Aufbau einer Regel mit Spalten (1) und Zeilen (2). Eine Spalte definiert genau einen Wert als Bedingung. In *Abb. A* sind das:

Spaltenname	Erstzulassung	Modell	Preis	Will_Verkaufen	PS
<b>Wert (Bedingung)</b>	01.01.2013	RS5	75.500	TRUE	450

Diese Spaltenwerte sind die Bedingungen der Regel, auf die die Zeilen in den entsprechenden Spalten geprüft werden. Der Wert einer Spalte kann einen der folgenden *Datentypen* haben (die in *Abb. A* alle verwendet wurden):

- NUMBER (*INTEGER*)
- REAL
- STRING
- BOOLEAN
- DATE

In *Abb. A* ist Zeile #3 die gültige Auswahl für die Regel, da alle Werte in der Zeile identisch zum jeweiligen Spaltenwert sind. Die Regel hat also den Wert `${Angebot_3}` (3) . Wird die Regel aufgerufen, so ist das Ergebnis:

```
Aufruf (EVAL-Anweisung):  ${NameDerRegel}
Rückgabewert:             ${Angebot_3}
Endwert:                  Verweis auf "Angebot_3"
```

Mit `${Angebot_3}` wird eine Ressource mit dem Namen `Angebot_3` zurückgegeben, in diesem Fall z.B. eine binäre Ressource, die das entsprechende Angebot als Bild enthält. Die Verwendung dynamischer Anweisungen wird im Folgenden vorgestellt.

**i** Der Operator `*` ist immer wahr. Enthält eine Zeile also in einer Spalte den Wert `'**'`, so erfüllt die Zeile die Bedingung dieser Spalte immer, unabhängig von der Spaltenbedingung. Mehr dazu im Abschnitt über **boolsche Operatoren**.

### **EVAL-Anweisungen und dynamische Einbindung innerhalb einer Regel**

Im oberen Abschnitt wurden in der Regel, abgesehen von den Rückgabewerten, nur statische Werte verwendet. In der Praxis werden Sie in Regeln jedoch fast immer auch dynamische Werte in Spalten und Zeilen verwenden, denn in einer Regel, in der alle Werte schon statisch feststehen, steht auch der Rückgabewert der Regel schon fest - Man könnte den Wert auch einfach einer Variablen zuweisen. Der Zweck von Regeln besteht aber darin, eine Möglichkeit zu schaffen, für unterschiedliche Eingangswerte ein unterschiedliches Verhalten zu implementieren.



Abbildung B

Typischerweise werden in einer Regel **Datenstrom-Variablen** verarbeitet, z.B. um dynamisch eine Ressource anzuwählen:

Beispiel
<p>Je nach Nationalität des Kunden soll dessen Nationalflagge angezeigt werden. Die Nationalität des Kunden wird durch die Datenstrom-Variable DS.NATIONALITY geliefert - Bei jedem Kunden individuell.</p> <p>Um abhängig von DS.NATIONALITY die richtige Ressource auszusuchen, benötigt man eine Regel.</p> <p>Diese liefert, je nach Wert von DS.NATIONALITY, die richtige Ressource zurück. Ruft man die Regel auf, erhält man automatisch die richtige Ressource.</p>

Mögliche Werte von DS.NATIONALITY :

Wert	GER	CH	AUT
Kunde kommt aus	Deutschland	Schweiz	Österreich

Binäre Ressourcen, die die Flaggen enthalten:

Name der Ressource	ger_flag	ch_flag	at_flag
Inhalt			

Dazugehörige Regel:

#	Nationalität \${DS.nationality} STRING	Rückgabewert		
1	GER	\${ger_flag}		
2	CH	\${ch_flag}		
3	AT	\${at_flag}		

Abbildung C

In *Abb. C* wird der Spaltenwert durch die dynamische Einbindung der Datenstrom-Variablen `DS.NATIONALITY` gebildet. Die Spalte hat also den Wert von `DS.NATIONALITY`. Die obige Tabelle zeigt uns, dass die Datenstrom-Variablen 3 Werte annehmen kann. Diese finden sich als Zeilen in der Regel wieder. Je nach aktuellem Wert von `DS.NATIONALITY` wird dann die dazu passende Ressource mit der passenden Flagge zurückgegeben. Hat die Datenstrom-Variablen beispielsweise den Wert **CH**, trifft die 2. Spalte auf die Bedingung zu und der Wert der Regel ist `${ch_flag}`, also die Binäre Ressource `ch_flag` mit der Schweizer Flagge. Die Regel kann dann in das Dokument eingebunden werden. Im Dokument wird dann die richtige Flagge angezeigt.

### Wertfindung innerhalb einer Regel

Gibt es in der Regel mehrere Spalten, die alle Spaltenbedingungen erfüllen, dann ist die oberste Regel in der Liste die zutreffende Regel und ihr Rückgabewert der Wert der Regel. Alle danach kommenden Zeilen werden nicht beachtet. (Das ist auch in Abb. A der Fall, denn Zeile #4 erfüllt ebenfalls alle Spaltenbedingungen, Zeile #3 steht jedoch weiter oben). Nachfolgend ist dargestellt, wie der Wert einer Regel bestimmt wird:

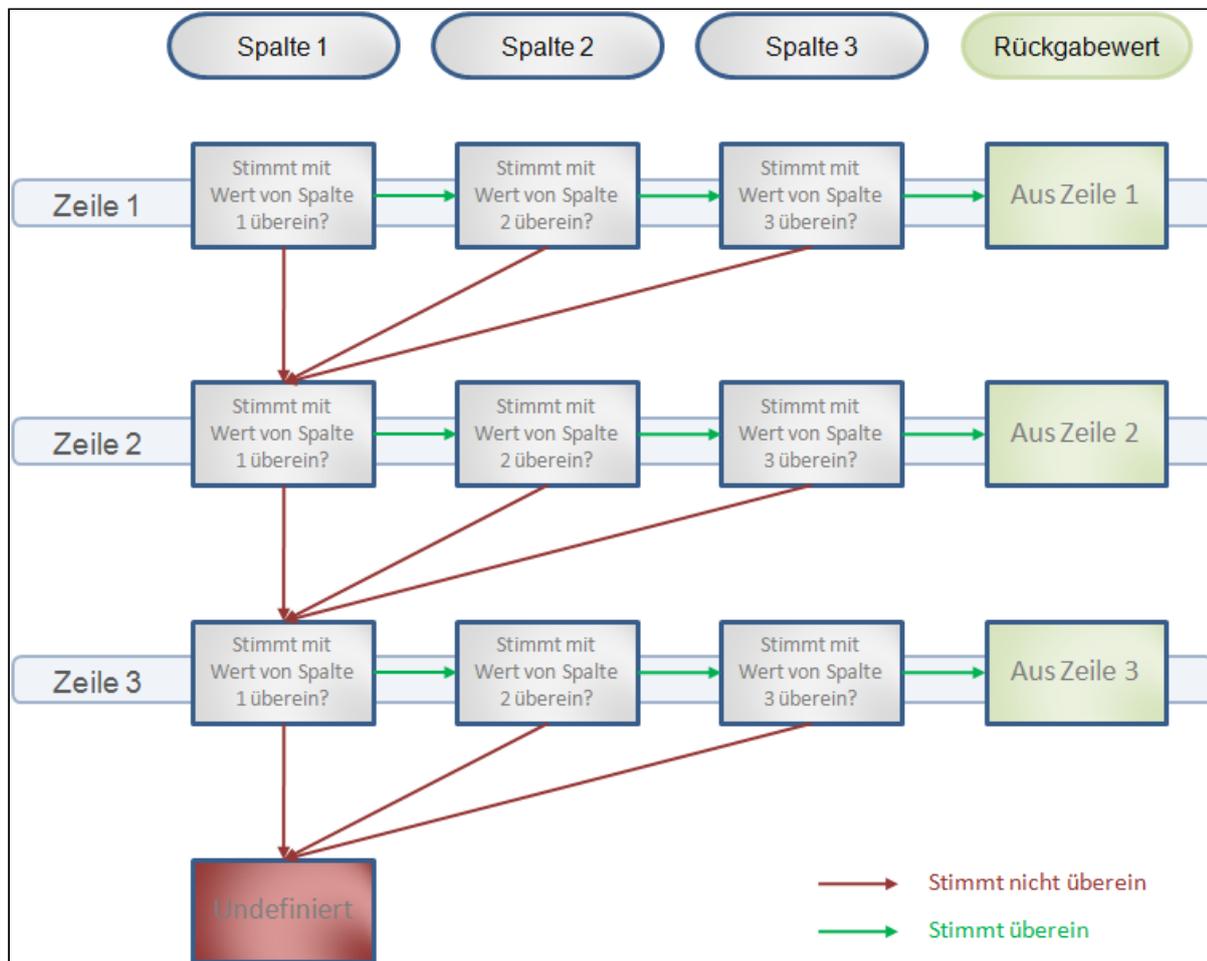


Abbildung D

Die Regel in Abb. D hat 3 Spalten und 3 Zeilen. Jede Spalte definiert eine Bedingung, jede Zeile einen Wert pro Spalte. Die Prüfung beginnt in Zeile 1, Spalte 1. Ist der Wert von Zeile 1 in Spalte 1 identisch zum Spaltenwert, wird mit der Prüfung von Spalte 2 in Zeile 1 fortgefahren. Ist der Wert von Zeile 1 in Spalte 2 wieder identisch zum Spaltenwert der 2. Spalte, wird mit der Prüfung von Spalte 3 in Zeile 1 fortgefahren. Ist auch der Wert von Spalte 3 in Zeile 1 identisch zum Spaltenwert, trifft die gesamte Zeile zu und der Rückgabewert der Zeile ist der Wert der Regel (grüne Pfeile). Hat eine der Spalten in Zeile 1 einen anderen Wert als die zugehörige Spaltenbedingung zu, trifft also nicht auf die Bedingung zu, wird die Prüfung der Zeile abgebrochen und mit der Prüfung der nächsten Zeile fortgefahren. Dort wird dann wieder in der 1. Spalte begonnen. Wurden alle Zeilen geprüft und keine davon erfüllt alle Spaltenbedingungen, ist der Wert der Regel undefiniert, da kein Rückgabewert ausgewählt werden kann (rote Pfeile).

Um zu verhindern, dass der Wert einer Regel undefiniert (leer) ist, falls keine der Zeilen alle Bedingungen erfüllt, können Sie unter den regulären Zeilen eine weitere Zeile einfügen, die in allen Spalten den Operator \* als Wert hat, also immer gültig ist. In dieser Zeile definieren Sie dann als Rückgabewert einen Standardwert. Trifft keine der regulären Auswahlmöglichkeiten zu, wird diese Standardzeile ausgewählt und der Wert der Regel ist der Standardwert. Dieses Vorgehen wurde in Abb. A durch Zeile #4 verwirklicht.

## Boolsche Operatoren

Das R-S definiert einige boolsche Operatoren, mit denen Sie Vergleiche anstellen können. Auch wenn Sie explizit keinen Operator angeben wird ein Operator verwendet, nämlich der Operator EQUAL : Die Bedingung ist wahr, wenn **Wert A = Wert B** , also wenn die Spalte in der Zeile den gleichen Wert hat wie die Spalte. Bisher sind wir in Regeln von diesem Verhalten ausgegangen: Der Wert in der Zeile muss identisch zum Wert der entsprechenden Spalte sein. Dies ist aber nur das Standardverhalten, das Sie durch boolsche Operatoren verändern können. In der Praxis werden Sie oft mit Operatoren arbeiten, nämlich immer dann, wenn Werte nicht nur gleich sein sollen, sondern in einer bestimmten Beziehung zueinander stehen sollen. Folgende Operatoren existieren im R-S:

### \* Operator

Den ALWAYSTRUE Operator \* haben Sie bereits im obigen Abschnitt kennengelernt. Er erwartet keine zu vergleichenden Argumente und hat immer den Wert TRUE , ist also immer wahr. Er wird oft benutzt, wenn eine Bedingung für die Zeile irrelevant ist, um einen Restbereich abzudecken, der durch die anderen Zeilen nicht abgedeckt wurde oder um ein alternatives Standardverhalten zu implementieren. Durch \* trifft diese Bedingung dann immer zu.

#	Weihnachten	Rückgabewert
	\${DS.todays_date} NUMBER	
1	24.12.2013	Heute ist Weihnachten!
2	*	Heute ist irgendein Tag, aber nicht Weihnachten!

#### Beispielregel A

Die Bedingung in *Beispielregel A* ist **24.12.2013** . Zeile #1 erhält das heutige Datum über die Datenstrom-Variable DS.TODAYS\_DATE . Ist das heutige Datum der 24.12.2013, so ist der Wert der Regel der STRING "Heute ist Weihnachten!". Ist heute nicht der 24.12.2013, trifft die Bedingung nicht zu und die Zeile ist ungültig. Zeile #2 ist immer gültig, da der Operator \* unabhängig von der Bedingung immer zutrifft (*Bsp. gilt natürlich nur für 2013*) . An allen anderen Tagen ist der Wert der Regel also der STRING "Heute ist irgendein Tag, aber nicht Weihnachten!". In Zeile #1 wird übrigens automatisch der EQUAL -Operator verwendet, da ja kein Operator angegeben wird. Die interne Darstellung ist also =24.12.2013 (der EQUAL -Operator darf aber **nicht** explizit manuell angegeben werden).

## != Operator

Der NOTEQUAL Operator != ist die Verneinung des EQUAL-Operators. Er wird benutzt, wenn 2 Werte nicht identisch sein sollen. Man könnte *Beispielregel A* also auch so implementieren:

#	Weihnachten \${DS.todays_date} NUMBER	Rückgabewert
1	!=24.12.2013	Heute ist nicht Weihnachten!
2	*	Heute ist Weihnachten!

### Beispielregel B

Wieder ist die Bedingung der Spalte Weihnachten das heutige Datum, abgefragt durch die Datenstrom-Variable DS.TODAYS\_DATE . Hat die Spalte den Wert **24.12.2013** , ist heute Weihnachten und Zeile #1 trifft zu ( DS.TODAYS\_DATE = 24.12.2013). Ist heute ein anderer Tag als der 24.12.2013, ist Zeile #2 gültig: DS.TODAYS\_DATE ist ungleich 24.12.2013 (Bsp. gilt natürlich nur für 2013) .

## < und > Operator

Die LESS < und GREATER > Operatoren überprüfen, ob ein Wert kleiner oder größer ist als der gegebene Wert. Der LESS Operator ist wahr, wenn der Wert kleiner ist als der gegebene Wert. Der GREATER Operator ist wahr, wenn der Wert größer ist als der gegebene Wert.

#	Price \${DS.Article_Price} NUMBER	Rückgabewert
1	>1000	\${Werbung_Auto}
2	1000	\${Werbung_Auto}
3	<1000	\${Werbung_Fahrrad}

### Beispielregel C

Die Bedingung der Spalte ist der Preis eines Artikels, abgefragt über die Datenstrom-Variable DS.ARTICLE\_PRICE . Ist DS.ARTICLE\_PRICE größer als 1000 ist Zeile #1 gültig, ist **DS.ARTICLE\_PRICE** exakt 1000 ist Zeile #2 gültig und wenn **DS.ARTICLE\_PRICE** kleiner als 1000 ist, ist Zeile #3 gültig.

## <= und >= Operator

Die EQUALLESS <= und EQUALGREATER >= Operatoren sind Erweiterungen der LESS und GREATER Operatoren. Der EQUALLESS Operator ist wahr, wenn der Wert kleiner oder gleich dem gegebenen Wert ist. Der EQUALGREATER Operator ist wahr, wenn der Wert größer oder gleich dem gegebenen Wert ist. *Beispielregel C* ließe sich also mithilfe des EQUALGREATER-Operators einfach implementieren:

#	Price \${DS.Article_Price} NUMBER	Rückgabewert		
1	>=1000	\${Werbung_Auto}		
2	<1000	\${Werbung_Fahrrad}		

### Beispielregel D

Wenn der Wert von DS.ARTICLE\_PRICE größer oder gleich 1000 ist, trifft Zeile #1 zu, ansonsten Zeile #2. Die Verwendung von EQUALLESS verläuft analog.

## [W1, W2, Wn] Operator

Mit dem INLIST Operator [W1, W2, Wn] können Sie beliebig viele Werte angeben, die geprüft werden sollen. Die einzelnen Werte werden durch ein Komma getrennt. Trifft einer der Werte zu, ist die Bedingung erfüllt.

#	Herkunftsstadt \${DS.CUST_CITY} STRING	Rückgabewert		
1	[Hamburg,Hannover,Bremen,Kiel]	\${Werbung_Region_NORD}		
2	[Berlin,Dresden,Leipzig,Aachen]	\${Werbung_Region_Ost}		

### Beispielregel E

Die Bedingung der Spalte ist die Herkunftsstadt des Kunden, abgefragt über die Datenstrom-Variable DS.CUST\_CITY. In jeder Zeile werden per INLIST -Operator 4 mögliche Werte angegeben. Trifft einer der Werte zu, ist die Zeile gültig.

## [W1, ..., W2] Operator

Mit dem INRANGE Operator [W1, ..., W2] kann ein Wertebereich angegeben werden. Liegt der Wert im Wertebereich, trifft die Bedingung zu.

#	Price \${DS.PRICE} NUMBER	Rückgabewert
1	[0, ..., 1000]	\${Werbung_Region_VW}
2	[1001, ..., 9999]	\${Werbung_Region_Audi}
3	>=10000	\${Werbung_Region_Porsche}

### Beispielregel F

Die Bedingung der Spalte ist der Rechnungspreis des Kunden, abgefragt über die Datenstrom-Variable DS.PRICE. In jeder Zeile wird ein Wertebereich angegeben. Die Preisspanne, in der sich der Rechnungsbetrag befindet, stellt die darauf abgestimmte Werbung (in einer Ressource).

Klicken Sie einen Operatortyp an, um seine Beschreibung zu sehen.

## Manipulation durch Funktionen

Natürlich können Sie in Regeln auch Funktionen in EVAL-Anweisungen benutzen. Dies ist oft sinnvoll, um Daten aufzubereiten und zu manipulieren. Nachfolgend werden einige Beispiele gezeigt, wie Sie Funktionen sinnvoll zur Datenmanipulation einsetzen können. Für einen Überblick über alle Funktionen, beachten Sie bitte den Artikel über [Funktionen in EVAL-Anweisungen](#).

### Beispiel 1: ToLowerS() / ToUpperS()

STRINGS werden im R-S *case-sensitive* behandelt, es wird also zwischen Groß- und Kleinschreibung unterschieden: "HalloWelt" und "hallowelt" sind verschiedene Strings. Wenn Sie eine Datenstrom-Variable oder andere Ressource, die einen String enthält, in einer Regel verwenden, müssen Sie Groß- und Kleinschreibung beachten:

#	Kunde MaxMustermann STRING	Rückgabewert
1	maxmustermann	0
2	*	1

### Beispielregel G

Der Rückgabewert der Regel ist aufgrund der Unterscheidung zwischen Groß- und Kleinschreibung 1. Da die exakte Schreibweise des Strings oft nicht bekannt ist, können Sie die Funktionen **ToLowerS** / **ToUpperS** benutzen: ToLowerS konvertiert alle Zeichen des Strings in Kleinbuchstaben, ToUpperS alle Zeichen in Großbuchstaben:

#	Kunde	Rückgabewert
	<code>\$(ToLowerS("MaxMustermann"))</code> STRING	
1	maxmustermann	0
2	*	1

#### Beispielregel H

Der Rückgabewert der *Beispielregel H* ist jetzt 0. Strings müssen in " " eingeschlossen werden. Sie müssen die Groß-/Kleinschreibung jetzt nicht mehr beachten, sondern nur sicherstellen, dass alle Strings ausschließlich Klein- oder Großbuchstaben enthalten. Externe Strings in Ressourcen müssen Sie deshalb auch durch die Funktion konvertieren lassen:

#	Kunde	Rückgabewert
	<code>\$(ToLowerS(DS.CUST_NAME))</code> STRING	
1	<code>\$(ToLowerS(KundeA))</code>	0
2	<code>\$(ToLowerS(KundeB))</code>	1
3	<code>\$(ToLowerS(KundeC))</code>	2

#### Beispielregel I

In *Beispielregel I* sind die Namen der Kunden in Variablen enthalten, die auch konvertiert werden müssen, wenn Sie nicht schon explizit kleingeschrieben sind.

#### Beispiel 2: IndexS()

Wenn Sie sich nur dafür interessieren, ob ein String einen anderen String enthält, können Sie die Funktion **IndexS** verwenden. Diese sucht nach einem Substring im String und gibt die Position zurück. Enthält der String den gesuchten String nicht, ist der Rückgabewert -1. Angenommen, im Transaktions-Formular ist die Datenstrom-Variable DS.PAYMENT\_TYPE definiert, mit folgender Syntax:

BANK\_STADT\_ZAHLUNGSART\_optional:KREDITKARTENANBIETER

wobei ZAHLUNGSART: **KREDITKARTE** sein kann.

DEUTSCHE BANK\_FRANKFURT\_KREDITKARTE\_VISA

Wir erstellen eine Regel die lediglich überprüfen soll, ob per Kreditkarte gezahlt wird:

#	Kreditkarte \${IndexS(DS.PAYMENT_TYPE,"KREDITKARTE")} STRING	Rückgabewert
1	-1	Die Zahlungsart ist nicht Kreditkarte
2	*	Die Zahlungsart ist nicht Kreditkarte

Beispielregel J

IndexS eignet sich hier deshalb, weil man nicht weiß, an welcher Stelle im String das Wort "KREDITKARTE" steht und weil es unmöglich wäre, für jede mögliche Bank, vor allem für jede mögliche Stadt eine eigene Zeile mit dem kompletten String zu definieren.

### Beispiel 3: SubtDate()

Mithilfe von **SubtDate** können Sie eine Anzahl von Tagen von einem Datum abziehen lassen (*Analog dazu: AddDate()*). Im folgenden Beispiel wird die Funktion dazu verwendet, zu berechnen, wie lange eine Person schon Kunde ist:

#	Datum \${SubDate(DS.Date,"90","YES")} DATE	Rückgabewert
1	>=\${DS.CUST_SINCE}	Als treuer Kunde bekommen Sie 20% Rabatt
2	<\${DS.CUST_SINCE}	Als neuer Kunde bekommen Sie 15% Rabatt

Beispielregel K

Mit SubtDate werden vom heutigen Datum DS.DATE 90 Tage abgezogen. Ist das neue Datum größer als der Wert in DS.CUST\_SINCE, ist die Person seit mehr als 90 Tagen Kunde: Zeile #1 trifft zu.

In diesen 3 Beispielen haben Sie gesehen, wie man Funktionen in Regeln verwenden kann. Weitere Funktionen finden Sie im Abschnitt über **Funktionen in EVAL-Anweisungen**.

## Physikalische Beileger

Physikalische Beileger sind Beileger, die bereits vorgedruckt und fertig im Druckzentrum vorliegen, in dem die Dokumente gedruckt werden. Dort werden die Beileger dann den Dokumenten beigelegt. Physikalische Beileger werden also nicht mit dem R-S erstellt und gestaltet, sondern im R-S wird nur definiert, ob und welche physikalischen Beileger ein Formular bekommen soll. Einen solchen Beileger binden Sie mit Hilfe der Ressource **physikalischer Beileger** an ein Formular. Im Gegensatz zu logischen Beilegern, die direkt im R-S erstellt und gestaltet werden und dann mit dem Formular mit gedruckt werden, müssen physikalische Beileger extern produziert und dann in der Kuvertierstraße vorhanden sein. Dies bedeutet einen zusätzlichen logistischen Aufwand und erschwert eine flexible Gestaltung der Dokumente. Außerdem ist die Anzahl an unterschiedlichen physikalischen Beilegern, die für einen Dokumentenstrom in der Kuvertieranlage bereitgehalten werden können, stark begrenzt. Wird der Beileger nicht mehr genutzt, können die bereits gedruckten Beileger nicht mehr verwendet werden und es fällt überflüssige Makulatur an. Physikalische Beileger sollten deshalb sparsam benutzt werden, hauptsächlich nur dann, wenn der Beileger zwingend extern gedruckt werden muss, eine spezielle Beschaffenheit aufweisen muss oder für spezielle Beileger, die nicht mit Hilfe von logischen Beilegern erzeugt werden könnten. Ansonsten empfehlen wir erst zu prüfen, ob die Anforderungen an einen Beileger nicht auch durch einen logischen Beileger erfüllt werden können. Bei Beilegern müssen Sie grundsätzlich ein **Material** angeben, welches die physischen Eigenschaften des Beilegers definiert. Das Symbol für physikalische Beileger im R-S ist .

### Physikalische Beileger im R-S Arbeitsablauf

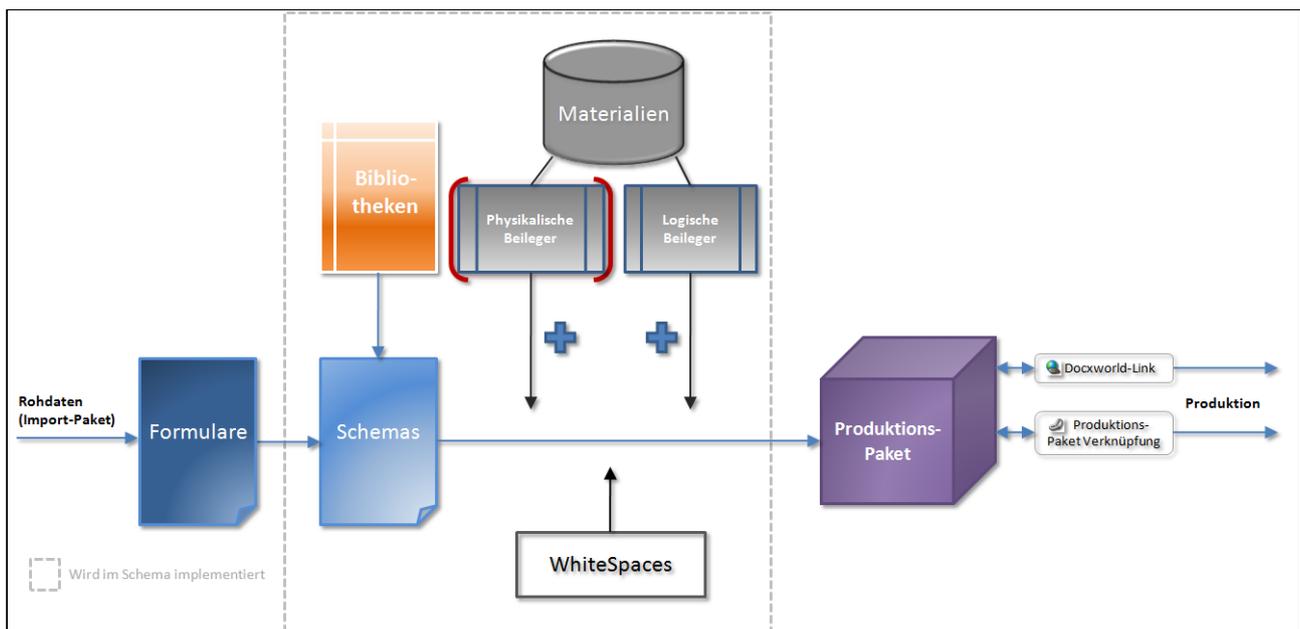


Abbildung A

Physikalische Beileger werden im Schema als Ressourcen erstellt. Physikalischen (und auch logischen) Beilegern muss immer ein Material zugewiesen werden, das die physischen Eigenschaften (Größe, Gewicht, etc.) des Beilegers definiert. Dies ist wichtig für den Produktions- und Versandprozess. Da physikalische Beileger immer auf formularfremdem Material gedruckt wurden (das Material des Beilegers ist dem R-S nicht bekannt) müssen Sie, bevor Sie einen physikalischen Beileger erstellen, immer zuerst ein Material definieren und dieses dann dem Beileger zuweisen. Was Materialien sind und wie sie erstellt werden lernen Sie in späteren Kapiteln. Natürlich können Sie einen physikalischen Beileger auch in einer Bibliotheks-Version

erstellen und diese dann in die Schema-Version einbinden. Da physikalische Beileger nicht im R-S gestaltet werden, dienen physikalische Beileger im R-S als Deklaration, die Dokumenten (Schemas) einen externen Beileger zuweisen. Deshalb enthält ein physikalischer Beileger im R-S nur eine Materialdefinition, die seine physischen Eigenschaften beschreibt. Logische Beileger hingegen enthalten auch das komplette Layout und Design des Beilegers, auf dessen Grundlage der Beileger dann mit den Dokumenten gedruckt wird.

### **Vorschau**

Da physikalische Beileger nicht im R-S gestaltet werden, ist dem R-S auch nicht bekannt, wie der Beileger aussieht. Bei der Erstellung einer Dokumentenvorschau für ein Schema werden deshalb auch keine physikalischen Beileger erzeugt (Im Gegensatz zu logischen Beilegern, die bei der Druckvorschau mit erzeugt werden). Es ist dennoch möglich zu überprüfen, ob die physikalischen Beileger korrekt beigelegt werden, da in den PDF-Dateien der Druckvorschau des R-S ein Inhaltsverzeichnis erzeugt wird, das alle zugewiesenen physikalischen Beileger enthält. Das Inhaltsverzeichnis lässt sich durch das PDF-Anzeigeprogramm darstellen. Weitere Informationen dazu finden Sie [hier](#).

### **Steuerung physikalischer Beileger durch Regeln**

Anstatt einen physikalischen Beileger direkt durch eine Erweiterung an das Dokument anzufügen ist es auch möglich, Regeln zu definieren, die physikalische Beileger abhängig von Bedingungen anfügen. Dazu geben Sie in der Erweiterung für physikalische Beileger keinen physikalischen Beileger direkt an, sondern eine Regel. Die Regel muss einen physikalischen Beileger zurückgeben. So ist es z.B. möglich, abhängig von Datenstrom-Variablen Beileger jedem Dokument individuell zuzuweisen.

### **Eine Bedingung für den Beileger**

Neben der Möglichkeit, durch Regeln Beileger gezielt zu steuern, können Sie in der Erweiterung für den physikalischen Beileger auch eine Bedingung definieren (Wie bei logischen Beilegern und WhiteSpaces). Die Bedingung muss einen booleschen Wert zurück liefern, also wahr / true (1) oder falsch / false (0). Einige Funktionen in EVAL-Anweisungen liefern z.B. boolesche Werte. Sinnvolle Operationen sind z.B. die Evaluierung von Datenstrom-Variablen, um Beileger nur für die Empfänger hinzuzufügen, für die der Beileger gedacht ist / sich eignet. Ist das Ergebnis der Bedingung wahr, wird der Beileger zum Dokument hinzugefügt, ist es falsch, wird er weggelassen.

### **Optionale Beileger**

Beileger können außerdem als optional gekennzeichnet werden. Für optionale Beileger gibt es keine Verarbeitungs-Garantie, da diese typischerweise einen werblichen Charakter besitzen. Dies bedeutet, dass optionale Beileger nur dann den Dokumenten beigelegt werden, wenn es produktionstechnisch sinnvoll ist. Ist ein Beileger nicht als optional gekennzeichnet, wird er auf jeden Fall beigelegt (Falls die dafür definierte Bedingung nicht falsch ist).

**Beschränkungen des Transaktions-Formulars**

Ob physikalische Beileger überhaupt erlaubt sind, wird durch das **Transaktions-Formular** vorgegeben (Transaktions-Formulare lernen Sie später kennen, sie enthalten die Rohdokumente). Werden physikalische Beileger durch das Transaktions-Formular verboten, können Sie den Dokumenten keine physikalischen Beileger hinzufügen.

Log. Beileger verwenden:	Log. Beileger werden verwendet in:
 <b>Material</b>	 <b>Schemas</b>

## Logische Beileger

Logische Beileger sind Beileger, die während des Produktionsprozesses direkt mit den eigentlichen Dokumenten mit gedruckt und dann mit ihnen verarbeitet und kuvertiert werden. Im Gegensatz zu physikalischen Belegern liegen logische Beileger also nicht schon vorgefertigt und kuvertierbereit vor, sondern werden zusammen mit den Dokumenten erstellt. Dies ermöglicht eine viel flexiblere Steuerung der Produktion, da Beileger erst dann erstellt werden, wenn sie auch wirklich gebraucht werden. Wechselt der Beileger, kann der neue Beileger sofort mit produziert werden und es fällt kein Überschuss an wie bei physikalischen, vorgefertigten Belegern. Einen solchen Beileger definieren Sie mit der Ressource **logischer Beileger**. Im Gegensatz zu physikalischen Belegern sind logische Beileger wesentlich flexibler und können viel einfacher individualisiert werden: Sie können die Belegern komplett im R-S entwerfen und gestalten, Ressourcen dynamisch benutzen und **Datenstrom-Variablen** verwenden, die dann nicht nur für die Dokumente, sondern auch die beigefügten Beileger angefragt werden. Physikalische Beileger hingegen haben, da Sie ja schon vorgefertigt sind, keinen Bezug zum Datenstrom und den restlichen Dokumenten. Bei Belegern (phys./log.) müssen Sie grundsätzlich ein **Material** angeben, welches die physischen Eigenschaften des Belegers definiert. Logische Beileger werden standardmäßig auf demselben Material / Papier gedruckt wie die übrigen Dokumente. Es aber auch möglich, logische Beileger auf anderem Material drucken zu lassen (Docxworld Premium). Das Symbol für logische Beileger im R-S ist .

### Logische Beileger im R-S Arbeitsablauf

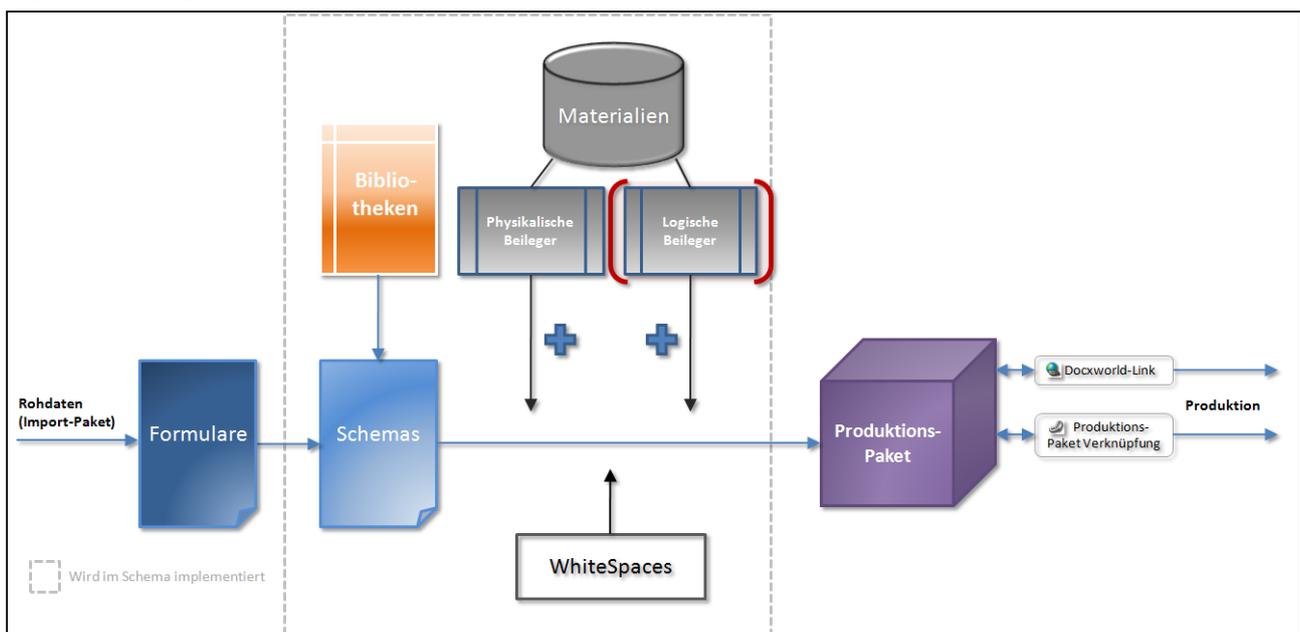


Abbildung A

Logische Beileger werden im Schema als Ressourcen erstellt. Logischen (und physikalischen) Belegern muss immer ein Material zugewiesen werden, das die physischen Eigenschaften (Größe, Gewicht, etc.) des Belegers definiert. Dies ist wichtig für den Produktions- und Versandprozess. Wird der logische Beileger auf demselben Material gedruckt wie die Dokumente, sind die Eigenschaften zwar schon bekannt, ein Material muss allerdings trotzdem angegeben werden. In diesem Fall sind allerdings nur die durch das Transaktions-Formular vorgegebenen Materialien zur Auswahl verfügbar. Wird der Beileger auf fremdem Material gedruckt, müssen dessen Eigenschaften im Material definiert sein, welches der Beileger verwendet. Was Materialien sind und wie sie erstellt werden lernen Sie in späteren Kapiteln. Das Design des Belegers erfolgt ebenfalls im

Schema mit Hilfe des **Design-Editors**. Natürlich können Sie logische Beileger auch in einer Bibliothek erstellen und designen und diese dann in eine Schema-Version einbinden.

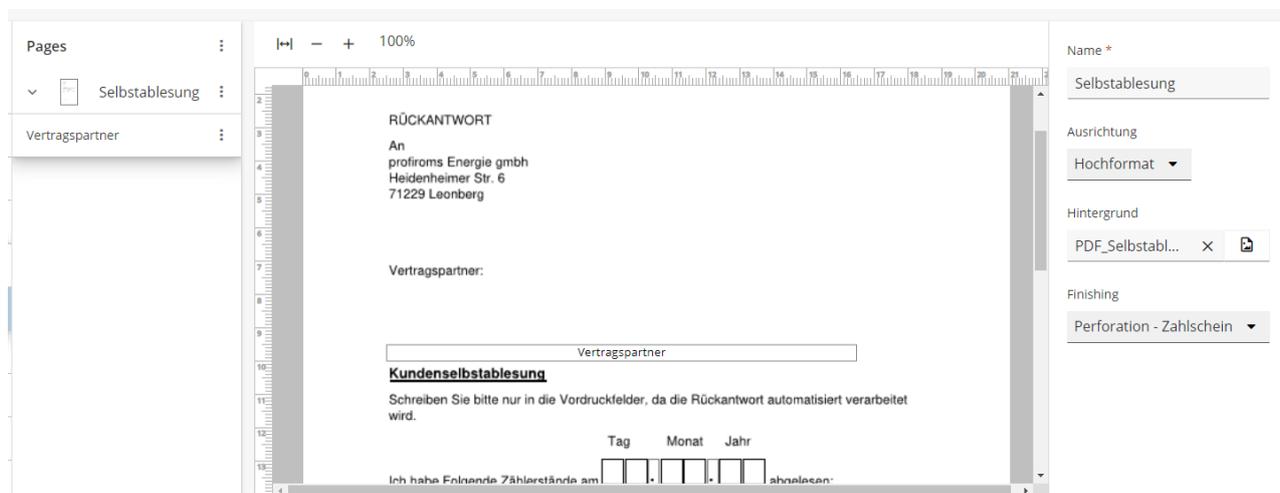


Abbildung B

Wie Sie mit Hilfe des Design-Editors logische Beileger gestalten können lernen Sie [hier](#).

### **Vorschau**

Da logische Beileger im R-S definiert, gestaltet und auch mit den Dokumenten in die Produktion geschickt werden, werden sie bei der Testvorschau-Generierung auch mit erstellt. Dazu müssen ihre Bedingungen anhand der Testdaten erfüllt (wahr) werden.

### **Steuerung logischer Beileger durch Regeln**

Anstatt einen logischen Beileger direkt durch eine Erweiterung an das Dokument anzufügen ist es auch möglich, Regeln zu definieren, die logische Beileger abhängig von Bedingungen anfügen. Dazu geben Sie in der Erweiterung für logische Beileger keinen logischen Beileger direkt an, sondern eine Regel. Die Regel muss einen logischen Beileger zurückgeben. So ist es z.B. möglich, abhängig von Datenstrom-Variablen Beileger jedem Dokument individuell zuzuweisen.

### **Eine Bedingung für den Beileger**

Neben der Möglichkeit, durch Regeln Beileger gezielt zu steuern, können Sie in der Erweiterung für den logischen Beileger auch eine Bedingung definieren (Wie bei physikalischen Beilegern und WhiteSpaces). Die Bedingung muss einen booleschen Wert zurück liefern, also wahr / true (1) oder falsch / false (0). Einige Funktionen in EVAL-Anweisungen liefern z.B. boolesche Werte. Sinnvolle Operationen sind z.B. die Evaluierung von Datenstrom-Variablen, um Beileger nur für die Empfänger hinzuzufügen, für die der Beileger gedacht ist / sich eignet. Ist das Ergebnis der Bedingung wahr, wird der Beileger zum Formular hinzugefügt, ist es falsch, wird er weggelassen.

**Optionale Beileger**

Beileger können außerdem als optional gekennzeichnet werden. Für optionale Beileger gibt es keine Verarbeitungs-Garantie, da diese typischerweise einen werblichen Charakter besitzen. Dies bedeutet, dass optionale Beileger nur dann den Dokumenten beigelegt werden, wenn es produktionstechnisch sinnvoll ist. So ist es möglich im Transaktions-Formular zusätzlich noch festgelegt werden, ob durch optionale Beileger ein Portosprung erfolgen darf oder nicht.

Ist ein Beileger nicht als optional gekennzeichnet, wird er auf jeden Fall beigelegt (Falls die dafür definierte Bedingung nicht falsch ist).

**Beschränkungen des Transaktions-Formulars**

Ob logische Beileger überhaupt erlaubt sind, wird durch das *Transaktions-Formular* vorgegeben (Transaktions-Formulare lernen Sie später kennen, sie enthalten die Rohdokumente, die im R-S gestaltet werden und diverse Konfigurationen dazu). Werden logische Beileger durch das Transaktions-Formular verboten, können Sie den Dokumenten keine logischen Beileger hinzufügen. Außerdem bestimmt das Formular, ob logische Beileger auf anderem Material (Papier) gedruckt werden dürfen wie die übrigen Dokumente. Wenn logische Beileger nur auf dem vom Formular vorgegebenen Material gedruckt werden dürfen, können Sie kein abweichendes Material angeben.

Log. Beileger verwenden:	Log. Beileger werden verwendet in:
 <i>Material</i>	 <i>Schemas</i>

## WhiteSpaces

WhiteSpaces ermöglichen es, dynamisch freie Bereiche auf den Dokumenten mit Inhalt zu füllen. Sie sind die einzige Möglichkeit, Inhalt außerhalb der festen Layout-Bereiche des Transaktions-Formulars zu platzieren. WhiteSpaces werden allerdings nicht einem festen Bereich auf den Dokumenten zugeordnet, sondern für jedes Formular wird einzeln berechnet, ob und wie der WhiteSpace eingefügt werden soll. WhiteSpaces, die Sie für ein Formular definieren, werden also nicht auf jedem Formular platziert, sondern nur auf denen, auf denen genügend Platz vorhanden ist. Dies ermöglicht es Ihnen, freie Flächen automatisch auszunutzen zu lassen. Zu diesem Zweck werden in WhiteSpaces Eckdaten angegeben, wie z.B. die Mindestgröße des freien Bereichs oder ob der WhiteSpace vor oder nach dem eigentlichen Dokument-Inhalt platziert werden muss. Aus diesen Daten berechnet das R-S dann individuell, ob der WhiteSpace eingefügt werden kann. Hauptanwendungsgebiet von WhiteSpaces ist Werbung: Ist auf einem Dokument noch genügend freie Fläche übrig, platziert das R-S dort den WhiteSpace mit Werbung. Das Symbol für WhiteSpaces im R-S ist .

### WhiteSpaces im R-S Arbeitsablauf

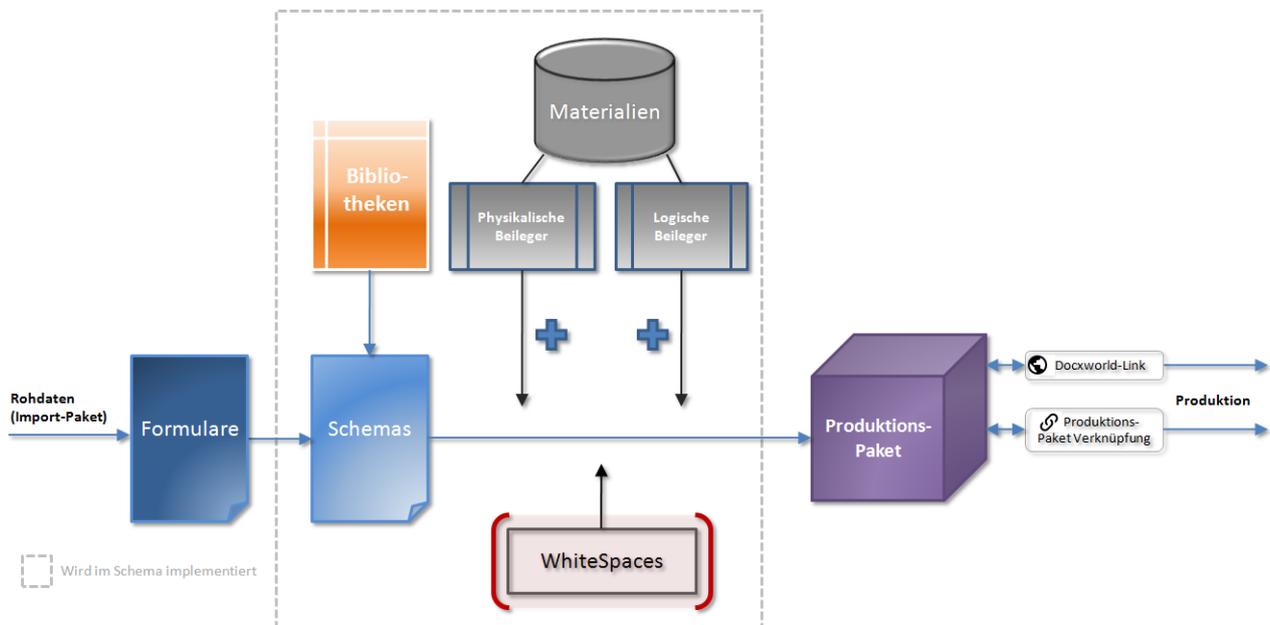


Abbildung A

WhiteSpaces werden durch WhiteSpace-Erweiterungen eingebunden. Neben der Möglichkeit, in WhiteSpace-Erweiterungen direkt einen WhiteSpace, den Sie im Dokument nutzen möchten, anzugeben, können Sie dort auch Variablen und Regeln angeben. Die Variable / Regel müssen dann allerdings einen WhiteSpace zurückgeben. So ist es möglich, durch Regeln WhiteSpaces bedingt einzubinden. Gibt die angegebene Variable / Regel einen anderen Wert zurück als einen WhiteSpace, so wird die WhiteSpace-Erweiterung inaktiv.

### Konfiguration des WhiteSpace

Auch wenn WhiteSpaces nicht manuell im Dokument eingefügt werden können gibt es diverse Einstellungsmöglichkeiten, mit denen Sie die Platzierung von WhiteSpaces bedingt steuern können. Folgende Einstellungen sind für WhiteSpaces verfügbar:

- + **Mindestgröße des freien Bereichs**
- + **Position auf dem Dokument**
- + **Position auf der Seite**
- + **Eine Bedingung für den WhiteSpace**
- + **Auf welchen Teilen des Formulars der WhiteSpace platziert werden darf**

### Mindestgröße des freien Bereichs

The image shows three examples of WhiteSpace configuration for a document named 'MyWhiteSpace'. Each example includes a configuration panel and a preview of the document with a red dashed box indicating the WhiteSpace area.

- Example 1:** Name: MyWhiteSpace, Breite (mm): 50, Höhe (mm): 20. The WhiteSpace is a small rectangle at the bottom of the document.
- Example 2:** Name: MyWhiteSpace, Breite (mm): 100, Höhe (mm): 40. The WhiteSpace is a larger rectangle at the bottom of the document.
- Example 3:** Name: MyWhiteSpace, Breite (mm): 250, Höhe (mm): 100. The WhiteSpace is a large rectangle at the bottom of the document, but it is not placed because the document's width is less than 250 mm.

Abbildung B

Sie können für den WhiteSpace eine Mindestgröße angeben. Diese Mindestgröße bestimmt, wie groß der freie Bereich auf dem Dokument mindestens sein muss, damit der WhiteSpace platziert wird. Der Inhalt des WhiteSpaces wird dann auch auf die angegebene Größe skaliert. Das Ergebnis sehen Sie in Abb. B: Mit einer Mindestgröße von 5x2 cm, 10x4 cm und 25x10 cm. Da der freie Bereich auf dem Dokument kleiner als 25x10 cm ist, wird der 3. WhiteSpace in Abb. C nicht auf dem Dokument platziert. Wie die Dimension des WhiteSpaces in der Praxis angegeben wird sehen Sie hier.

### Position auf dem Dokument

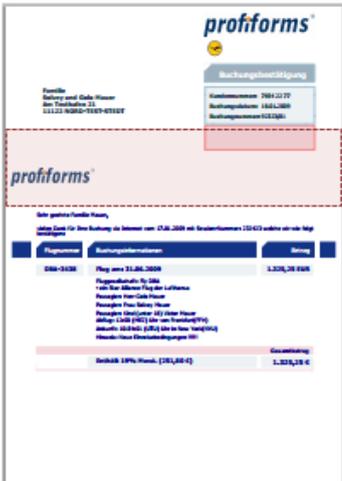
Sie können bestimmen, auf welchen Seiten des Dokuments der WhiteSpace aufgebracht werden darf. Folgende Optionen sind verfügbar:

Option	Beschreibung	Erlaubte Seiten
FIRST	Nur auf der ersten Seite des Dokuments.	   
LAST	Nur auf der letzten Seite des Dokuments	   
ANY	Auf einer beliebigen Seite des Dokuments	   

Ist auf keine der erlaubten Seite genug freie Fläche zur Platzierung des WhiteSpaces, wird er weggelassen. Wenn Sie **ANY** auswählen, wird der WhiteSpace auf der ersten geeigneten Seite eingefügt. WhiteSpaces werden pro Formular nur 1 mal eingefügt. Wie die Position des WhiteSpaces auf dem Dokument in der Praxis angegeben wird sehen Sie [hier](#).

### Position auf der Seite

Analog dazu müssen Sie auch festlegen, wo auf der Seite der WhiteSpace aufgebracht werden darf. Folgende Optionen sind verfügbar:

Option	Beschreibung	Beispiel
TOP	Vor dem Inhalt der Seite (oben)	

Option	Beschreibung	Beispiel
BOTTOM	Nach dem Inhalt der Seite (unten)	

Wie die Position des WhiteSpaces auf der Seite in der Praxis angegeben wird sehen Sie [hier](#).

### **Eine Bedingung für den WhiteSpace**

Optional können Sie dem WhiteSpace eine Bedingung geben. Diese Bedingung muss einen booleschen Wert zurückgeben (wahr oder falsch, 1 oder 0). Möglich sind z.B. **EVAL-Anweisungen**. Der WhiteSpace wird nur auf das Dokument angewendet, wenn die Bedingung wahr zurückgibt. Andernfalls wird der Inhalt des WhiteSpace auch dann nicht in das Dokument eingefügt, wenn genügend Platz vorhanden ist.

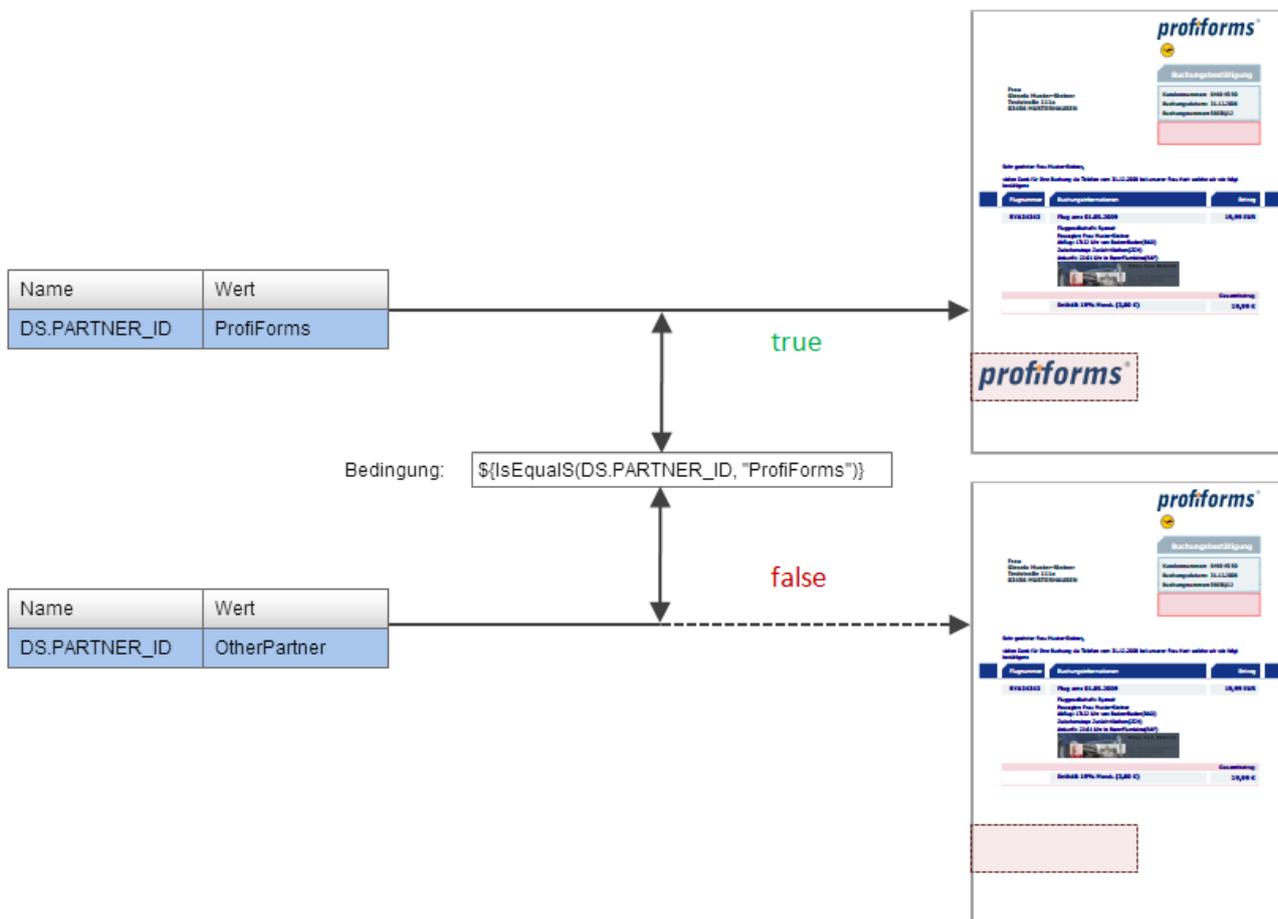


Abbildung C

In *Abb. C* ist exemplarisch die Verwendung eines WhiteSpaces mit Bedingung dargestellt. Das Dokument wird 2x generiert. Die dazwischen geschaltete Bedingung verlangt, dass die Variable DS.PARTNER\_ID den Wert **ProfiForms** hat. Dies ist beim 1. Durchlauf (oberer) der Fall, beim 2. Durchlauf nicht (Dort ist der Wert **OtherPartner**). Die Bedingung ist also nur beim 1. Durchlauf erfüllt. Dementsprechend wird der WhiteSpace auch nur beim 1. Durchlauf angewandt und in das Dokument eingefügt. Wie die Bedingung des WhiteSpaces in der Praxis angegeben wird sehen Sie [hier](#).

### **Auf welchen Teilen des Formulars der WhiteSpace platziert werden darf**

Das Transaktions-Formular kann einzelne Formular-Teile (Bereiche) vorgeben, auf denen WhiteSpace platziert werden darf. Damit ist es möglich, Teile des Dokumentes, auf denen kein WhiteSpace platziert werden soll, festzulegen (z.B. wichtige Unterlagen). Gibt das Transaktions-Formular keine Formular-Teile vor, darf WhiteSpace auf dem kompletten Dokument platziert werden. Es ist dann auch nicht möglich, einen WhiteSpace auf nur einen Teil des Dokuments anwenden zu lassen. Definiert das Transaktions-Formular Formular-Teile, müssen Sie die gewünschten Teile, auf die Sie den WhiteSpace anwenden möchten, angeben. Wenn Sie keine Formular-Teile auswählen, wird der WhiteSpace auf allen erlaubten Bereichen des Dokuments angewendet. Wie Formular-Teile in der Praxis ausgewählt werden sehen Sie [hier](#).

**Beschränkungen des Transaktions-Formulars**

Neben den Formular-Teilen werden durch das Transaktions-Formular noch weitere Einschränkungen definiert. So legt das Formular fest, ob WhiteSpaces überhaupt in einem Dokument verwendet werden dürfen. Ebenso gibt das Transaktions-Formular eine Obergrenze für die Nutzung von Werbe-Fläche an. Liegt die Obergrenze für Werbung beispielsweise bei 50%, dürfen nicht mehr als 50% der freien Fläche auf den Dokumenten durch WhiteSpaces eingenommen werden. Außerdem bestimmt das Formular, ob WhiteSpaces einen Seitenüberlauf verursachen dürfen, also ob durch das Einfügen eines WhiteSpaces in das Dokument der Inhalt so verschoben werden darf, dass eine neue Seite für den verschobenen Inhalt nötig wird. Es wird allerdings nie eine neue Seite begonnen, auf der sich ausschließlich ein WhiteSpace befindet, sondern immer nur, wenn auch Inhalt auf die Seite verschoben wird. Wenn die Option aktiviert ist, wird auch festgelegt, ob die neue Seite auch die Rückseite der Dokument-Seite sein darf oder ob nur eine neue Seite. Das Transaktions-Formular kann solche Verschiebungen allerdings auch komplett verbieten, falls dadurch ein Portosprung entstehen würde. **Hier** wird erklärt, wo Sie diese Einstellungen einsehen können.

WhiteSpaces verwenden:	WhiteSpaces werden verwendet in:
<a href="#">Ressourcen</a>	 <a href="#">Schema</a>

## Datenstrom-Variablen

Datenstrom-Variablen bilden eine Schnittstelle zwischen ERP-/CRM-/HOST-Systemen, also den datenverarbeitenden Systemen, die kundenspezifische Informationen enthalten und diese in Form eines Datenstroms für jedes Formular ausgeben, und dem R-S. Dieser Datenstrom, der zu jedem Formular die benötigten Kundeninformationen enthält, kann durch Datenstrom-Variablen abgefangen und genutzt werden. Eine Datenstrom-Variable ist also eine Ressource, die ein Datenfeld eines an den Produktionsprozess angeschlossenen Datensystems repräsentiert und dieses Datenfeld in das R-S migriert. Im R-S kann die Datenstrom-Variable dann als Ressource benutzt werden. Der Wert der Datenstrom-Variable wird erst bei der endgültigen, externen Dokumentengenerierung eingesetzt (z.B. durch den ReportWriter). Der Wert der Datenstrom-Variable steht also im R-S noch nicht fest. Das Symbol für Datenstrom-Variablen im R-S ist  .

### Deklaration

Datenstrom-Variablen haben immer die Form:

**DS.NAME\_DER\_VARIABLE**

Nur Datenstrom-Variablen dürfen im R-S mit **DS.** beginnen.

 Datenstrom-Variablen können im R-S weder erstellt noch bearbeitet, gelöscht, kopiert, importiert, exportiert oder umbenannt werden, da Sie externe Ressourcen repräsentieren.

### Inhalt und Werte von Datenstrom-Variablen

Nicht alle Datenfelder, die im zugrunde liegenden ERP-/CRM-/HOST-System existieren, müssen auch im R-S als Datenstrom-Variablen verfügbar sein. Welche Datenstrom-Variablen im R-S verfügbar sind, wird durch den Formular-Designer bestimmt und ist im Import-Paket ( **Transaktions-Formular** ) festgelegt. Dort können Sie in der **Registerkarte Datenstrom-Variablen** einsehen (gibt es auch bei Schema-Versionen), welche Datenstrom-Variablen ein Formular bereitstellt. Datenstrom-Variablen müssen auch nicht zwingend denselben Namen haben wie das Datenfeld im Datensystem, das Sie repräsentieren. Zur Vereinfachung kann der Formular-Designer die Namen der Datenstrom-Variablen selbst festlegen. Welchen Wert eine Datenstrom-Variable annehmen wird, ist zum Zeitpunkt der Bearbeitung im R-S unbekannt, da dies ja für jedes Formular individuell ist (Jeder Kunde hat einen anderen Namen, eine andere Kundennummer, einen anderen Rechnungsbetrag, nicht jeder kommt aus Deutschland, usw.). Um im R-S mit Datenstrom-Variablen arbeiten zu können, muss jedoch bekannt sein, welchen Inhalt eine Datenstrom-Variable repräsentiert. Dazu enthält optimalerweise jede Datenstrom-Variable eine Beschreibung und eine Angabe möglicher Werte, um dem R-S Nutzer mitzuteilen, welche Werte eine Datenstrom-Variable enthalten kann (oder welche Werte, die eine Datenstrom-Variable annehmen kann, relevant sind). Diese Angaben werden durch den Formular-Designer gemacht und sind in der Registerkarte Datenstrom-Variablen einzusehen:

>	<b>DS.PARTNERID</b> Partner ID
▼	<b>DS.PAYMENT_TYPE</b> ZAHLUNGSART
	BANK                      Banküberweisung
	MASTER                      Einzug erfolgt über MASTER-Card
	VISA                              Einzug erfolgt über VISA-Card
>	<b>DS.gender</b> Geschlecht des Kunden

 Die angegebenen Werte müssen nicht zwingend alle möglichen Werte repräsentieren. Es werden nur die Werte gezeigt, die der Formular-Designer angegeben hat.

Datenstrom-Variablen enthalten immer einen STRING (Text). Wie normale Variablen können aber auch Datenstrom-Variablen alle Datentypen enthalten, die das R-S kennt. Mehr über Datentypen finden Sie [hier](#). Wenn Sie also z.B. in Regeln Bedingungen definieren, die von einer Datenstrom-Variablen abhängen, müssen Sie sicher sein, dass die Datenstrom-Variablen auch den Datentyp enthält, den Sie voraussetzen. Geben Sie bspw. in einer Regel der Spalte als Bedingung eine Datenstrom-Variablen:

**DS.ORDER\_DATE**

und als Datentyp der Spalte DATE an, müssen Sie sicherstellen, dass der Inhalt von DS.ORDER\_DATE auch in jedem Fall mit dem R-S Datentyp DATE konform ist, da die Bedingung sonst nicht ordnungsgemäß ausgeführt werden kann.

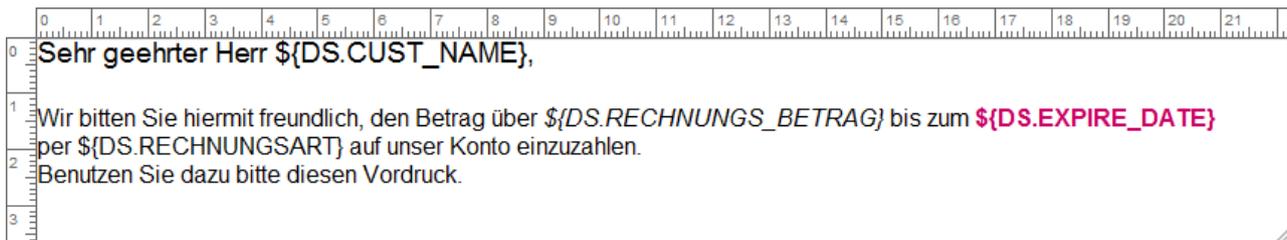
### **Nutzung von Datenstrom-Variablen**

Datenstrom-Variablen können wie jede andere R-S Ressource verwendet werden: Sie können entweder direkt per Erweiterung in einen Layout-Bereich des Formular eingefügt werden (Erweiterungen lernen Sie später kennen) oder in einer anderen Ressource mit Hilfe dynamischer Einbindung. Dynamische Einbindung mit Hilfe einer **EVAL-Anweisung** haben Sie bereits in den vorherigen Ressource-Beschreibungen kennengelernt: Sie können in einer Ressource eine andere Ressource einfügen (auf sie verweisen), indem Sie folgenden Term benutzen: `#{Name_der_Ressource}`. Die Verwendung in anderen Ressourcen ist für Datenstrom-Variablen in der Regel sinnvoller, da Datenstrom-Variablen einfache Werte enthalten, die normalerweise nicht unformatiert in ein Dokument eingefügt werden sollen. Oft dienen Datenstrom-Variablen auch als Entscheidungs-Variablen. Die häufigsten Verwendungsarten von Datenstrom-Variablen sind:

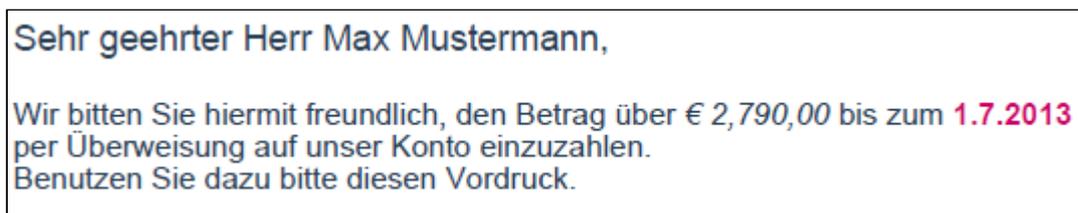
## 1. In Variablen oder RichTexten

Datenstrom-Variablen werden genutzt, um Texte im Dokument zu personalisieren und kundenspezifische Werte in den Text einzufügen. Sie dienen dann im Text als "Platzhalter" für ihren Wert. Für formatierte und größere Texte sollten **RichTexte** benutzt werden. Im RichText können Sie den Wert der Datenstrom-Variablen mitformatieren, indem Sie die Formatierung einfach auf die gesamte EVAL-Anweisung anwenden (bei Variablen ebenfalls mit Hilfe von **INLINE-Sequenzen** möglich). Fügen Sie die Datenstrom-Variablen einfach per `${Name_der_Datenstromvariable}` ein:

### Der RichText:



### und seine Anzeige im Dokument (mit Beispielwerten):



## 2. In Regeln

In Regeln werden hauptsächlich Datenstrom-Variablen für Spaltenbedingungen benutzt, um ein von der Datenstrom-Variable abhängiges Verhalten zu implementieren. Dazu müssen die relevanten Werte der Datenstrom-Variable bekannt sein. Da Datenstrom-Variablen alle verfügbaren formular-spezifischen Informationen enthalten sind sie die Möglichkeit, Formulare im R-S zu personalisieren. Wird als Spaltenbedingung einer Regel eine Datenstrom-Variable angegeben, hängt der Rückgabewert der Regel von dem Wert der Datenstrom-Variable ab. So lassen sich bedingte Ausführungen zur Laufzeit implementieren. In Regeln verwenden Sie Datenstrom-Variablen wie üblich per EVAL-Anweisung. So lassen sich eine Fülle verschiedener Anforderungen implementieren, z.B.:

- **Steuerung von Ressourcen:**

Die Regel gibt auf Basis einer Datenstrom-Variable die richtige Ressource zurück, z.B. einen RichText oder eine Binäre Ressource. Beispiele: Ein Text, der in mehreren Sprachen mehrfach in verschiedenen RichTexten vorliegt. Abhängig von einer Datenstrom-Variable, die die Nationalität des Kunden angibt, wird die richtige Ressource ausgewählt und zurückgegeben.

Oder: Sie haben mehrere Werbeanzeigen in binären Ressourcen und möchten die passendste für den Kunden aussuchen. Dazu implementieren Sie eine Regel, die auf Basis verschiedener Datenstrom-Variablen (z.B. Alter, Rechnungshöhe, Wohnort, etc.) die passendste Ressource aussucht.

- **Steuerung von Beilegern:**

Sie haben logische / physikalische Beileger und möchten den passendsten an das Formular anhängen. Dazu implementieren Sie eine Regel, die auf Basis von Datenstrom-Variablen den richtigen logischen Beileger zurückgibt. Oder: Die Regel (oder auch die Bedingung, siehe [hier](#)) entscheidet, ob überhaupt ein Beileger angefügt werden soll. Soll kein Beileger angehängt werden, gibt die Regel nichts zurück und die Erweiterung, die die Regel verwendet, hat keinen Beileger zum Anhängen.

- **Logische Berechnungen, Funktionen:**

Mit Hilfe von **boolschen Operatoren**, EVAL-Anweisungen und **Funktionen** können Sie Berechnungen, Vergleiche und bedingte Ausdrücke definieren. So lassen sich die Informationen aus dem Datensystem vielfältig für Bedingungen nutzen.

Es gibt unzählige Anforderungen, die sich durch Datenstrom-Variablen in Regeln verwirklichen lassen. In der **Regel-Definition** finden Sie weitere Beispiele und lernen, wie man mit Regeln ordnungsgemäß arbeitet. Bevor Sie mit Regeln arbeiten, sollten Sie unbedingt diesen Abschnitt lesen. Außerdem hilfreich sind die Artikel über die **Umgebung des R-S** sowie der Abschnitt über **EVAL-Anweisungen, Operatoren und Datentypen**.

## Basisobjekte im R-S

Sie haben im vorigen Kapitel Ressourcen kennengelernt, die genutzt werden, um Daten und Informationen zu speichern und diese in das Dokument einzubringen. Die zweite Gruppe von Objekten im R-S sind die Basisobjekte.

Basisobjekte sind die Elemente im R-S, mit deren Hilfe der Dokument-Erstellungsprozess modelliert wird. Die Haupt-Basisobjekte sind Transaktions-Formulare, Schemas und Produktions-Pakete. Einige Basisobjekte unterliegen der Versionierung, d.h. ein Basisobjekt kann gleichzeitig in mehreren Versionen vorliegen. Diese Versionen haben einen Lebens-Zyklus, der durch ihren Status bestimmt wird. Lesen Sie dazu bitte den Abschnitt über den **R-S Workflow**, falls Sie das noch nicht getan haben. Die einzelnen Basisobjekte gehen untereinander Beziehungen ein, so können z.B. Bibliotheken in Schemas verwendet werden, Schemas binden sich an ein Transaktions-Formular und Produktions-Pakete bilden aus Formular und Schema ein produktionsbereites Paket.

In diesem Kapitel wird die theoretische Grundlage der Basisobjekte beschrieben und ihre Funktion im R-S erklärt. Die praktische Nutzung der Objekte wird in den Kapiteln "Arbeiten mit..." beschrieben und vertieft.

### Insgesamt gibt es sieben R-S Objekte:

#### Die drei Hauptobjekte für den Dokument-Erstellungs-Prozess sind:

-  **Transaktions-Formulare:** Sie beinhalten die Rohdokumente, die im R-S gestaltet werden sollen.
-  **Schemas:** Schemas gestalten die Rohdokumente des Transaktions-Formulars mit Hilfe von Ressourcen.
-  **Produktions-Pakete:** Enthält den Inhalt eines Schemas und seines Trans.-Formulars in konsolidierter Form.

#### Objekte, um in Schemas Ressourcen und Material (für Beileger) bereitzustellen:

-  **Material:** Materialien beschreiben die Eigenschaften von Beilegern (den Dokumenten zugeführtes Material).
-  **Bibliotheken:** Bibliotheken enthalten Ressourcen, die wieder verwertbar in Schemas genutzt werden können.

**Objekte, um Produktions-Pakete für den weiteren Produktions-Prozess außerhalb des R-S verfügbar zu machen:**

 **Docxworld-Link:** Machen Produktions-Pakete extern durch eine Docxworld-Vertragsnummer verfügbar.

 **Produktions-Paket Verknüpfungen:** Machen Produktions-Pakete extern durch einen Namen verfügbar.

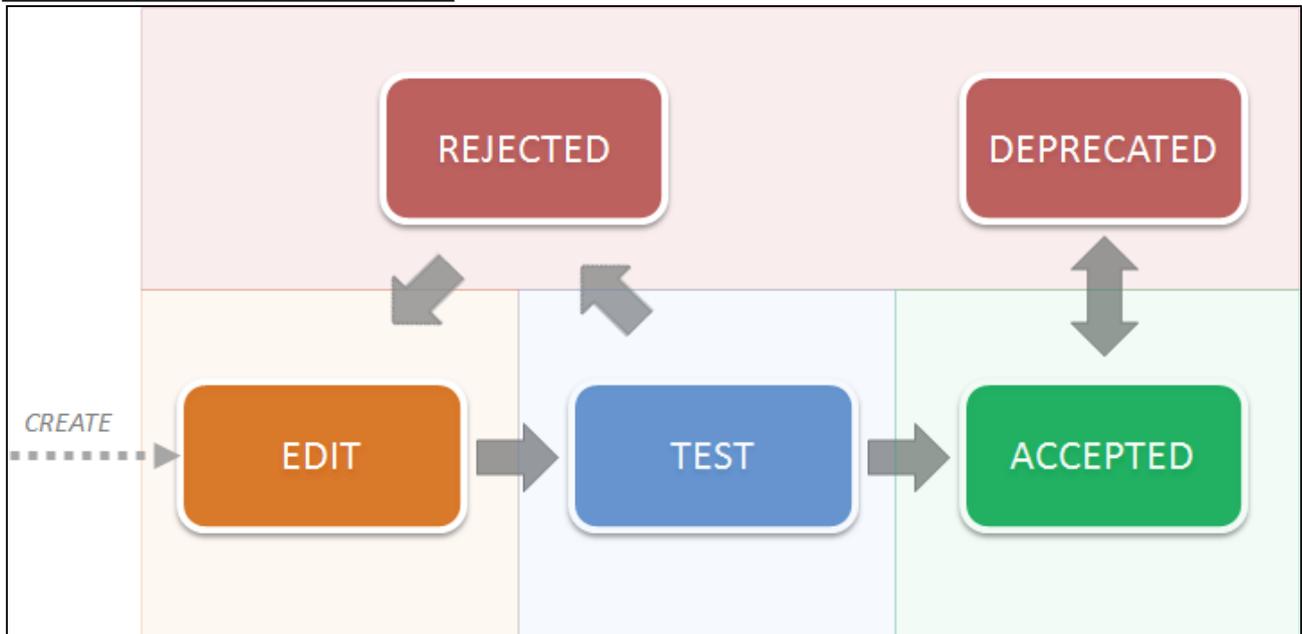


**Import-Pakete (und damit auch Transaktions-Formulare) enthalten u.a. Folgendes:**

- Die **Rohdokumente**, die im R-S gestaltet werden sollen.
- **Layout-Bereiche** in den Rohdokumenten (*Erweiterungs-Punkte*), die im R-S mit Inhalt gefüllt werden dürfen.
- Den **Namen** des Formulars.
- Den Namen und die benötigte Version des **Binär-Pakets** (*Lesen Sie dazu bitte [diesen Abschnitt](#)*).
- **Datenstrom-Variablen** aus dem Datenstrom oder TCI, die im R-S verwendet werden können.
  - Datenstrom-Variablen enthalten **Daten aus den ERP-/CRM-/HOST-Systemen**, z.B. SAP.
  - Sie enthalten optional eine **Beschreibung** der möglichen Werte der Datenstrom-Variable.
  - Datenstrom-Variablen enthalten Dokument-spezifische Informationen. Mehr darüber erfahren Sie [hier](#).
- Welche **Fonts** im Dokument verwendet werden dürfen.
- **Testdaten**, um eine Testvorschau der Dokumente generieren zu können.
- Ob **logische** und **physikalische Beileger** sowie **WhiteSpaces** erlaubt sind und wenn ja:
  - Ob logische Beileger auf **anderem Material** wie die eigentlichen Dokumente gedruckt werden dürfen,
  - Ob Beileger und WhiteSpaces einen **Portosprung** verursachen dürfen,
  - Auf welchen **Dokument-Teilen WhiteSpaces** erlaubt sind.

Die Vorgaben des Import-Pakets können im R-S nicht geändert werden. Sie bestimmen, wie im R-S mit den Rohdokumenten verfahren werden darf. Mehr über den internen Aufbau von Import-Paketen erfahren Sie [hier](#). Die Rohdokumente aus dem Transaktion-Formular werden im R-S mit Hilfe von Schemas gestaltet. Sie werden im nächsten Abschnitt vorgestellt.

Status eines Transaktions-Formulars



Für genauere Informationen über die Statuswechsel einer Formular-Version beachten Sie den Artikel über den [R-S Workflow](#).

Formulare verwenden:	Formulare werden verwendet in:
<i>Import-Paket</i>	★ <i>Schema</i>

## Schemas

Schemas ermöglichen die Gestaltung der Rohdokumente aus dem Transaktions-Formular. Sie sind die elementaren Objekte, um aus den Rohdaten des Transaktions-Formulars individuell angepasste Dokumente zu generieren. Schemas generieren **Produktions-Pakete**, die Schema und dazugehöriges Transaktions-Formular für die externe Produktion konsolidiert beinhalten. Sie sind der Versionierung unterworfen und haben deshalb einen Status und einen Lebens-Zyklus.

### Schemas im R-S Arbeitsablauf

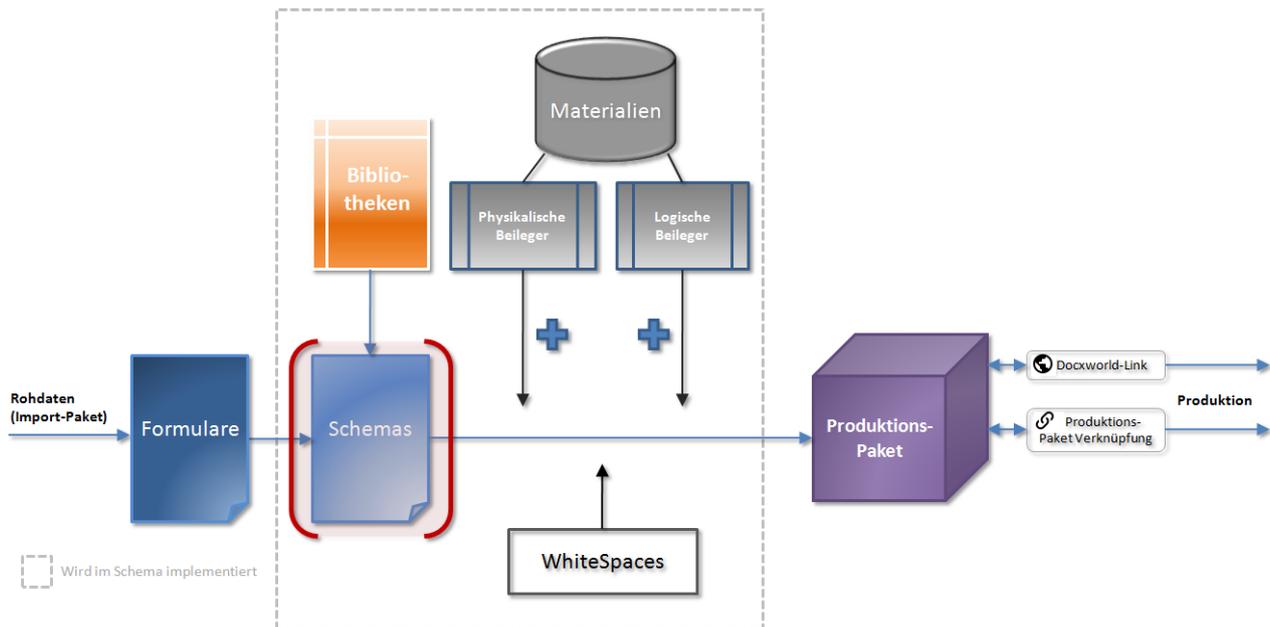


Abbildung A

Wie Sie in Abb. A sehen, werden alle gestalterischen Aspekte der Dokument-Bearbeitung im Schema implementiert. Ein Schema ist mit der Version eines **Transaktions-Formulars** verbunden, das die Rohdokumente und die dazugehörigen Daten enthält, die durch das Schema gestaltet werden sollen. Die Rohdokumente werden durch das Schema mit Ressourcen (Inhalt) gefüllt und die **Datenstrom-Variablen** aus dem Datenstrom verarbeitet. Außerdem werden im Schema Beleger an das Dokument angehängt. Mit Hilfe von WhiteSpaces lassen sich freie Bereiche im Dokument flexibel mit Inhalt (z.B. Werbung) füllen. Ob und wie Beleger und WhiteSpaces im Schema verwendet werden dürfen, legt das zu Grunde liegende Transaktions-Formular fest. Mehr zu diesen Objekten erfahren Sie in den nächsten Abschnitten.

## Inhalt eines Schemas

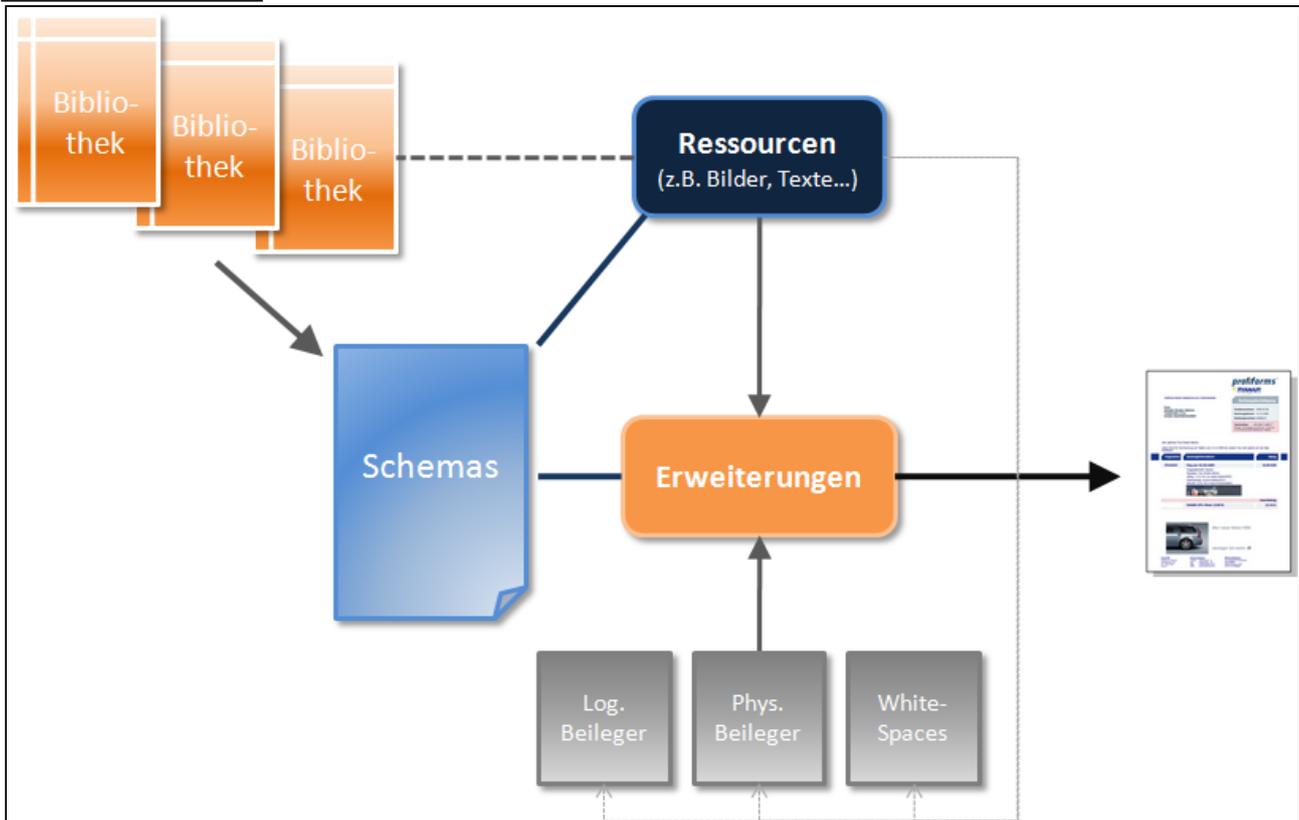


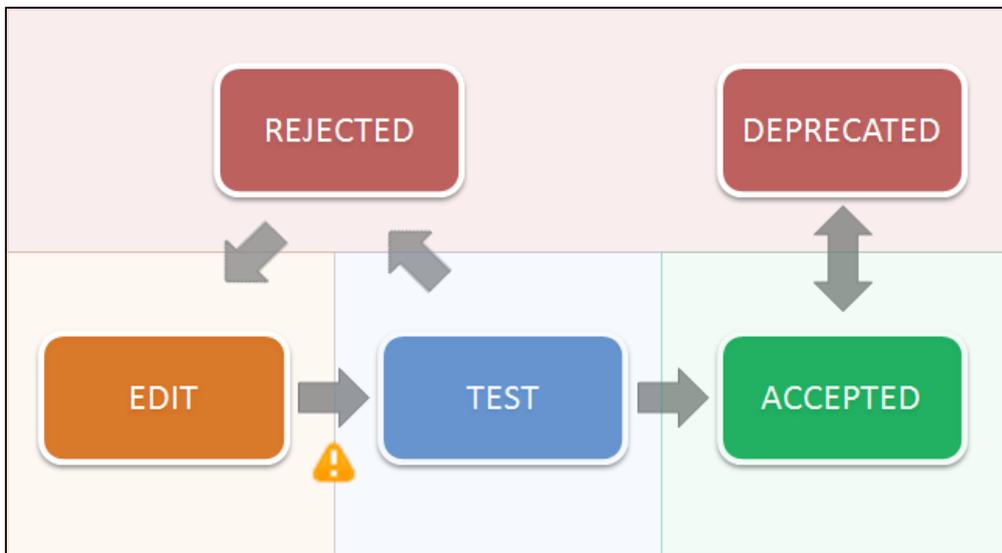
Abbildung B

## Elementarer Inhalt eines Schemas:

- **Ressourcen**
  - Die Ressourcen, die in ein Dokument eingefügt werden sollen, sind im Schema enthalten. Diese Ressourcen können entweder lokal im Schema definiert sein oder durch externe Bibliotheken eingebunden werden. Auch Beileger und WhiteSpaces sind Ressourcen. Global definierte **Materialien** dienen als physische Spezifikation für Beileger.
- **Bibliotheken**
  - Bibliotheken beinhalten Ressourcen, die Sie in ein Schema einbinden können. Die Nutzung von Bibliotheken bietet einige Vorteile, v.a. bei einer großen Anzahl von Schemas und Ressourcen.
- **Erweiterungen**
  - In das Dokument eingefügt werden Ressourcen, Beileger und WhiteSpaces durch Erweiterungen. Sie binden Ressourcen an die Erweiterungs-Punkte des Dokuments und fügen Beileger zu dem Dokument hinzu.
- **Vorschau**
  - Sie haben mit Hilfe von Testdaten die Möglichkeit, im Schema eine Vorschau zu generieren, die das gleiche Ergebnis der Gestaltung der Rohdokumente durch das Schema enthält, wie es später beim Druck der Fall ist. Die dazu verwendeten Testdaten stammen entweder aus dem Transaktions-Formular oder wurden manuell im Schema erstellt.

Schemas werden zusammen mit ihrem Transaktions-Formular zu Produktions-Paketen gebündelt, die Sie dann für den weiteren Produktions-Prozess außerhalb des R-S mit Docxworld-Links und Produktions-Paket Verknüpfungen verfügbar machen können. Wie genau man mit Schemas arbeitet, erfahren Sie im Abschnitt [Arbeiten mit Schemas](#).

**Statuswechsel einer Schema-Version**



Für genauere Informationen über die Statuswechsel einer Schema-Version beachten Sie den Artikel über den [R-S Workflow](#).

Schemas verwenden:	Schemas werden verwendet in:
<b>Transaktions-Formular</b>	<b>Produktions-Paketen</b>
<b>Bibliotheken</b>	<b>Docxworld-Vertragsverlinkungen</b>
<b>Materialien</b>	<b>Produktions-Paket Verknüpfungen</b>

## Produktions-Pakete

Produktions-Pakete stellen den letzten Abschnitt im internen Arbeitsablaufs des R-S dar. Produktions-Pakete werden durch Schema-Versionen generiert, wenn diese TEST-Status erreichen. Zu einem Produktions-Paket gehört also immer eine Schema-Version (und ein Schema). Von einer Schema-Version können mehrere Produktions-Pakete existieren, aber immer nur eines kann "aktuell" sein (PRODUCTION-Status). Dieses aktuelle Produktions-Paket kann dann durch Docxworld-Links oder Produktions-Paket Verknüpfungen, die darauf "verlinken", von externer Software erreicht und exportiert werden und damit in Produktion und Druck geschickt werden. Produktions-Pakete enthalten den Inhalt der Schema-Version (die Gestaltung der Rohdokumente durch Ressourcen, Beileger, etc.), durch die sie generiert wurden, den Inhalt der dazugehörigen Transaktions-Formular-Version (also den Inhalt des ursprünglichen Import-Pakets, siehe dazu auch den Abschnitt über die **Umgebung des R-S**) und weitere Konfigurationsdaten.

Produktions-Pakete beinhalten also das Resultat der Arbeit mit dem R-S und stellen damit das Ergebnis jeder Dokument-Bearbeitung durch das R-S dar. Sie sind die Objekte des R-S, die außerhalb genutzt werden können (für den weiteren externen Produktionsprozess) Aus dem Produktions-Paket können dann z.B. durch den ReportWriter fertige Dokumente erzeugt werden, die für den Druck geeignet sind. Produktions-Pakete sind der Versionierung unterworfen und haben deshalb einen Status und einen Lebens-Zyklus.

### Produktions-Pakete im R-S Arbeitsablauf

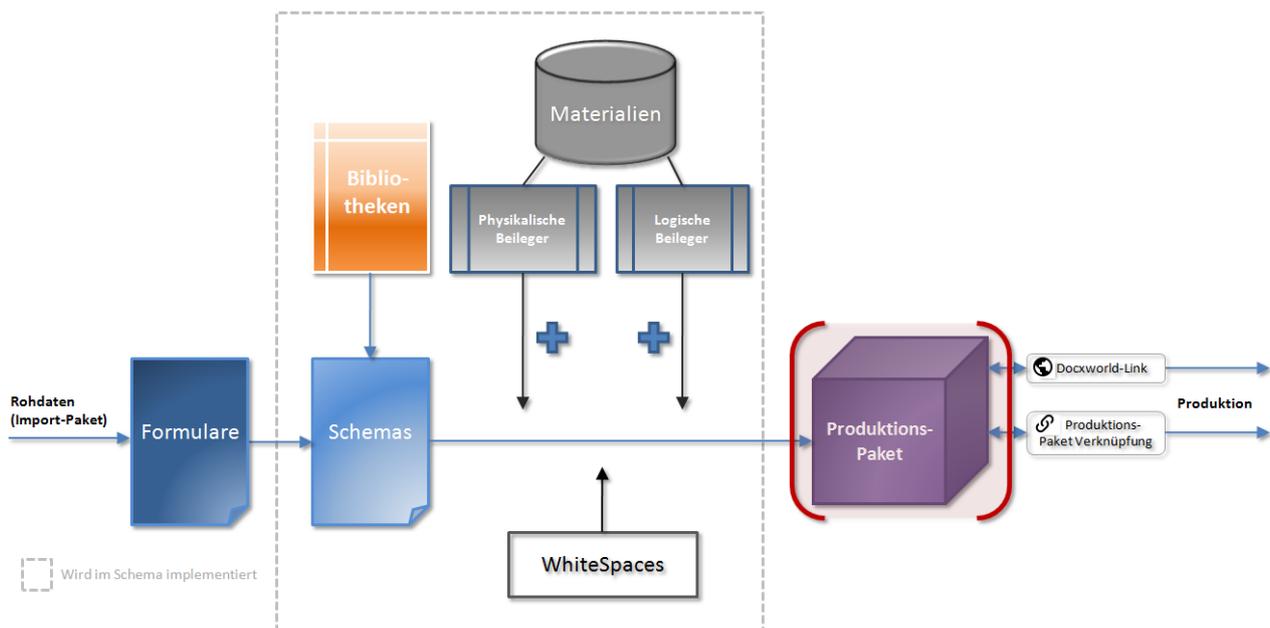


Abbildung A

### Export von Produktions-Paketen

Nach außen verfügbar gemacht werden Produktions-Pakete durch Zuordnungstabellen im R-S. Diese dienen als Schnittstellen zwischen dem R-S und bspw. (E)OMS-/docxworld-Systemen, die dadurch auf die Produktions-Pakete zugreifen können.

Dazu gibt es 2 Möglichkeiten:

## 1. Docxworld-Links

Docxworld-Links müssen genutzt werden, wenn Sie **docxworld** von profiforms benutzen. Druckaufträge sind dann immer einer eindeutigen Docxworld-Vertragsnummer zugeordnet, über die der Auftrag im Docxworld identifiziert ist. Diese Vertragsnummer muss in dem Docxworld-Link, der auf das Produktions-Paket verweist, angegeben werden. Externe Komponenten des Produktionsprozesses wie der ReportWriter können dadurch das Produktions-Paket finden und darauf zugreifen, indem sie aus dem Datenstrom des Druckauftrags die Vertragsnummer auslesen und vom R-S das zugewiesene Produktions-Paket erhalten. (Siehe dazu **Umgebung des R-S** und **Docxworld-Links**)

## 2. Produktions-Paket Verknüpfungen

Wenn Sie kein Docxworld verwenden, müssen Sie das Produktions-Paket durch eine Produktions-Paket Verknüpfung verfügbar machen. In einer Produktions-Paket Verknüpfung geben Sie einen eindeutigen Namen an, durch den das zugewiesene Produktions-Paket identifiziert werden kann. Dieser Name muss auch im Datenstrom (als Formularename) definiert sein. Durch die Zuordnung der Namen anstatt der Docxworld-Vertragsnummer können Produktions-Pakete auch so aus dem R-S exportiert werden (Siehe dazu **Umgebung des R-S** und **Produktions-Paket Verknüpfungen**).

Intern sind Produktions-Pakete ZIP-Dateien mit einer spezifischen Struktur. Nachfolgend der interne Aufbau eines Produktions-Pakets:

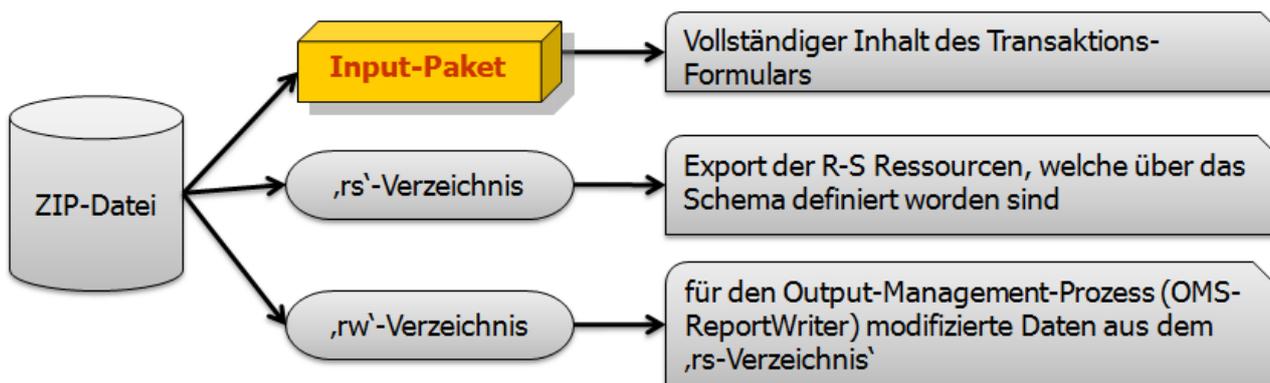
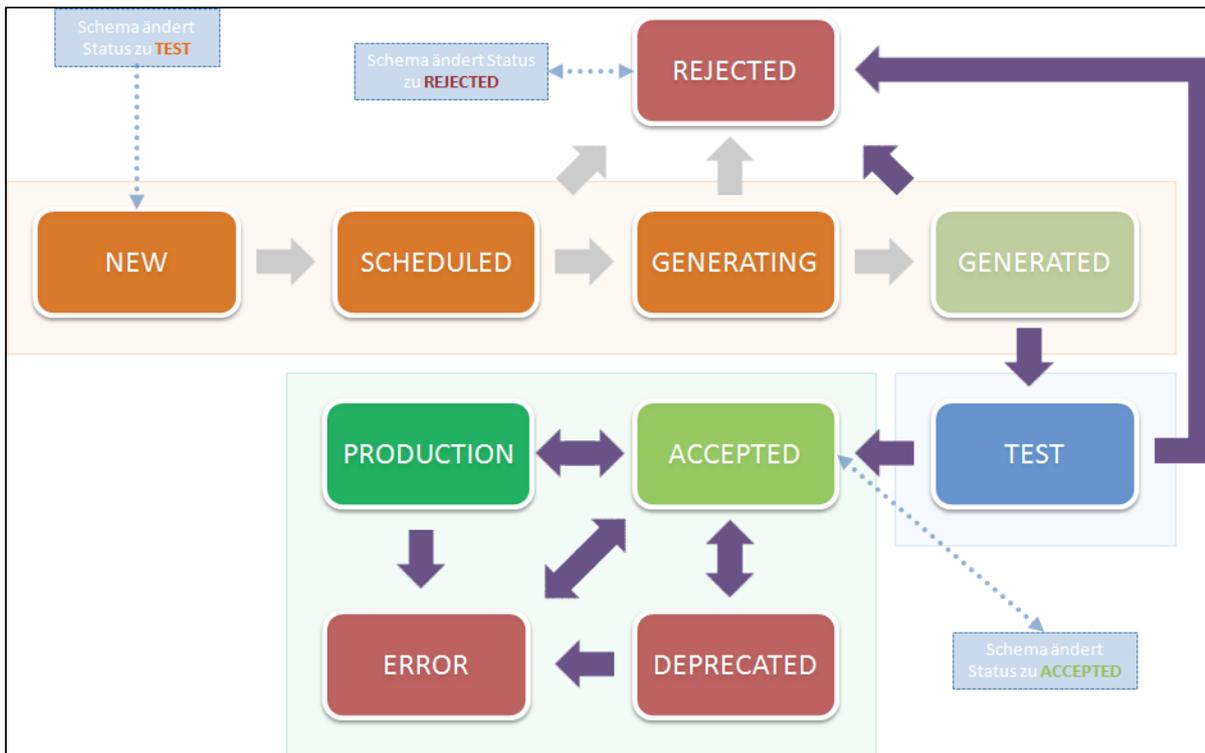


Abbildung B

Neben dem Input-Paket (ursprünglich Import-Paket des Transaktions-Formulars), das die Rohdaten und Dokumente aus dem Formular-Designer enthält, enthält das Produktions-Paket die Ressourcen aus dem Schema (die Gestaltung der Dokumente durch das R-S) und für den Output-Management-Prozess modifizierte Daten aus dem R-S. Im Input-Paket ist außerdem festgelegt, welche Versionen der Programme im Binär-Paket für die Dokument-Aufbereitung benutzt werden müssen. Das Binär-Paket ist die 2. Komponente neben dem Produktions-Paket, das für die Druckaufbereitung benötigt wird. Es wird immer zusammen mit dem Produktions-Paket exportiert und enthält Programme und Skripts, die aus dem Produktions-Paket Dokumente generieren (Mehr über Binär-Pakete erfahren Sie **hier**). Lesen Sie dazu den Abschnitt über die **Umgebung des R-S**.

Produktions-Pakete können den Status PRODUCTION annehmen. Erst wenn ein Paket diesen Status hat, ist es über Docxworld-Links oder Produktions-Paket Verknüpfungen für den externen Produktions-/Druckprozess verfügbar und exportierbar. Von Produktions-Paketen mit dem gleichen Schema als Grundlage (nicht von der gleichen Schema-Version!) kann immer nur 1 Paket PRODUCTION-Status haben. Eine zeitgesteuerte Produktivsetzung ist ebenfalls möglich. Wie genau man mit Produktions-Paketen arbeitet, erfahren Sie im Abschnitt **Arbeiten mit Produktions-Paketen**.

**Status eines Produktions-Pakets**



Für genauere Informationen über die Statuswechsel eines Produktions-Pakets beachten Sie bitte den Artikel über den **R-S Workflow**.

Produktions-Pakete verwenden:	Produktions-Pakete werden verwendet in:
☰ <b>Transaktions-Formular</b>	🌐 <b>Docxworld-Vertragsverlinkungen</b>
★ <b>Schema</b>	🔗 <b>Produktions-Paket Verknüpfungen</b>

## Bibliotheken

Eine Bibliothek ist eine Sammlung von Ressourcen, die Sie in eine Schema-Version einbinden können. Anstatt Ressourcen lokal in der Schema-Version anzulegen, können Sie benötigte Ressourcen in Bibliotheken ablegen und diese Bibliotheken dann in Ihre Schema-Version einbinden. Die Ressourcen der Bibliothek stehen dann im Schema zur Verfügung. Die Verwendung von Bibliotheken bietet einige Vorteile, unter anderem:

- + Besseren Überblick über Ressourcen durch Aufteilung in einzelne, thematisch geordnete Bibliotheken.
- + Wiederverwendung von Ressourcen, da diese nur einmal zentral in einer Bibliothek angelegt werden müssen.
- + Wird eine neue Ressource angelegt, steht sie sofort in allen Schema-Versionen, die die Bibliothek nutzen, zur Verfügung.
- + Ressourcen müssen nicht in jeder Schema-Version aktualisiert werden, sondern nur zentral in der Bibliothek.

### Bibliotheken im R-S Arbeitsablauf

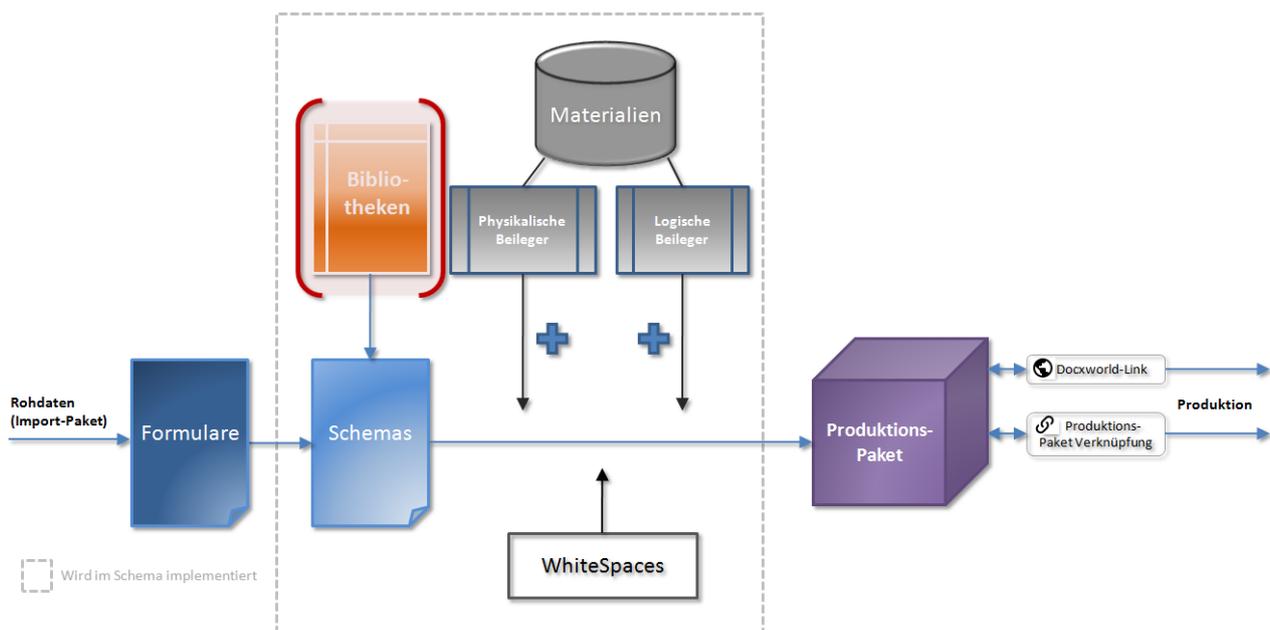


Abbildung A

In eine Schema-Version können beliebig viele Bibliotheken eingebunden werden. Es ist allerdings nicht möglich, eine Bibliothek in eine andere Bibliothek einzubinden. Bibliotheken können ihre Ressourcen also auch nicht an andere Bibliotheken vererben (an andere Bibliotheken weitergeben, die die ursprüngliche Bibliothek dann durch weitere Ressourcen ergänzen). Dennoch können in Bibliotheken Ressourcen aus anderen Bibliotheken genutzt werden, solange beide Bibliotheken später in der gleichen Schema-Version eingebunden werden (mehr dazu [hier](#)).

## Inhalt und Aufbau von Bibliotheken

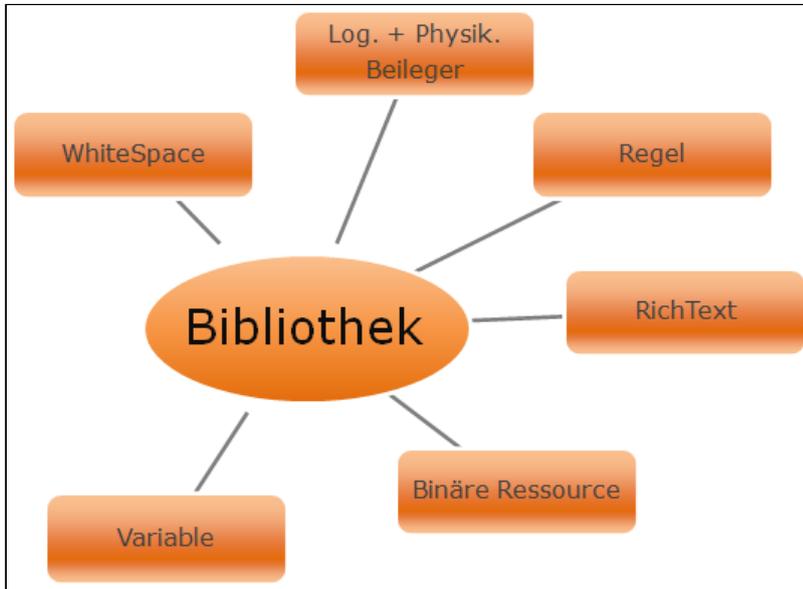


Abbildung B

Eine Bibliothek kann alle R-S Ressourcen enthalten. Bibliotheks-Ressourcen können im Schema normal verwendet werden. In Bibliotheken gibt es jedoch neben den sogenannten INHALTS-WERTEN (den in der Bibliothek definierten Ressourcen Abb. B) auch sogenannte ERFORDERLICHE RESSOURCEN:

### Erforderliche Ressourcen

Erforderliche Ressourcen sind in einer Bibliothek definierte, erwartete Ressourcen. Beim Erstellen einer Erforderliche Ressourcen geben Sie der Erforderliche Ressourcen einen Namen. Die Bibliothek erwartet dann bei Einbindung in eine Schema-Version eine Ressource mit diesem Namen. Diese Ressource kann aus der Formular-Version, die die Schema-Version benutzt (**Datenstrom-Variablen**), der Schema-Version oder einer anderen, in die Schema-Version eingebundenen, Bibliothek stammen. Erforderliche Ressourcen sind nötig, damit Sie in einer Bibliothek mit Ressourcen arbeiten können, die in der Bibliothek nicht existieren, von denen Sie aber wissen, dass Sie später im Schema verfügbar sein werden (z.B. Datenstrom-Variablen des Transaktions-Formulars). Da Bibliotheken unabhängig von Formularen und Schemas sind, weiß eine Bibliothek nicht, welche Ressourcen später im Schema zur Verfügung stehen werden. Erforderliche Ressourcen dienen also als Schnittstellen zwischen der Bibliothek und dem Schema / Formular.

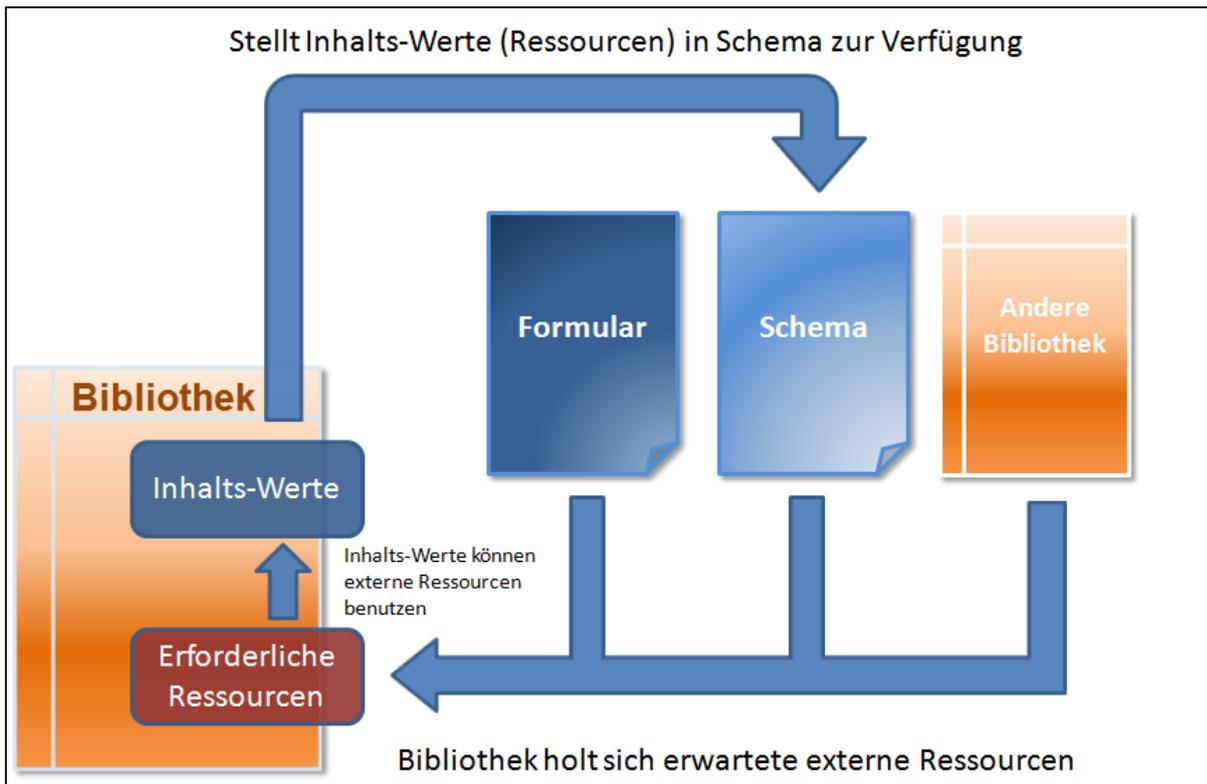


Abbildung C

Erforderliche Ressourcen können Sie dann innerhalb der Bibliothek wie ganz normale Ressource benutzen, z.B. in einer **EVAL-Anweisung**:

```
{Name_der_erforderlichen_Ressource}
```

Bei der späteren Einbindung in das Schema wird in den Ausdruck dann die Ressource eingesetzt, die in der Schema-Version vorhanden ist und den gleichen Namen hat. Erforderliche Ressourcen sind typunabhängig, das bedeutet der Typ der Ressource, die später die Erforderliche Ressourcen ausfüllt ist irrelevant und kann nicht vorgegeben werden. Sie sollten deshalb bereits bei der Erstellung der Erforderliche Ressourcen wissen, durch welche Ressource sie später ausgefüllt wird. Da der Ausdruck ungültig werden würde, wenn keine Ressource mit diesem Namen im Schema existiert, müssen alle Erforderliche Ressourcen in der Schema-Version vorhanden sein. Andernfalls verursacht die Bibliothek bei der Dokumenterstellung einen Fehler und bricht den Prozess ab.

Veranschaulichung:

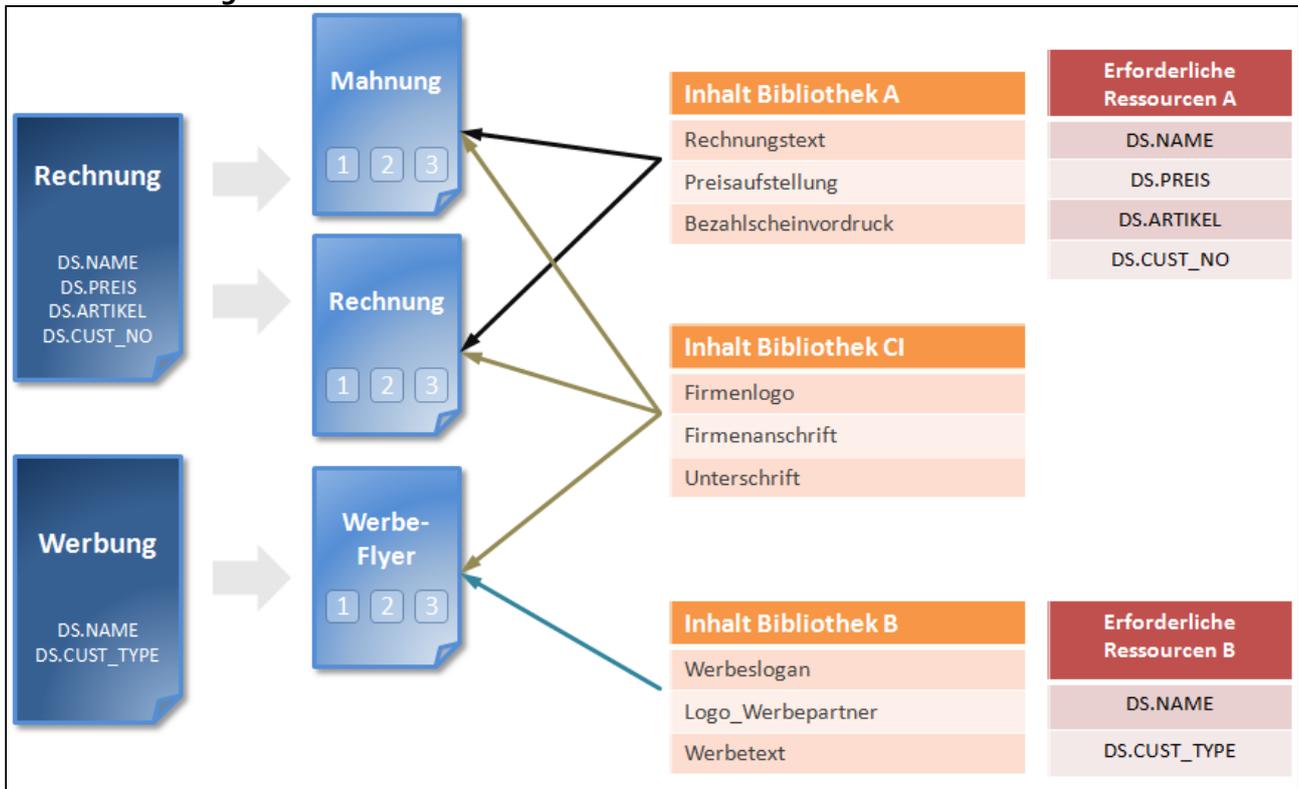


Abbildung D

In unserem Beispiel haben wir 2 Formulare (bzw. Formular-Versionen):

Das Formular **Werbung** und das Formular **Rechnung**. Die Datenstromvariablen des jeweiligen Formulars sind mit angegeben.

	In unserem Beispiel haben wir 2 Formulare (bzw. Formular-Versionen): Das Formular <b>Werbung</b> und das Formular <b>Rechnung</b> . Die Datenstromvariablen des jeweiligen Formulars sind mit angegeben.
	In unserem Beispiel haben wir 3 Schemas mit je 3 Versionen: Das Schema <b>Werbe-Flyer</b> , bei dem alle 3 Versionen mit dem Formular <b>Werbung</b> als Vorlage arbeiten sowie die Schemas <b>Mahnung</b> und <b>Rechnung</b> , bei denen alle 3 Versionen mit dem Formular <b>Rechnung</b> als Vorlage arbeiten.
	In unserem Beispiel haben wir 3 Bibliotheken: Die Bibliothek <b>CI</b> , die keine erforderlichen Ressourcen besitzt und die Bibliotheken <b>A</b> und <b>B</b> , deren erforderliche Ressourcen in den Tabellen daneben angegeben werden.

Alle Schema-Versionen enthalten nun durch die Einbindung von Bibliothek **CI** die Ressourcen Firmenlogo, Firmenanschrift und Unterschrift. Die Schema-Versionen des Schemas **Werbe-Flyer** enthalten zusätzlich die Ressourcen Logo\_Werbepartner, Werbeslogan und Werbetext aus Bibliothek **B**.

Die Schema-Versionen der Schemas **Mahnung** und **Rechnung** enthalten zusätzlich die Ressourcen Rechnungstext, Preisaufstellung und Bezahlsscheinvordruck aus Bibliothek **A**.

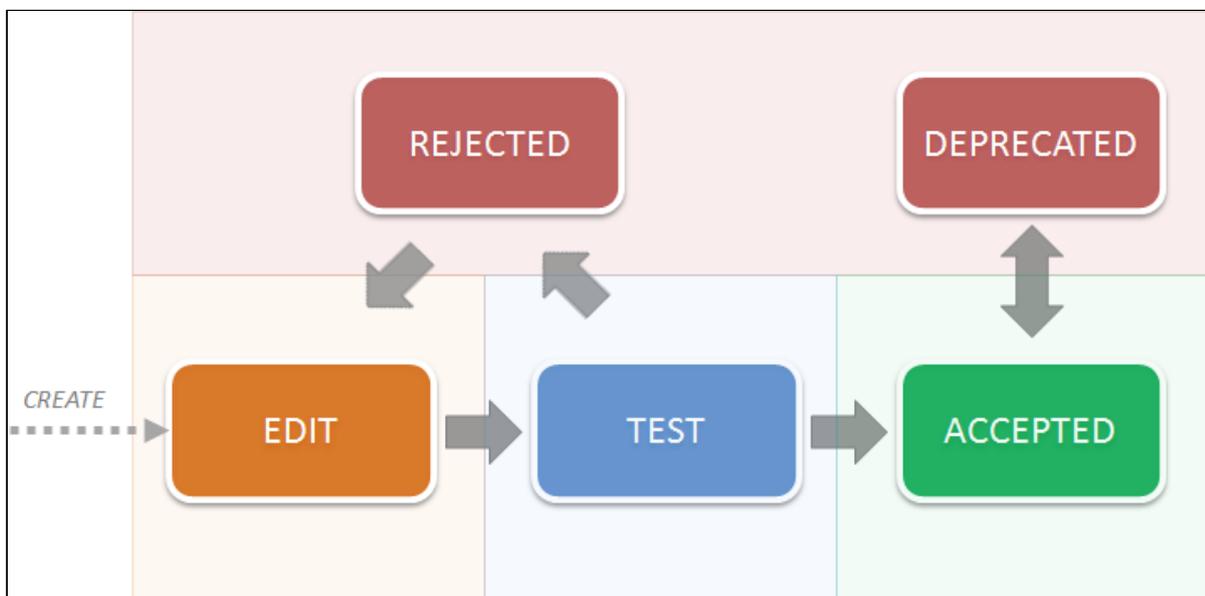
Durch die Erforderlichen Ressourcen, die in den Bibliotheken **A** und **B** definiert sind und durch die Formulare **Werbung** und **Rechnung** den Bibliotheken zur Verfügung gestellt werden, können diese Bibliotheken direkt mit den Datenstrom-Variablen arbeiten. Bibliothek **B** kann nicht in die Schema-Versionen von **Mahnung** und **Rechnung** eingebunden werden, da Bibliothek **B** die Ressource **DS.CUST\_TYPE** voraussetzt, die das den Schema-Versionen zugrunde liegende Formular **Rechnung** nicht enthält. Ebenso kann Bibliothek **A** nicht in die Schema-Versionen des Schemas **Werbe-Flyer** eingebunden werden, da das den Schema-Versionen zugrunde liegende Formular **Werbung** die Ressourcen **DS.PREIS**, **DS.ARTIKEL** und **DS.CUST\_NO** nicht enthält. Erforderliche Ressourcen können zwar auch im Schema enthalten sein, Datenstrom-Variablen stammen aber immer aus dem Formular und Ressourcen im Schema dürfen deshalb zur eindeutigen Identifizierung von Datenstrom-Variablen keine Ressourcen enthalten, die mit **"DS."** beginnen.

Verändert sich nun beispielsweise das Firmenlogo oder der Firmenslogan, so muss dies nur einmal in der Bibliothek **CI** aktualisiert werden anstatt 9 Mal. Um die oben dargestellte Struktur ohne Bibliotheken zu realisieren, müsste jede Ressource aus den Bibliotheken einzeln in jeder Schema-Version erstellt werden, insgesamt also 54 Ressourcen anstatt 9. Bei einer großen Anzahl von Formularen vereinfacht die Benutzung von Bibliotheken den Arbeitsaufwand also immens.

### Namenskonflikte

Wenn 2 Ressourcen den gleichen Namen haben, z.B. im Schema und in einer eingebundenen Bibliothek oder in 2 eingebundenen Bibliotheken, überschreibt eine Ressource die Anderen, falls dies erlaubt ist. Andernfalls kommt es zu einem Fehler. Dabei haben im Schema definierte Ressourcen immer eine höhere Priorität als die einer Bibliothek. Beschrieben wird die Hierarchie bei Bibliotheken [hier](#), das Verhalten bei Namenskonflikten [hier](#).

### Statuswechsel einer Bibliotheks-Version



Für genauere Informationen über die Statuswechsel einer Bibliotheks-Version beachten Sie den Artikel über den [R-S Workflow](#).

**Fonts**

Im R-S werden die Fonts (und ihre Ausprägungen), die ein Schema verwenden darf, im Transaktions-Formular festgelegt. Implementiert ein Schema das Transaktions-Formular, so hat es nur Zugriff auf die ihm erlaubten Fonts. Da Bibliotheken, wie bereits erwähnt, unabhängig von Formularen und Schemas sind, weiß eine Bibliothek nicht, auf welche Fonts sie nach der Einbindung in eine Schema-Version beschränkt ist. Sie können daher in einem Arbeitsbereich Fonts registrieren, um allen Bibliotheken in diesem Arbeitsbereich, analog wie bei den ERFORDERLICHEN RESSOURCEN, mitzuteilen, welche Fonts sie erwarten kann. Lesen Sie dazu die Dokumentation des Administrator-Client .

 Falls Sie in einer Bibliothek eine Schriftart verwenden, die in dem Schema, in dem Sie eingebunden ist, nicht erlaubt ist, wird der Text auch nicht in dieser Schriftart dargestellt.

Bibliotheken verwenden:	Bibliotheken werden verwendet in:
-	 <i>Schema</i>

## Material

Beileger haben Sie bereits bei den Ressourcen kennengelernt. Im Schema können Sie den Beileger als Ressource erstellen und somit den Dokumenten beifügen. Um eine Beileger-Ressource zu erstellen, müssen Sie aber zuvor ein Material für den Beileger als Vorlage erstellen und der Ressource dann zuweisen. Materialien definieren die physischen Eigenschaften des Beilegers, was wichtig für den Druckprozess und den Versand ist. Dabei gibt es für die beiden unterschiedlichen Beileger auch unterschiedliche Materialien:

 **Material für logische Beileger**

Sie werden direkt mit den Dokumenten im Anhang mit gedruckt.

 **Material für physikalische Beileger**

Sie sind bereits vorgedruckt und werden während des Kuvertiervorganges beigelegt.

## Material für logische Beileger

Da Beileger zusätzliches Material zu den eigentlichen Dokumenten bedeutet, ist es wichtig, dass mit jedem Beileger Informationen wie Gewicht oder Größe mit angegeben werden. Dies ist wichtig, um einen reibungslosen Ablauf des Druckprozesses zu gewährleisten. Die physischen Eigenschaften eines physikalischen Beilegers sind immer unbekannt, da physikalische Beileger dem Druckprozess extern zugeführt werden und sie bereits vorgedruckt sind. Auch logischen Beilegern muss ein Material zugewiesen werden. Die Erstellung eines eigenen Materials ist allerdings nur dann nötig, wenn der logische Beileger nicht auf demselben Material / Papier gedruckt wird wie die Dokumente des Formulars. Dann müssen Sie ein Material mit den Eigenschaften des fremden Papiers erstellen und dem logischen Beileger zuweisen. Wird der Beileger allerdings auf demselben Material gedruckt wie die Dokumente des Formulars, müssen Sie ihm ein durch das Formular zur Verfügung gestelltes Material zuweisen. Zu diesem Zweck können Formulare Materialien mitliefern. Liefert das Formular keine Materialien mit, müssen Sie auf jeden Fall ein eigenes Material erstellen, um einen logischen Beileger definieren zu können. Formulare können die Verwendung formularfremden Materials auch ganz verbieten. In diesem Fall können Sie Beilegern nur Materialien aus dem Formular zuweisen und keine anderen Materialien für logische Beileger erstellen und verwenden. Natürlich muss das Formular dann Materialien mitliefern.

Sowohl für logische als auch physikalische Beileger müssen **immer** Materialangaben gemacht werden.

### **Beim Material für logische Beileger sind folgende Angaben obligatorisch:**

- **Die Dimensionen des Beilegers**
  - Dazu müssen Sie Höhe, Breite und Dicke des Beilegers angeben. Die Dicke ist abhängig vom verwendeten Papier und wichtig, um zu berechnen, wie dick der Umschlag durch die Dokumente wird. Die Angaben erfolgen in mm.
- **Das Gewicht des Beilegers**
  - Die Angaben zum Gewicht des Beilegers sind nötig, um das Porto der Sendung zu berechnen. Die Angaben erfolgen in Gramm.

 Wenn Sie später den logischen Beileger gestalten, hat der Beileger fest die Größe, die das verwendete Material angibt, da der Beileger ja auch diese Maße haben wird. Die Angaben, die zum Material des Beilegers gemacht werden, müssen unbedingt korrekt sein, da es sonst beim Druck- und Versandprozess zu Komplikationen kommen kann.

Wie Sie Materialien für logische Beileger erstellen und verwalten lernen Sie [hier](#).

## Material für physikalische Beileger

Wie bereits beim Material für logische Beileger erläutert, dienen Materialien im R-S dazu, die physischen Eigenschaften eines Beilegers festzulegen. Diese Angaben sind wichtig für Druckprozess und Versand. Während Sie bei logischen Beilegern nur dann eigenes Material erstellen müssen, wenn der logische Beileger nicht auf demselben Material / Papier gedruckt wird wie die eigentlichen Dokumente, müssen Sie dies bei physikalischen Beilegern immer tun, da physikalische Beileger extern zugeführt werden und keine Kenntnis über ihre Eigenschaften besteht.

### **Beim Material für physikalische Beileger sind folgende Angaben obligatorisch:**

- **Die Dimensionen des Beileger**
  - Dazu müssen Sie Höhe, Breite und Dicke des Beilegers angeben. Die Dicke ist abhängig vom verwendeten Papier und wichtig, um zu berechnen, wie dick der Umschlag durch die Dokumente wird. Die Angaben erfolgen in mm.
- **Das Gewicht des Beilegers**
  - Die Angaben zum Gewicht des Beilegers sind nötig, um das Porto der Sendung zu berechnen. Die Angaben erfolgen in Gramm.
- **Der Falz-Typ des Beilegers**
  - Der Falz-Typ bestimmt, ob der Beileger bereits gefalzt (gefaltet) vorliegt oder nicht. Ist der Beileger noch nicht gefalzt, kann er automatisch gefalzt werden, wenn dies erlaubt ist. Beachten Sie, dass wenn der Beileger nicht gefalzt werden darf, die benötigte Umschlaggröße dem Format des Beilegers entspricht. Optionen hier sind: STANDARD, GEFALZT, FALZBAR, NICHT FALZBAR.
- **Der Falz-Layer des Beilegers**
  - Liegt der Beileger gefalzt vor, müssen Sie angeben, wie viele Papierschichten der gefaltete Beileger hat. Liegt der Beileger bspw. im Zickzack-Falz vor, ist der Falz-Layer 3. Diese Angabe ist wichtig, da die Dicke der Beileger dadurch schwankt.

✔ Außerdem können Sie dem Material eine Vorschau anheften, die den physikalischen Beileger zeigt (z.B. Scan). So ist es für Sie und andere R-S Nutzer einfacher, den Beileger richtig zu konfigurieren und damit zu arbeiten.

⚠ Die Angaben, die zum Material des Beilegers gemacht werden, müssen unbedingt korrekt sein, da es sonst beim Druck- und Versandprozess zu Komplikationen kommen kann.

Wie Sie Materialien für physikalische Beileger erstellen und verwalten lernen Sie [hier](#).

## Docxworld-Links

Docxworld-Links dienen dazu, fertige Produktions-Pakete für den weiteren, externen Produktions-Prozess verfügbar zu machen. Durch Docxworld-Links können Komponenten wie der **Spooler** das gewünschte Produktions-Paket mit Hilfe der den Dokumenten zugehörigen Docxworld-Vertragsnummern finden, es exportieren und weiterverarbeiten. Sie dienen also als Schnittstelle zwischen dem R-S und externen Komponenten, die den weiteren Dokument-Erstellungsprozess und den Druckprozess übernehmen. Eine Alternative zu Docxworld-Links, falls Sie kein **Docxworld** für Ihre Produktion verwenden, sind **Produktions-Paket Verknüpfungen**. Docxworld-Links und Produktions-Paket Verknüpfungen sind die 2 Möglichkeiten, um Produktions-Pakete und damit fertige Dokumente aus dem R-S zu exportieren.

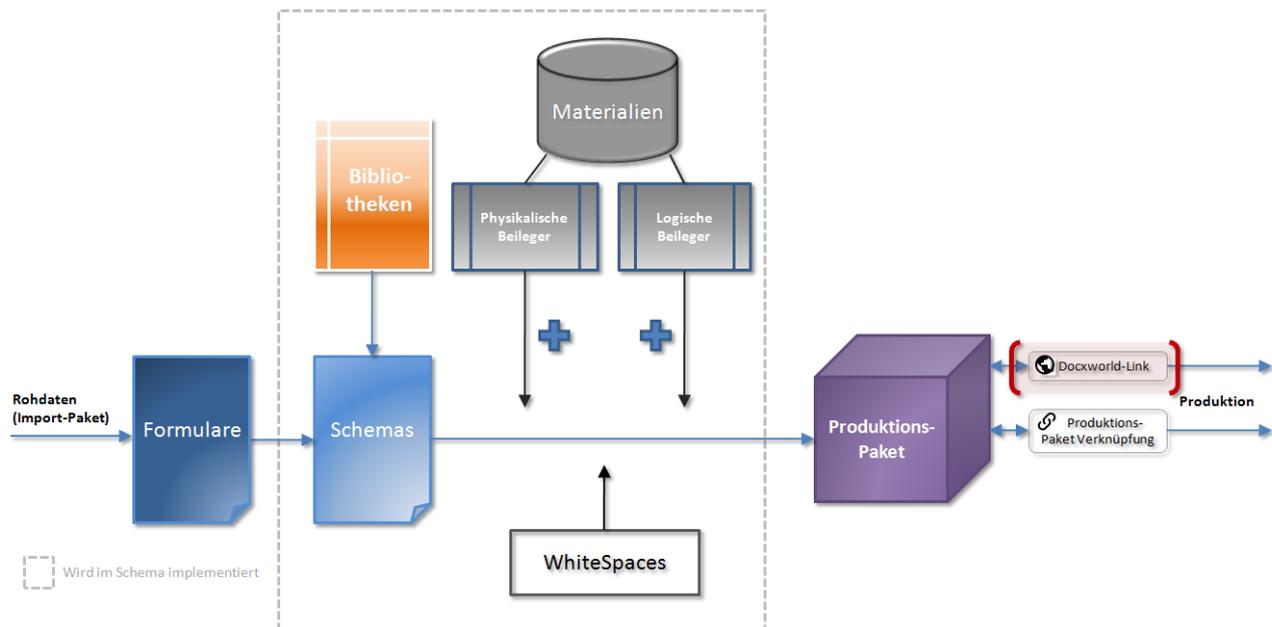


Abbildung A

In einem Docxworld-Link geben Sie zum einen die Vertragsnummer an, zum anderen das dazugehörige **Schema**. Da im R-S von einem Schema immer nur ein produktives Produktions-Paket existieren darf, geben Sie im Link nicht das Paket selbst, sondern das Schema an. Dies hat den Vorteil, dass wenn das produktive Produktions-Paket eines Schemas wechselt, automatisch das neue Paket verwendet wird, da der Link immer auf das Paket dieses Schemas zeigt, dass PRODUCTION -Status hat. Es besteht aber auch die Möglichkeit, direkt ein Produktions-Paket auszuwählen.

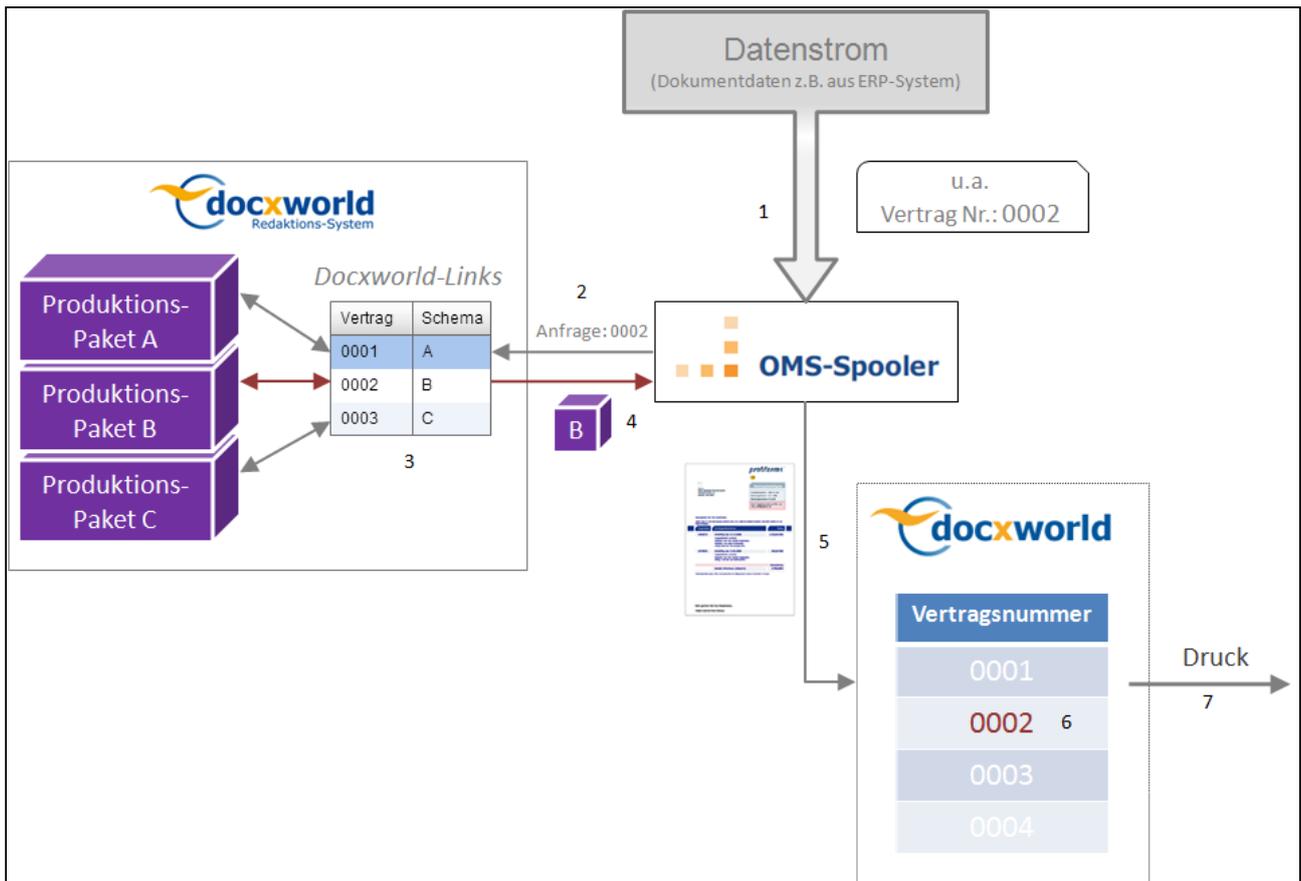


Abbildung B

Im Folgenden wird die Funktionsweise von Docxworld-Links vorgestellt (unter Nutzung des Spoolers):

Nummer	Beschreibung
1	Der Datenstrom sendet die dokumentspezifischen Daten an den Spooler (Datenstrom-Variablen, siehe <b>Umgebung des R-S</b> und <b>Datenstrom-Variablen</b> ). Unter anderem enthält der Datenstrom die den Dokumenten zugehörige Docxworld-Vertragsnummer (hier als Beispiel: 0002).
2	Der Spooler fragt beim R-S an, ob die Vertragsnummer 0002 mit einem Produktions-Paket verknüpft ist.
3	Ist die Vertragsnummer mit einem Produktions-Paket verknüpft, gibt das R-S das Paket inklusive dem zugehörigen <b>Binär-Paket</b> (Lesen Sie dazu auch den Abschnitt über die <b>Umgebung des R-S</b> ) an den Spooler zurück.
4	Der Spooler empfängt das Produktions-Paket (und das Binär-Paket) vom R-S (hier: Produktions-Paket B).
5	Die Programme des Binär-Pakets generieren aus dem Produktions-Paket und den Daten aus dem Datenstrom die fertigen Dokumente. Der Spooler sendet diese an Docxworld.

Nummer	Beschreibung
6	Durch die Vertragsnummer können die Dokumente im Docxworld ihrem Vertrag zugeordnet werden, der alle nötigen Informationen über das weitere Verfahren mit den Dokumenten enthält .
7	Die Dokumente werden in den richtigen Druck geschickt.

Da Docxworld-Vertragsnummern eindeutig sind und nie doppelt vergeben, ist auch die Zuordnungstabelle des R-S arbeitsbereichsübergreifend. Das bedeutet, dass alle bestehenden Docxworld-Vertrags-Verlinkungen aller installierten R-S Clients in einer zentralen Zuordnungstabelle gespeichert sind. Sie selbst sehen nur die Verlinkungen Ihres Arbeitsbereichs. Wenn Sie eine Docxworld-Verlinkung mit einer Vertragsnummer erstellen möchten, die bereits durch jemand anderen vergeben ist, erscheint folgende Meldung:



Beachten Sie, dass jede Vertragsnummer nur einmal existiert. Wenn Sie diese Meldung sehen, kontrollieren Sie bitte, ob die von Ihnen verwendete Vertragsnummer korrekt ist.

### Testproduktion

Neben der Möglichkeit, Produktions-Pakete durch Docxworld-Links für die Produktion verfügbar zu machen, können Sie in einem Docxworld-Link auch Testproduktionen zur Qualitätssicherung definieren. Dabei verlinken Sie auf ein Produktions-Paket, das nicht zwingend PRODUCTION -Status haben muss, sondern nur TEST -Status. Wie Sie Produktions-Pakete zur Testproduktion freigeben können erfahren Sie [hier](#) . Das Paket wird dann ebenfalls wie in [Abb. B](#) verarbeitet und zu Docxworld geschickt, dort allerdings nur in die Testproduktion gesendet. Um Pakete in die Testproduktion schicken zu können, müssen Sie bei uns einen Docxworld-Testvertrag registrieren lassen, der dann nur für Testproduktionen freigeschaltet ist. Wenden Sie sich dazu bitte an unseren [Support](#) .

Docxworld-Links verwenden:	Docxworld-Links werden verwendet in:
<b>Produktions-Pakete</b>	—

## Produktions-Paket Verknüpfungen

Produktions-Paket Verknüpfungen dienen wie Docxworld-Links dazu, fertige Produktions-Pakete für den weiteren, externen Produktions-Prozess verfügbar zu machen. Wenn Sie kein Docxworld für Ihre Produktion nutzen, müssen Sie hierfür **Produktions-Paket Verknüpfungen** verwenden. Durch Produktions-Paket Verknüpfungen können Komponenten wie der **Spooler** das gewünschte Produktions-Paket mit Hilfe des Dokumentennamen finden, es exportieren und weiterverarbeiten. Produktions-Paket Verknüpfungen funktionieren praktisch gleich wie Docxworld-Links, nur dass Sie anstatt der Docxworld-Vertragsnummer den Namen der Dokumente, den das ERP-/HOST-/CRM-System (z.B. SAP) vorgibt, angeben. Docxworld-Links und Produktions-Paket Verknüpfungen sind die 2 Möglichkeiten, um Produktions-Pakete und damit fertige Dokumente aus dem R-S zu exportieren.

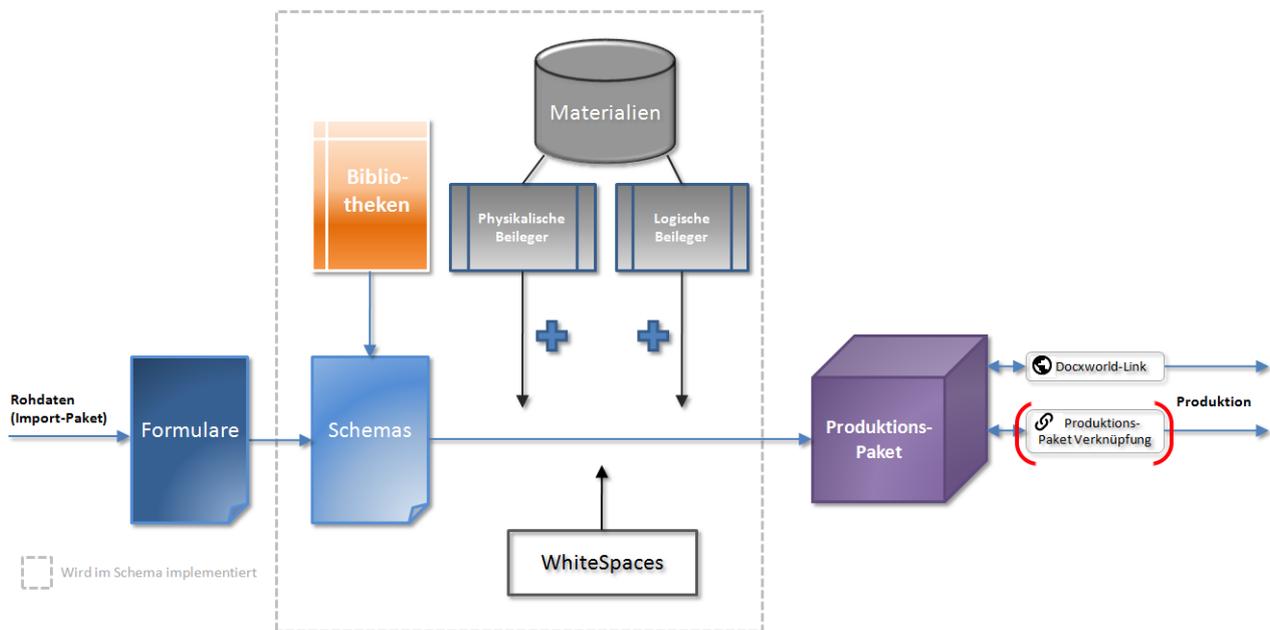


Abbildung A

In einem Docxworld-Link geben Sie zum einen den Dokumentennamen an, zum anderen das dazugehörige Produktions-Paket. Der Dokumentenname, den Sie dem Produktions-Paket in der Verknüpfung zuweisen, muss mit dem Formularnamen aus dem Datenstrom übereinstimmen. (Jedes Formular im Datenstrom hat einen Formularnamen. Dieser Formularnamen muss nicht eindeutig sein, vielmehr können z.B. alle Rechnung von Mai 2013 den gleichen Namen "Rechnung\_Mai\_13" haben. Wenn Sie den Formularnamen nicht wissen, wenden Sie sich an Ihren System-Administrator. Da im R-S von einem Schema immer nur ein produktives Produktions-Paket existieren darf, geben Sie im Link nicht das Paket selbst, sondern das Schema an. Dies hat den Vorteil, dass wenn das produktive Produktions-Paket eines Schemas wechselt, automatisch das neue Paket verwendet wird, da der Link immer auf das Paket dieses Schemas zeigt, das PRODUCTION -Status hat. Es besteht aber auch die Möglichkeit, direkt ein Produktions-Paket auszuwählen.

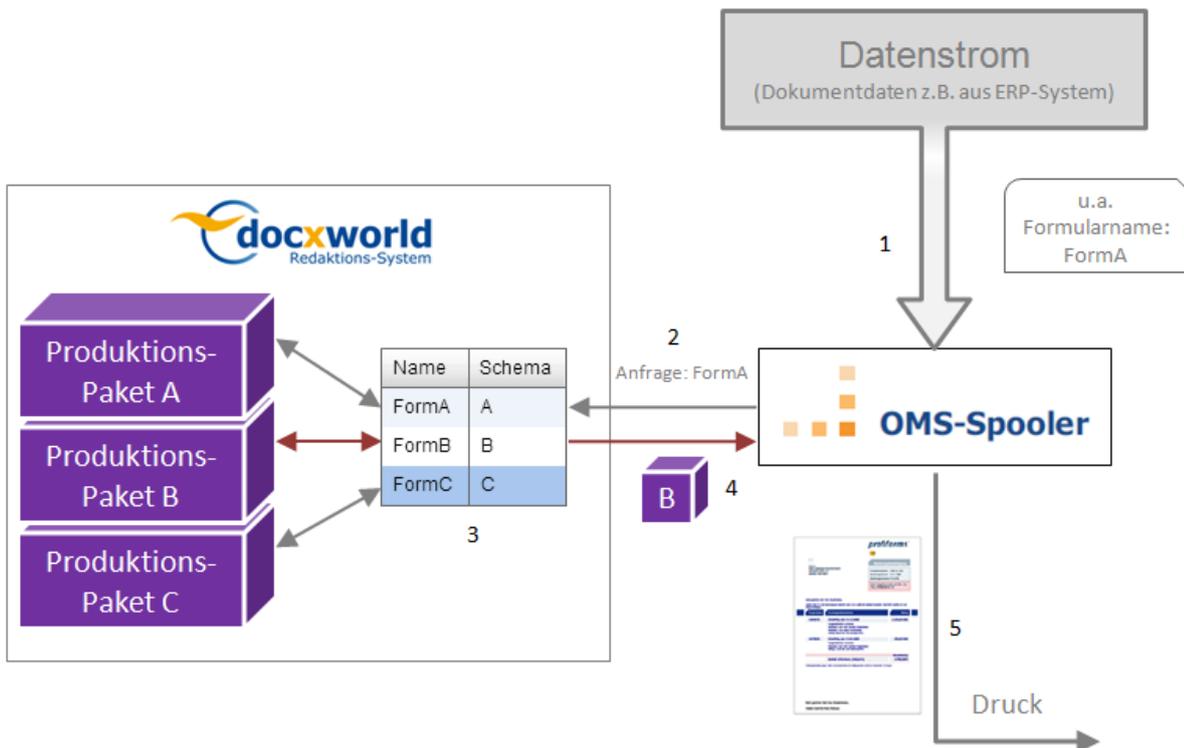


Abbildung B

Im Folgenden wird die Funktionsweise von Produktions-Paket Verknüpfungen vorgestellt (unter Nutzung des Spoolers):

Nummer	Beschreibung
1	Der Datenstrom sendet die dokumentspezifischen Daten an den Spooler (Datenstrom-Variablen, siehe <b>Umgebung des R-S</b> und <b>Datenstrom-Variablen</b> ). Unter anderem enthält der Datenstrom den Namen der Dokumente (<name>NAME</name>). Dieser im Datenstrom festgelegte Formularname dient als Verknüpfungspunkt. Testdaten haben übrigens als Namen im R-S den Formularnamen des simulierten Testdatenstroms (hier: FormA).
2	Der Spooler fragt beim R-S an, ob mit dem Formularnamen FormA ein Produktions-Paket verknüpft ist.
3	Ist der Formularname mit einem Produktions-Paket verknüpft, gibt das R-S das Paket inklusive dem zugehörigen <b>Binär-Paket</b> (Lesen Sie dazu auch den Abschnitt über die <b>Umgebung des R-S</b> ) an den Spooler zurück.
4	Der Spooler empfängt das Produktions-Paket (und das Binär-Paket) vom R-S (hier: Produktions-Paket B).
5	Die Programme des Binär-Pakets generieren aus dem Produktions-Paket und den Daten aus dem Datenstrom die fertigen Dokumente. Der Spooler sendet diese weitere zum nächsten Produktionsschritt (evtl. eines Drittanbieters).

Zwar werden Produktions-Paket Verknüpfungen keinem Docxworld-Vertrag zugeordnet (da ja kein Docxworld genutzt wird), es existieren für Produktions-Paket Verknüpfungen aber trotzdem lokale Übersichtstabellen. Im Gegensatz zu Docxworld-Links, bei denen alle Links aller R-S Nutzer global gespeichert werden, damit ein Vertrag nicht 2 Produktionen zugeordnet werden kann, werden in der Zuweisungstabelle für Produktions-Paket Verknüpfungen nur die Verknüpfungen, die in den Arbeitsbereichen, die durch denselben Datenstrom erreichbar sind, gespeichert. Dies ist nötig, um einen Datenstrom eindeutig einem Produktions-Paket zuweisen zu können. Enthielten in unterschiedlichen Arbeitsbereichen 2 Produktions-Paket Verknüpfungen denselben Namen, wüsste der Datenstrom nicht, welches Produktions-Paket er exportieren und benutzen soll. Sie selbst sehen nur die Verknüpfungen Ihres Arbeitsbereichs. Wenn Sie eine Produktions-Paket Verknüpfung erstellen möchten, die bereits (in Ihrem oder einem anderen Arbeitsbereich) vergeben ist, erscheint folgende Meldung:



 Diese Meldung bedeutet, dass bereits ein anderes Produktions-Paket für diesen Datenstrom vorgesehen ist. Sprechen Sie sich mit Ihrem R-S Administrator und den anderen R-S Nutzern ab.

**Testproduktion**

Neben der Möglichkeit, Produktions-Pakete durch Produktions-Paket Verknüpfungen für die Produktion verfügbar zu machen, können Sie wie in einem Docxworld-Link auch durch Produktions-Paket Verknüpfungen Testproduktionen zur Qualitätssicherung definieren. Dabei verknüpfen Sie auf ein Produktions-Paket, das nicht zwingend PRODUCTION -Status haben muss, sondern nur TEST -Status. Wie Sie Produktions-Pakete zur Testproduktion freigeben können erfahren Sie [hier](#). Eventuell müssen Sie sich mit Ihrem Druckanbieter in Kontakt setzen, um Pakete in die Testproduktion schicken zu können.

Prod.-Paket Verknüpfungen verwenden:	Prod.-Paket Verknüpfungen werden verwendet in:
 <b>Produktions-Pakete</b>	—

## Erweiterungen

In den vorigen Abschnitten haben Sie Ressourcen kennengelernt und gesehen, wie man mit ihnen arbeitet. Bisher wurde aber nicht erklärt, wie genau man Ressourcen in das Dokument einfügt. Dies wird mit Hilfe von Erweiterungen realisiert. Mit ihnen können Sie Ressourcen in Dokumente einfügen und Beileger an die Dokumente anheften.

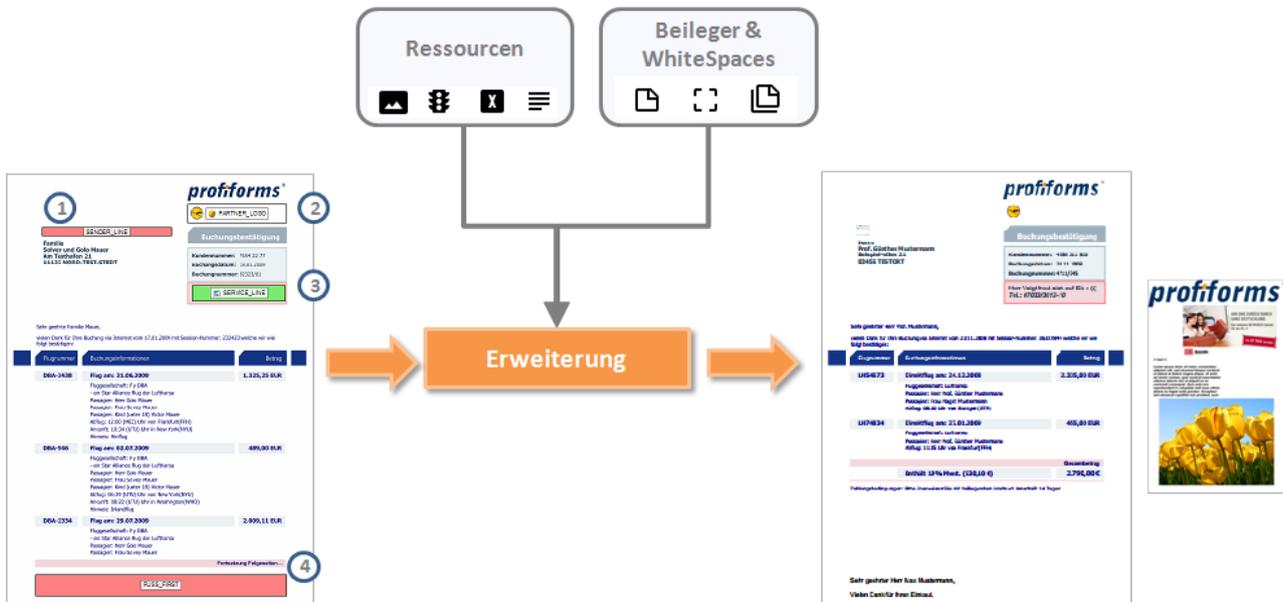


Abbildung A

In Abb. A sehen Sie, wie Erweiterungen verwendet werden (*hier speziell Layout-Bereichs Erweiterungen, siehe unten*). Links sehen Sie das Rohdokument mit den Erweiterungs-Punkten (1) - (4). In diese Erweiterungs-Punkte können Sie durch Erweiterungen Ressourcen einfügen (und nur dort). Durch die Erweiterung wird jeweils eine Ressource in einen Erweiterungs-Punkt eingefügt. Beileger hingegen werden dem Dokument durch die Erweiterung einfach hinzugefügt, WhiteSpaces können nur freien Flächen zugewiesen werden. Rechts sehen Sie das Enddokument mit den eingefügten Ressourcen.

Es gibt vier Typen von Erweiterungen:

- Erweiterungen für Layout-Bereiche
- Erweiterungen für physikalische Beileger
- Erweiterungen für logische Beileger
- Erweiterungen für WhiteSpaces

Für die Materialien physikalische und logische Beileger und WhiteSpaces gibt es jeweils eigene Erweiterungs-Typen, mit denen Sie diese zum Dokument hinzufügen können. Lesen Sie dazu auch die Definitionen der verschiedenen Materialien. Für Ressourcen gibt es Layout-Bereiche:

### Erweiterungen für Layout-Bereiche

Als Layout-Bereiche werden die Bereiche im Dokument bezeichnet, in die per Erweiterung eine Ressource eingefügt werden kann. Erweiterungen für Layout-Bereiche sind die Erweiterungen, mit denen Sie das Dokument gestalten.

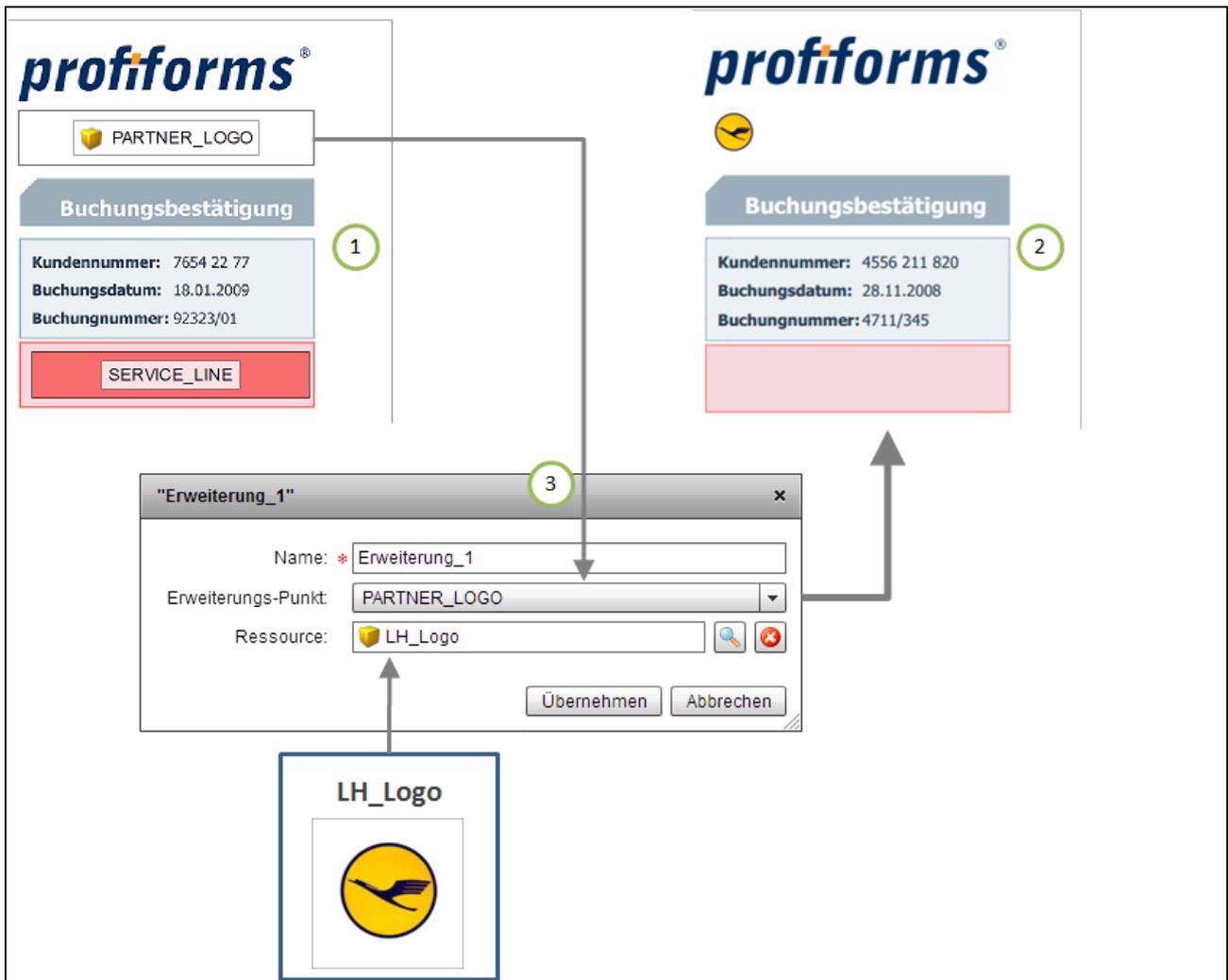


Abbildung B

Wenn Sie eine Layout-Bereichs Erweiterung erstellen, geben Sie in der Erweiterung einen Erweiterungs-Punkt und eine Ressource an. In Abb. B (1) sehen Sie das Rohdokument mit dem Erweiterungs-Punkt PARTNER\_LOGO. Die Erweiterung in (3) bindet die Ressource LH\_Logo an den Erweiterungs-Punkt. In (2) sehen Sie dann das Resultat bei der Dokumenterstellung mit der eingefügten Ressource. Ressourcen können nur dort im Dokument eingefügt werden, wo ein Erweiterungs-Punkt definiert ist. Die Erweiterungs-Punkte können vom R-S Nutzer nicht geändert werden und sind durch das Transaktions-Formular fest vorgegeben.

Erweiterungen definieren Sie in der Registerkarte Erweiterungen in Schema-Versionen. Wie Sie in der Praxis mit Erweiterungen arbeiten lernen Sie [hier](#).

## Erste Schritte

Nachdem Sie sich nun mit den theoretischen Grundlagen und dem Aufbau des R-S vertraut gemacht haben, lernen Sie in diesem Abschnitt, wie Sie sich im R-S zurechtfinden.

- Zunächst zeigen wir Ihnen, wie man sich in einem Arbeitsbereich **anmeldet**.
- Im 2. Schritt machen wir Sie mit der **Nutzeroberfläche** und der **Navigation** vertraut:
  - Sie lernen die Benutzeroberfläche und die verschiedenen Navigationsbereiche des R-S kennen und erfahren, wie Sie mit den Übersichtstabellen arbeiten. Sie sehen außerdem, wie Sie den Detail-Bereich mit den Übersichtstabellen mit Hilfe verschiedener Ansichtstypen anpassen können.
- Zuletzt erfahren Sie mehr über **allgemeine Aspekte** zur Nutzung des R-S:
  - Über die verschiedenen Hilfe-Systeme im R-S, wie Eingabefelder im R-S zu nutzen sind und wie Berechtigungen eines Nutzers die Sichtbarkeit der verschiedenen Elemente im R-S bestimmen.

## Anmeldung am R-S Client

Nachdem Sie das R-S in ihrem Browser geöffnet haben, werden Sie als Erstes gebeten, sich anzumelden. Dazu erscheint folgender Dialog:

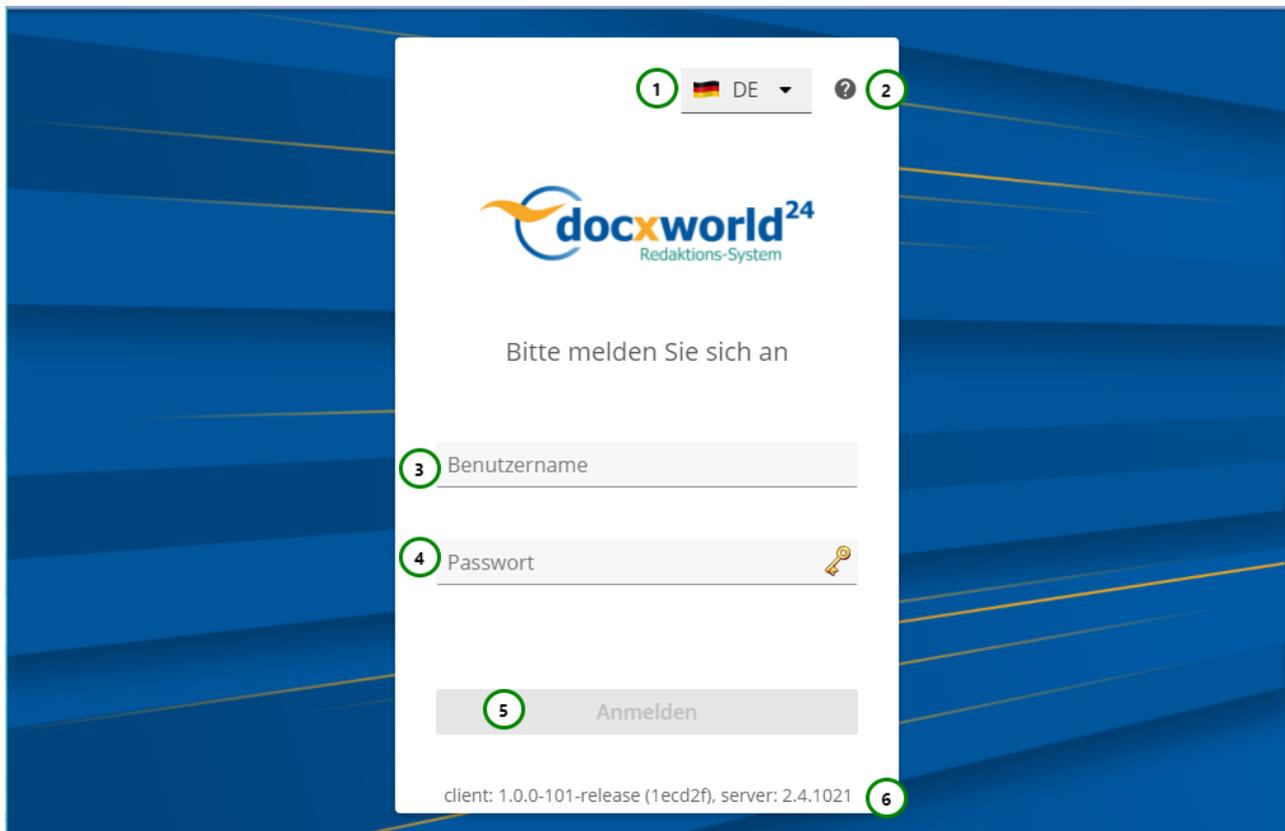


Abbildung A

Um sich anzumelden, müssen Sie Ihren Nutzernamen Abb. A (3) und Ihr Passwort (4) eingeben.

### Sie können außerdem:

- eine **Sprache** wählen Abb. A (1).
  - Um die Sprache zu ändern, klicken Sie bitte auf (1) und wählen eine andere Sprache aus. Die Standardsprache ist deutsch. Momentan verfügbare Sprachen sind Deutsch und Englisch .
- eine **Hilfe** zur Anmeldung aufrufen. Klicken Sie auf (2), um zu dieser Online-Dokumentation weitergeleitet zu werden



- Achten Sie hierbei auf **Groß-** und **Kleinschreibung!**
- Um sich anmelden zu können benötigen Sie ein **Benutzerkonto**.
  - Falls Sie kein Benutzerkonto haben, wenden Sie sich bitte an ihrem Administrator, damit dieser Ihnen ein Konto einrichtet.



In Abb. A (6) wird Ihnen die Versionsnummer Ihres R-S Clients und des R-S Servers angezeigt.

Nachdem Sie alle **Pflichtfelder** ausgefüllt haben, klicken Sie auf Abb. A (5). Sie sind nun eingeloggt und befinden sich auf dem ersten, für Sie freigegebenen, Arbeitsbereich (alphabetisch sortiert).

Zum ändern des Arbeitsbereiches können Sie, nachdem Sie sich eingeloggt haben, oben rechts einen Arbeitsbereich auswählen:



Abbildung B: Arbeitsbereich auswählen



Wenn Sie nur für einen **Arbeitsbereich** zugelassen sind, haben Sie hier keine weitere Auswahl!

Wenn Sie den Arbeitsbereich wechseln wollen, wählen Sie diesen aus der Liste Abb. B (1) aus. Sie wechseln nun automatisch in den neuen Arbeitsbereich.

## Nutzeroberfläche und Navigation

In diesem Kapitel werden Sie mit der Nutzeroberfläche des R-S und seinen Bedienelementen vertraut gemacht. Sie lernen, sich im R-S zurechtzufinden und erfahren, wie Sie mit Eingabefeldern und Hilfe-Systemen arbeiten können und was es bei der Bedienung des R-S hinsichtlich Berechtigungen, Sichtbarkeit und Datentypen zu beachten gibt.

Dieses Kapitel beinhaltet 2 wesentliche Abschnitte:

### Die Nutzeroberfläche des R-S:

- **Die 3 Layout-Bereiche im R-S:**
  - Die **Top-Navigation** mit grundlegenden Funktionen.
  - Der **Navigations-Pfad**, der die aktuelle Position im R-S anzeigt.
  - Der **Detail-Bereich**: Hier befinden sich die Übersichtstabellen, mit denen Sie arbeiten.
- Die verschiedenen Ansichten, die es ermöglichen, den Detail-Bereich zu konfigurieren.
- Aspekte wie das Abmelden vom R-S Client im R-S, die Systemverwaltung und die Hilfe.

### Die Bedienung des R-S:

- Die **Pflicht-Eingabefelder** im R-S, insbesondere die Pflichtfelder.
- Wie sich Nutzerrechte auf **Sichtbarkeit und Berechtigungen** auswirken.
- Die **Hilfe-System**, die in das R-S integriert sind.
- Die **Datentypen** im R-S und wie die Eingabefelder sie validieren.

## Nutzeroberfläche des R-S

In diesem Kapitel machen wir Sie mit dem Aufbau der Bildschirm-Masken und den Möglichkeiten zur Navigation vertraut. Es wird außerdem gezeigt, welche Ansichten Ihnen im R-S zur Verfügung stehen und welche Objekte sie jeweils beinhalten.

Nachdem Sie sich angemeldet haben, werden Sie direkt in den ersten Arbeitsbereich weitergeleitet.

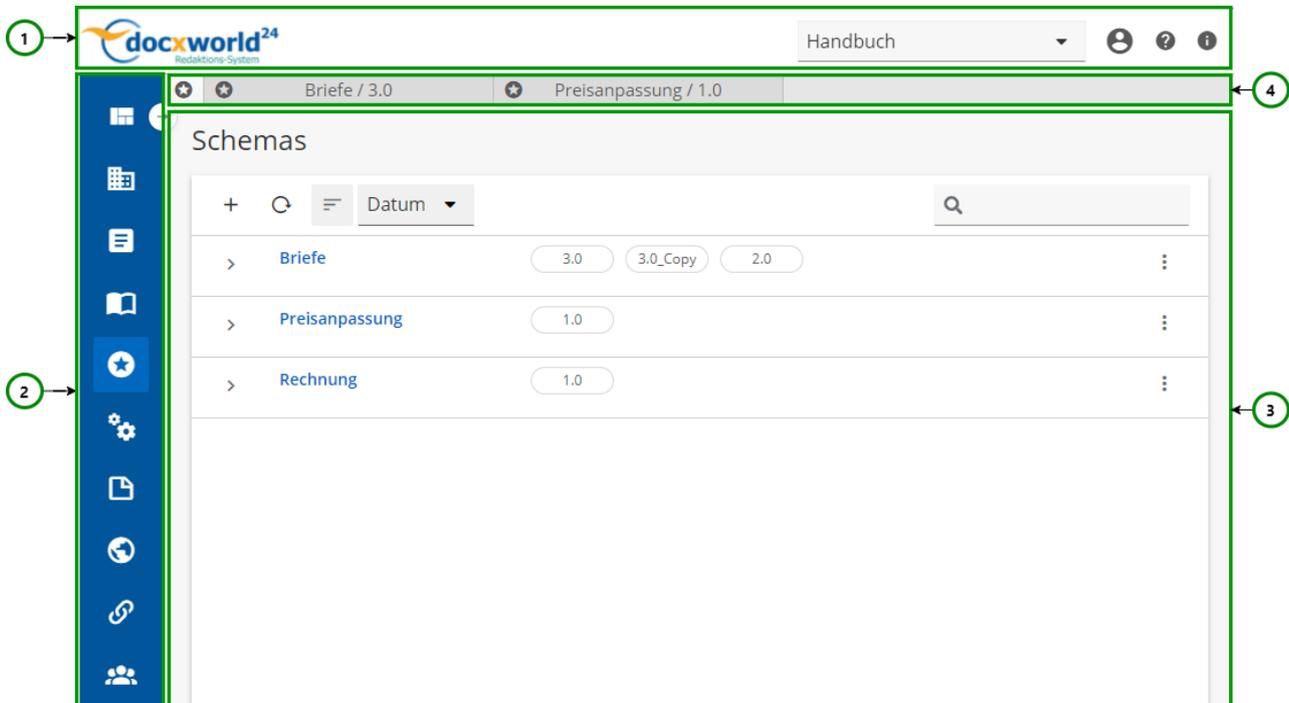


Abbildung A

**Der Aufbau der Bildschirm-Masken besteht immer aus vier Teilen:**

- **Header:** Hier finden Sie grundlegende Steuerungsoptionen des R-S *Abb. A (1)*.
- **Seitennavigation:** Mit dem Navigations-Pfad wechseln Sie durch die Basisobjekte (2).
- **Detail:** Der Detail-Bereich beinhaltet die Übersichtstabellen, mit denen Sie arbeiten (3).
- **Tab:** Hier sehen Sie die geöffneten Objekte (4).

## Bereich - Header



Abbildung A

In dem Header finden Sie die grundlegenden Elemente zur Steuerung Ihrer R-S Applikation.

### Im Bereich der Top-Navigation können Sie:

- **Arbeitsbereiche** wechseln *Abb. A (1)*.
- **Nutzeraktionen** auswählen (2).
- **Hilfe** erhalten (3).
- **Informationen** über Client und Server erhalten (4).

#### ▼ **Arbeitsbereiche wechseln:**

Wenn Sie den Arbeitsbereich wechseln möchten, werden Ihnen alle Arbeitsbereiche, zu denen Sie Zugang haben, in der Übersichtstabelle zur Auswahl angeboten. Was Arbeitsbereiche sind haben Sie bereits [hier](#) erfahren.

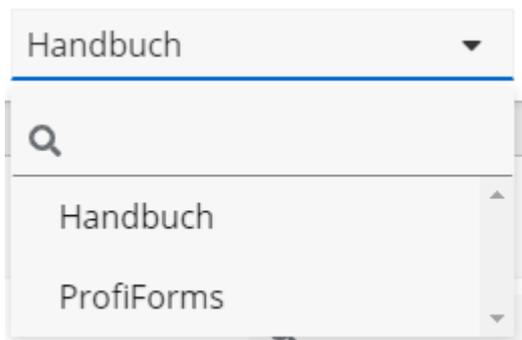


Abbildung A

Um den Arbeitsbereich zu wechseln, wählen Sie aus der Übersichtstabelle den gewünschten Arbeitsbereich aus. Sie landen danach automatisch auf der gleichen Seite des neuen Arbeitsbereiches.

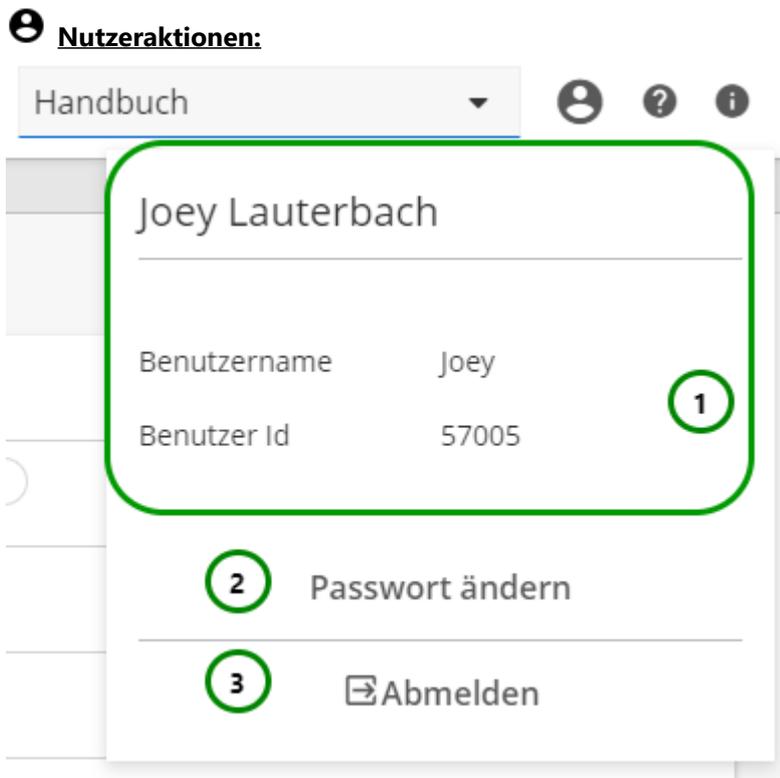


Abbildung B

Das Profil zeigt Ihnen Informationen Abb. B (1) über den Nutzer des Arbeitsbereichs. Sie sehen den Vor- und Nachname des Nutzers, den Benutzernamen und die Profil-ID.

Sie können sich von hier aus Abmelden (3) oder ein neues Passwort vergeben (2).

Passwort ändern

Neues Passwort \*

Neues Passwort wiederholen \*

Altes Passwort \*

Abbrechen Speichern

Abbildung C - neues Passwort vergeben

**? Hilfe:**

Hier kommen Sie direkt zu dieser Online-Dokumentation.

**i Informationen:**

Es öffnet sich ein neues Fenster mit den Informationen zur Client- und Serverversion.



## Bereich - Seitennavigation

Mithilfe der Seitennavigation können Sie sich im Redaktions-System bewegen. Jeder Menüeintrag führt Sie zu diesem Basisobjekt, dessen Übersichtslisten anschließend im Detail-Bereich angezeigt werden.

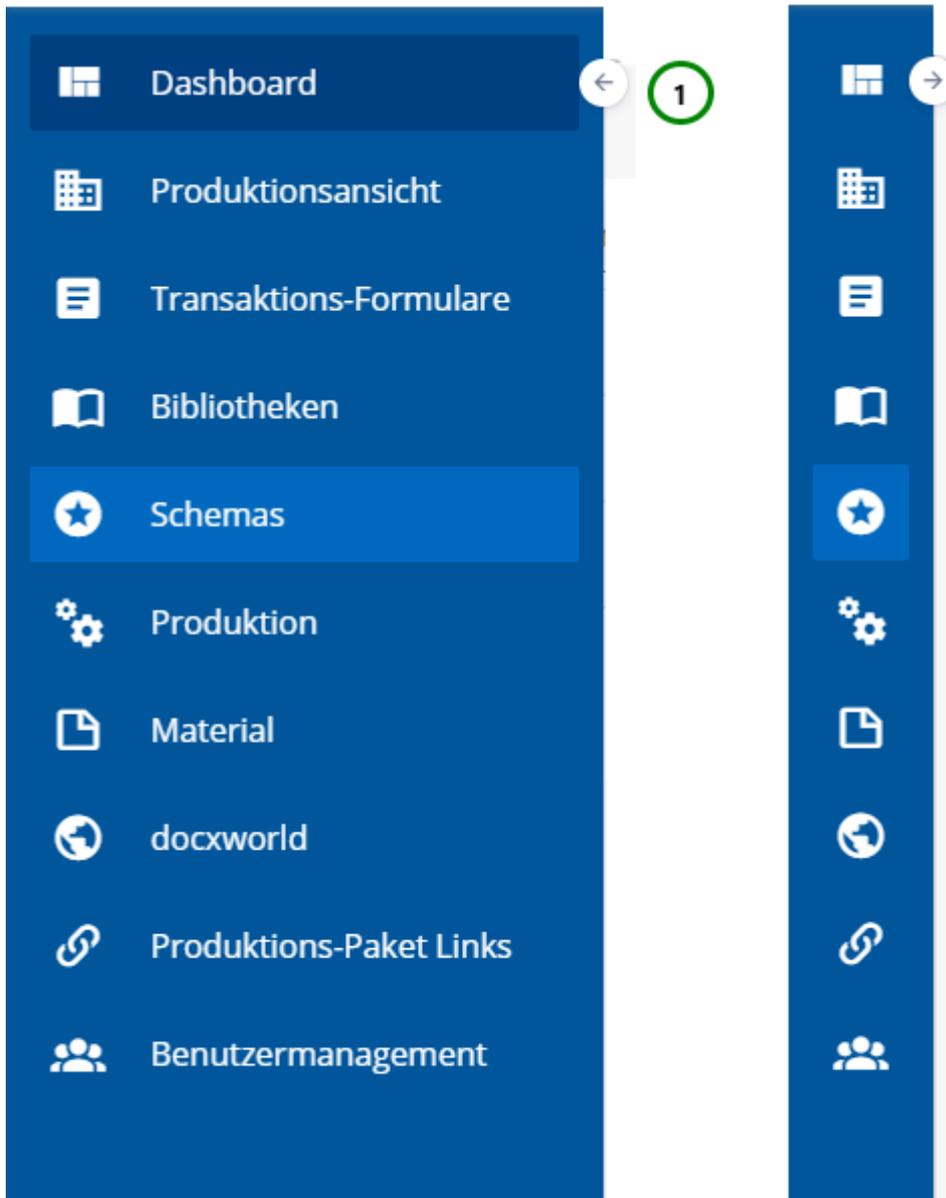


Abbildung A

Das Menü ist immer eingeklappt und zeigt Ihnen anhand von Icons die Objekte, die Sie damit erreichen. Sie können das Menü mit  *Abb. A (1)* auch aufklappen, um so den Namen des dazugehörigen Basisobjektes zu sehen.

Sie sehen zu jeder Zeit die Seitennavigation und können sich auch während der Bearbeitung von Versionen in andere Bereiche bewegen. Mithilfe der Tabs kommen Sie wieder zurück in Ihre Versions-Bearbeitung.



Benutzen Sie zum Aktualisieren der R-S Seite nicht den Refresh-Button Ihres Browsers, da sonst das gesamte R-S neu geladen wird und Sie aus dem aktuellen Arbeitsbereich abgemeldet werden.

## Bereich - Detail

Der Detail-Bereich beinhaltet die Übersichtstabellen Abb. A (grüne Markierung), die Ihnen das Arbeiten im R-S ermöglichen.

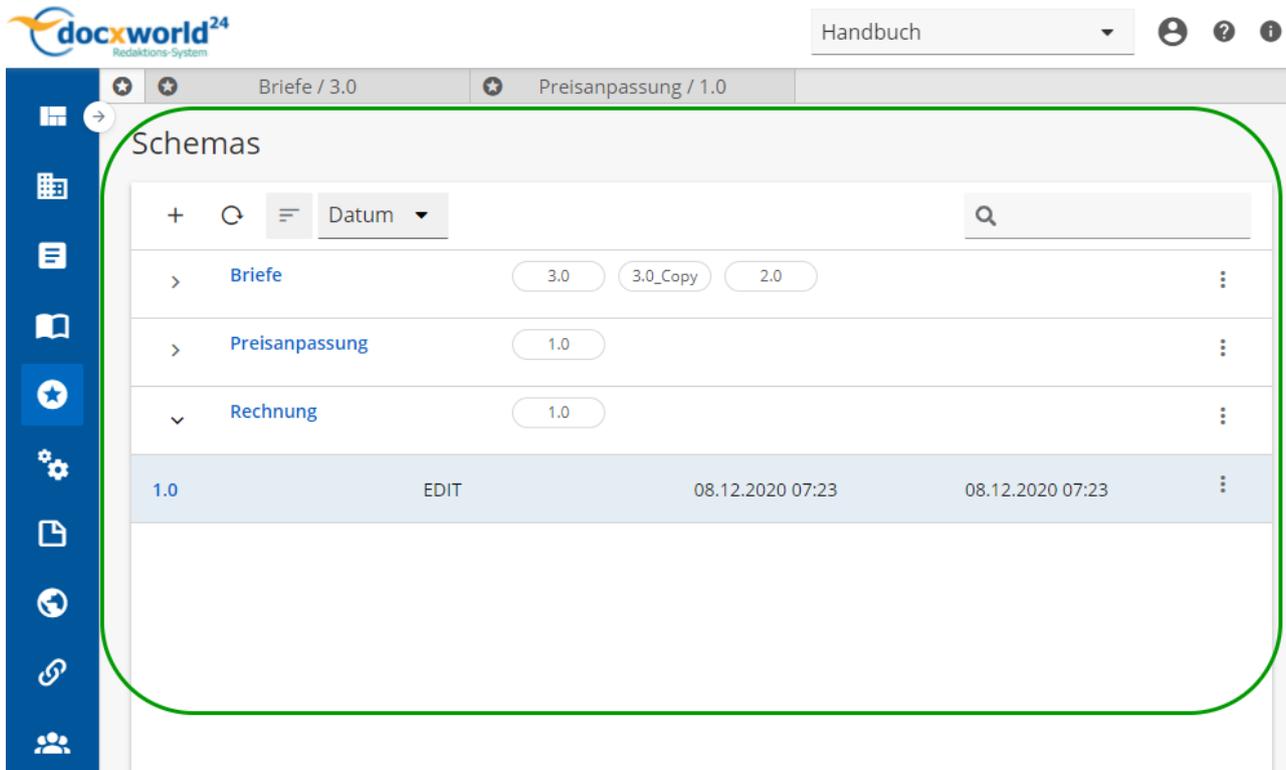


Abbildung A

Je nach gewählter Ansicht werden Ihnen zu unterschiedlichen Objekten Übersichtstabellen angezeigt. Die Übersichtstabellen enthalten die R-S Objekte und ermöglichen ihre Erstellung, Bearbeitung und andere Aktionen.

Übersichtstabellen werden je nach Objekt in einer zweistufigen Tabelle dargestellt. Die erste Ebene ist immer das Parent-Element z. B. das Schema. Die zweite Ebene zeigt die dazugehörigen Child-Elemente z. B. Schema-Versionen.

Der Detail-Bereich ist der Bereich des R-S, in dem Sie arbeiten werden.

## Suchfunktion

An vielen Stellen im R-S können Sie Übersichten und Tabellen durchsuchen. Die Suchfunktion ist dabei immer gleich aufgebaut und funktioniert nach demselben Prinzip. Hier sehen Sie ein exemplarisches Szenario für die Suchfunktion.

Schemas

		Geändert ▾		Q 1.0	0
▼	Briefe	1.0		⋮	
1	1.0 01.12.2020 - initiale Version	ACCEPTED	07.12.2020 15:52	08.12.2020 08:04	⋮
▼	Preis Anpassung	1.0		⋮	
1	1.0	EDIT	08.12.2020 07:23	08.01.2021 14:55	⋮
>	Rechnung	1.0		⋮	

Abbildung A

In Abbildung A wird die Suche exemplarisch in der Übersicht der Schemas dargestellt. Der eingegebene Suchtext (0) schränkt die Auswahl in der Übersicht ein, und zeigt Ihnen nur noch die Versionen (1), die zu diesem Suchtext passen.

Sie können in den Übersichten auch nach dem Status und dem Erst-/Änderungsdatum suchen:

Schemas

		Geändert ▾		Q EDIT
▼	Briefe	3.0_Copy2 3.0		⋮
	3.0_Copy2 08.12.2020 - Änderungen in der Fußzeile	EDIT	08.01.2021 14:42	25.01.2021 15:58
	3.0 08.12.2020 - Änderungen in der Fußzeile	EDIT	08.12.2020 07:24	25.01.2021 08:23
▼	Preis Anpassung	1.0		⋮
	1.0	EDIT	08.12.2020 07:23	08.01.2021 14:55
>	Rechnung	1.0		⋮

Abbildung B

In Abbildung C wird Ihnen die Suche in Ressourcen Dargestellt:

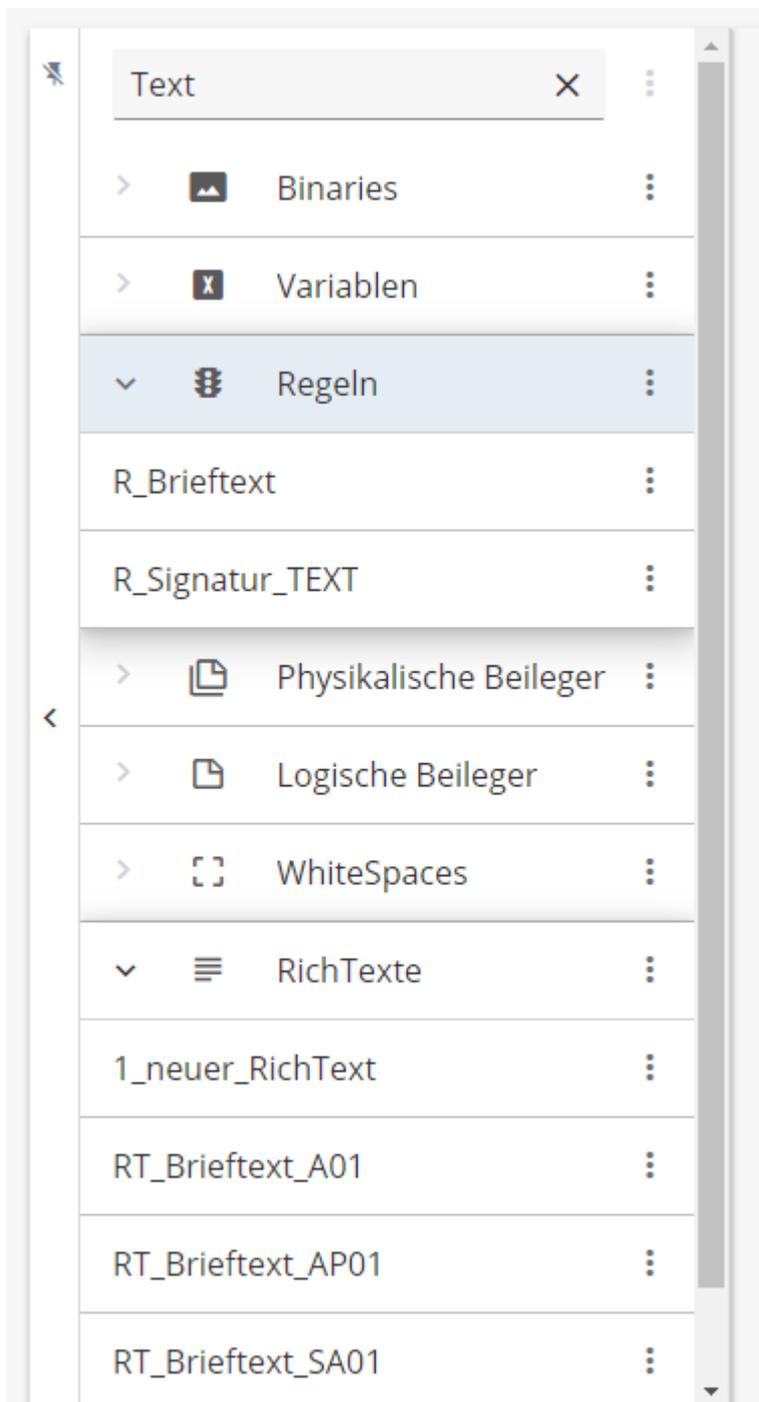


Abbildung C

Durch Eingabe eines Suchtextes schränken Sie die angezeigten Ressourcen ein. Sie können hier nach dem Namen einer Ressourcen suchen.

## Register

Briefe / 3.0
✎ Bearbeiten


---

Details
Bibliotheken
Ressourcen
Erweiterungen
Vorschau
Datenstrom-Variablen

Das Register gliedert den Inhalt eines R-S Objekts in thematisch geordnete Untergruppen. Es erlaubt Ihnen, bequem zwischen den Eigenschaftsseiten eines Objekts zu wechseln. Alle Eigenschaften und Inhalte eines Objekts sind durch das Register anwählbar. Ein Register beinhaltet immer mindestens 2 Registerkarten. Welche Registerkarten die verschiedenen R-S Objekte besitzen, erfahren Sie in den Kapiteln zum jeweiligen R-S Objekt, insbesondere zur Formular-Version ([hier](#)), zur Schema-Version ([hier](#)) und zu den Produktions-Paketen ([hier](#)).

## Bereich - Tab

Der Bereich Tab beinhaltet all Ihre geöffneten Objekte. Dazu zählen Transaktionsformular-, Schema- und Bibliotheksversionen sowie Materialien, Docxworld-Links, Produktions-Paket-Links und Produktions-Pakete.

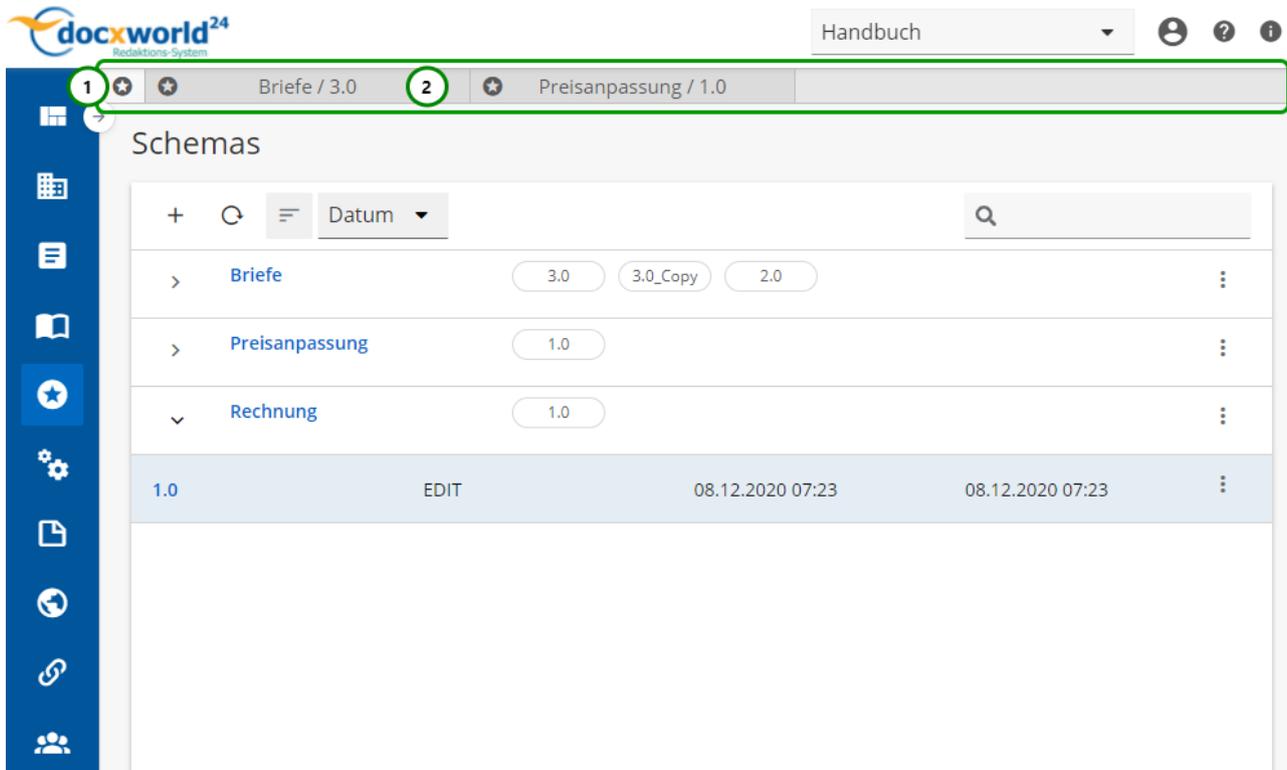


Abbildung A

Sie sehen:

- (1) Home-Button
- (2) Tabs

Sobald Sie ein Objekt öffnen, öffnet sich ein neuer Tab Abb. A (2) in dem das Objekt geöffnet wird.

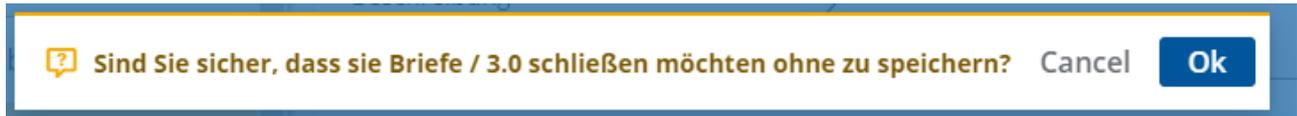


Abbildung B

Diesen aktiven Tab können Sie auch wieder schließen, indem Sie auf  [Abb. B \(1\)](#) klicken. Ungespeicherte Änderungen an dem Objekt erkennen Sie an dem \* .



Damit Sie nicht versehentlich ungespeicherte Änderungen verwerfen, müssen Sie beim Schließen eines Tabs mit ungespeicherten Änderungen eine zusätzliche Abfrage bestätigen.



Sie können jederzeit zwischen den einzelnen Tabs, dem Home-Button und anderen Navigationselementen hin- und herspringen.

Der Home-Button [Abb. A \(1\)](#) passt sich immer dem letzten Navigationspunkt indem Sie waren an. So gelangen Sie schnell in die vorher geöffnete Übersicht, um ggf. weitere Objekte zu öffnen. Sie können auch alle anderen Navigations-Punkte aus der Seiten-Navigation aufrufen.

## Bedienung des R-S

Nachdem Sie sich mit der Navigation des R-S vertraut gemacht haben, werden hier weitergehende Bedienelemente und Hinweise zur Bedienung des R-S gezeigt.

### **In diesem Kapitel werden folgende Aspekte behandelt:**

- Die **Eingabefelder** im R-S, insbesondere die Pflichtfelder.
- Wie sich Ihre Nutzerrechte auf Ihre **Berechtigungen** im R-S und darauf, welche Objekte Sie sehen können, auswirkt.
- Welche **Datentypen** das R-S unterstützt und wie Eingabefelder die Datentypen überprüfen.
- Welche **Hilfe-System** in das R-S integriert sind.

## Pflicht-Eingabefelder

Auf dieser Seite erläutern wir, welche Felder "Pflichtfelder" sind und somit ausgefüllt werden müssen.

Pflichtfelder für Objekte und Ressourcen werden immer mit \* gekennzeichnet.

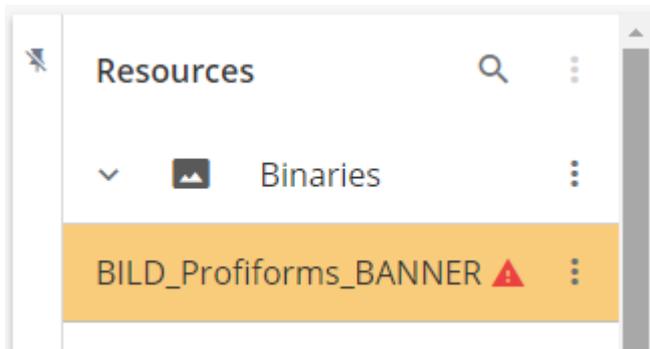
Haben Sie ein Pflichtfeld nicht ausgefüllt, sehen Sie eine Information unter dem Eingabefeld.

Beispiel einer Binären Ressource:

Name \*

Darf nicht leer sein

Verlassen Sie die Ressource ohne einen Namen zu vergeben, so sehen Sie auch in der Ressourcenübersicht, dass etwas an der Ressource fehlt:



In Eingabefeldern (bei Objektnamen, Objektbeschreibungen, etc.) nicht erlaubt sind führende / abschließende Leerzeichen.

## Sichtbarkeit und Berechtigungen

Wie kann es sein, dass Sie in der Benutzeroberfläche andere Daten oder Bedienelemente sehen als Ihr Kollege? Das liegt an den Berechtigungen und den damit zusammenhängenden Auswirkungen. (Mehr über Nutzerrechte und Berechtigungen finden Sie im entsprechenden **Kapitel** der Administrator-Dokumentation).

**Solche Auswirkungen sind, dass**

- Sie in Übersichtstabellen **keine Objekte** finden. Das liegt daran, dass Sie für READ (lesen) keine Berechtigung haben.
- Sie keine neue Version **erstellen** können  
Das liegt daran, dass Sie keine Berechtigung für CREATE (erstellen) haben.

Zugriff verweigert -> für Nutzer [4008] Aktivität [create]  
Zugriffspfad [/sr/Beta2Schema/test]

Ok

- Sie **nur bestimmte Objekte** in einer Übersichtstabelle sehen. Das liegt daran, dass Ihre Leserechte auf einen Filter gesetzt sind.  
Jetzt ist nur noch eine Auswahl an Objekten sichtbar, z.B. solche, die mit "F" beginnen.
- Sie keine **Änderungen** an einer Version vornehmen können  
Das liegt daran, dass Sie keine Berechtigung für **UPDATE** (ändern) haben.

Schreibzugriff nicht erlaubt

- Sie keine R-S Objekte **löschen** können. Das liegt daran, dass Sie keine Berechtigung für DELETE (löschen) haben.

Zugriff verweigert -> für Nutzer [4008] Aktivität [delete]  
Zugriffspfad [/sr/Beta2Schema/NewSchema]

Ok

- Sie Produktions-Pakete nicht in den PRODUCTION-Status versetzen können. Das liegt daran, dass Sie keine Berechtigung für PRODUCTION haben.
- Sie Basisobjekte nicht in den TEST-Status versetzen können. Das liegt daran, dass Sie keine TEST Berechtigungen haben.

## Hilfe-System

Die Haupthilfe über das R-S bildet diese **Online-Dokumentation**. Neben der Dokumentation über das R-S steht Ihnen direkt im R-S ein weiteres Hilfesystem zur Verfügung, welches Sie bei der Arbeit unterstützt.

Die kontextbezogene Hilfe unterstützt Sie beim Eingeben von **EVAL-Anweisungen**. Wenn Sie eine EVAL-Anweisung mit **\$(** einleiten, öffnet sich automatisch die kontextbezogene Hilfe Abb. A (1):

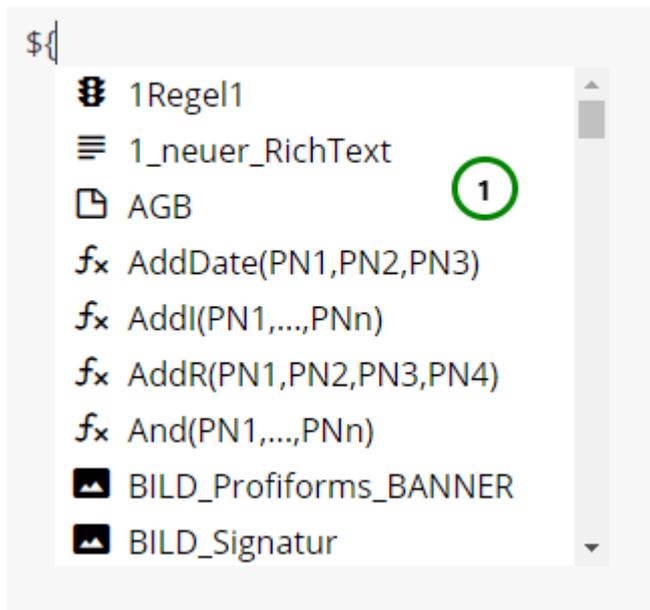


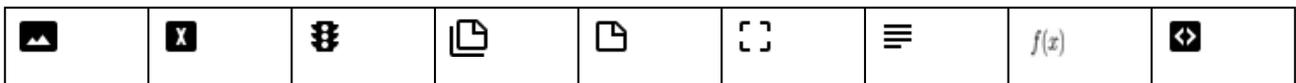
Abbildung A

Sie können die Hilfe ebenfalls über STRG + SPACE öffnen.

Ihnen wird eine Liste aller gültigen Befehle für die Anweisung angezeigt. Dazu gehören:

- **Ressourcen**
- **Funktionen**
- **Datenstromvariablen**

Das Symbol spezifiziert den **Typ** des Eintrags:



Wenn Sie bei geöffneter Kontexthilfe "Text" in die Anweisung eingeben, aktualisiert sich das Auswahlmü automatisch und zeigt nur Einträge an, die mit dem eingegebenen Term beginnen (*Groß- und Kleinschreibung wird nicht beachtet*):

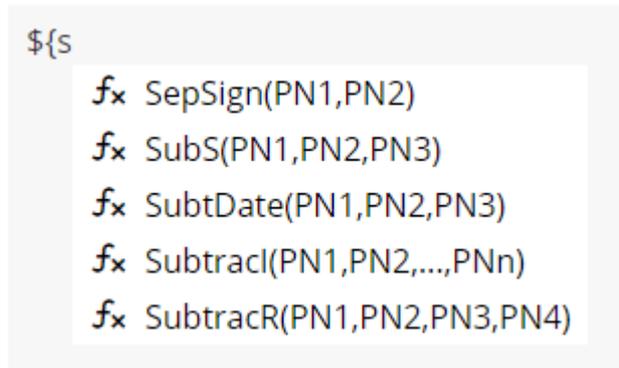


Abbildung B

Wählen Sie einen Eintrag aus, indem Sie darauf klicken. Der ausgewählte Eintrag (*hier: die Funktion **Subtractl** inkl. Parameter*) wurde übernommen und die Anweisung automatisch geschlossen:

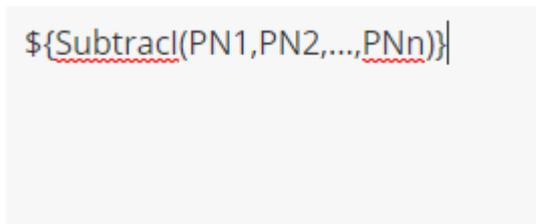


Abbildung C

 Mit ESC können Sie die kontextbezogene Hilfe schließen.

## Datentypen und Eingabe-Feld-Validierung

Einige Eingabefelder im R-S werden seitens des R-S Server und R-S Client inhaltlich auf Gültigkeit geprüft. Dies sind vor allem Eingabefelder, bei denen ein bestimmter **Datentyp** erwartet wird oder Felder, die Werte in einem bestimmten Format (z.B. Datumsformat) erwarten. Nachfolgend sind die wesentlichen Validierungstypen aufgeführt:

Daten-Typ	Beschreibung	Beispiele
RESOURCE NAME	<p>Namen von Ressourcen im R-S, wie zum Beispiel die Namen von Variablen oder Regeln in Schema-Versionen.</p> <p>Erlaubt sind folgende Zeichen:</p> <ul style="list-style-type: none"> <li>• a-z</li> <li>• A-Z</li> <li>• 0-9</li> <li>• - (Bindestrich)</li> <li>• _ (Unterstrich)</li> </ul>	<p>Logo_Firma_ab_02-2013 K2 2012-03-01-Regel_Fusszeile</p>
OBJECT NAME	<p>Namen von Objekten des R-S, wie zum Beispiel die Namen für logische oder physikalische Beileger</p> <p>Erlaubt sind folgende Zeichen:</p> <ul style="list-style-type: none"> <li>• a-z</li> <li>• A-Z</li> <li>• 0-9</li> <li>• - (Bindestrich)</li> <li>• _ (Unterstrich)</li> </ul>	<p>Flyer_Lufthansa_2013-01 Werbung-Weihnachten_2012</p>
PASSWORD	<p>Passwörter von Nutzern des R-S mit folgenden Restriktionen:</p> <ul style="list-style-type: none"> <li>• mindestens 8 Zeichen (die Mindestlänge ist im R-S für alle Nutzer einstellbar)</li> <li>• mindestens ein Großbuchstabe</li> <li>• mindestens ein Kleinbuchstabe</li> <li>• mindestens eine Ziffer</li> </ul> <p>Erlaubt sind folgende Zeichen:</p> <ul style="list-style-type: none"> <li>• a-z</li> <li>• A-Z</li> <li>• 0-9</li> </ul>	<p>Jonas23Tanz Ys4ypk9qw</p>
STRING	<p>Ist eine beliebige Zeichenkette aus Buchstaben, Ziffern und Sonderzeichen.</p> <p>Nähere Erläuterungen: <a href="#">Wikipedia 'Zeichenketten'</a></p>	<p>Dies ist eine Zeichenkette. Das ist eine 1.256,67 € Zeichenkette.</p>

Daten-Typ	Beschreibung	Beispiele
NUMBER	<p>Ganzzahliger numerischer Wert mit positivem oder negativem Vorzeichen. Das Vorzeichen kann vor oder nach dem ganzzahligen numerischen Wert stehen.</p> <p>⚠ Hinweise</p> <ul style="list-style-type: none"> <li>Maximaler Wertebereich: -2147483648 bis +2147483648</li> </ul>	<p>-45890 67- +1 87383+</p>
REAL	<p>Numerischer Wert mit Nachkommastellen mit positivem oder negativem Vorzeichen. Das Vorzeichen kann vor oder nach dem ganzzahligen Wert stehen.</p> <p>⚠ Hinweise</p> <ul style="list-style-type: none"> <li>Tausender Trennzeichen sind NICHT erlaubt !</li> <li>Das Trennzeichen für Nachkommastellen ist: '.'</li> <li>Es sind keine lokalisierten Schreibweisen möglich.</li> <li>Maximaler Wertebereich: -3,4E-4932 bis +3,4E-4932</li> </ul>	<p>-1234.45 +345.0 6272728.6727278- 1.0+</p>
BOOLEAN	<p>Ist eine 'Schaltvariable' zum Speichern von logischen Werten 'wahr' und 'falsch'.</p> <p>Nähere Erläuterungen: <a href="#">Wikipedia 'Boolsche Variable'</a></p>	<p>Werte, die ein 'wahr' erzeugen:</p> <ul style="list-style-type: none"> <li>- 1</li> <li>- true oder TRUE</li> <li>- yes oder YES</li> </ul> <p>Werte, die ein 'falsch' erzeugen:</p> <ul style="list-style-type: none"> <li>- 0</li> <li>- false oder FALSE</li> <li>- no oder NO</li> </ul>
DATETIME	<p>Datums-/Zeitformat nach der Norm ISO 8601, wobei die Angabe von Wochen, Wochentagen und Tagen im Jahr nicht unterstützt wird.</p> <p>Nähere Erläuterungen: <a href="#">Wikipedia 'Norm ISO 8601'</a></p>	<p>2011-12-31 2011-12-31T16:47Z</p>

Daten-Typ	Beschreibung	Beispiele
POSITION	<p>Beschreibt die X oder Y Position/Koordinaten eines Objektes in Millimetern.</p> <p><b>!</b> Hinweise</p> <ul style="list-style-type: none"> <li>• Tausender Trennzeichen sind NICHT erlaubt !</li> <li>• Das Trennzeichen für Nachkommastellen ist: '.'</li> <li>• es sind maximal 4 Nachkommastellen erlaubt</li> <li>• Maximaler Wertebereich: 0 bis +3,4E-4932</li> </ul>	<p>0 345.0 6272728.6727</p>
DIMENSION	<p>Beschreibt die Dimension von Objekten wie z.B. Höhe, Breite oder Stärke (Dicke).</p> <p><b>!</b> Hinweise</p> <ul style="list-style-type: none"> <li>• Tausender Trennzeichen sind NICHT erlaubt !</li> <li>• Das Trennzeichen für Nachkommastellen ist: '.'</li> <li>• es sind maximal 3 Nachkommastellen erlaubt</li> <li>• Maximaler Wertebereich: 0 bis +3,4E-4932</li> </ul>	<p>0 345.0 6272728.672</p>
WEIGHT	<p>Beschreibt das Gewicht von Objekten (z.B. Beileger).</p> <p><b>!</b> Hinweise</p> <ul style="list-style-type: none"> <li>• Tausender Trennzeichen sind NICHT erlaubt !</li> <li>• Das Trennzeichen für Nachkommastellen ist: '.'</li> <li>• es sind maximal 2 Nachkommastellen erlaubt</li> <li>• Maximaler Wertebereich: 0 bis +3,4E-4932</li> </ul>	<p>0 345.0 6272728.67</p>

Daten-Typ	Beschreibung	Beispiele
INTEGER	<p>Ganzzahliger numerischer Wert ohne Nachkommastellen.  Nähere Erläuterungen: <a href="#">Wikipedia 'Integer (Datentyp)'</a></p> <p>⚠ Hinweise</p> <ul style="list-style-type: none"> <li>Maximaler Wertebereich: 0 - 65535</li> </ul>	<p>0  345  65535</p>
PERCENT	<p>Ganzzahliger numerischer Wert ohne Nachkommastellen.  Beschreibt einen prozentualen Wert.</p> <p>⚠ Hinweise</p> <ul style="list-style-type: none"> <li>Maximaler Wertebereich: 0 - 100</li> </ul>	<p>0  12  100</p>

## Dashboard

Hier machen wir Sie mit dem Aufbau des Dashboards vertraut.  
Im Dashboard erhalten Sie einen Überblick der letzten Aktivitäten, welche im R-S durchgeführt wurden.

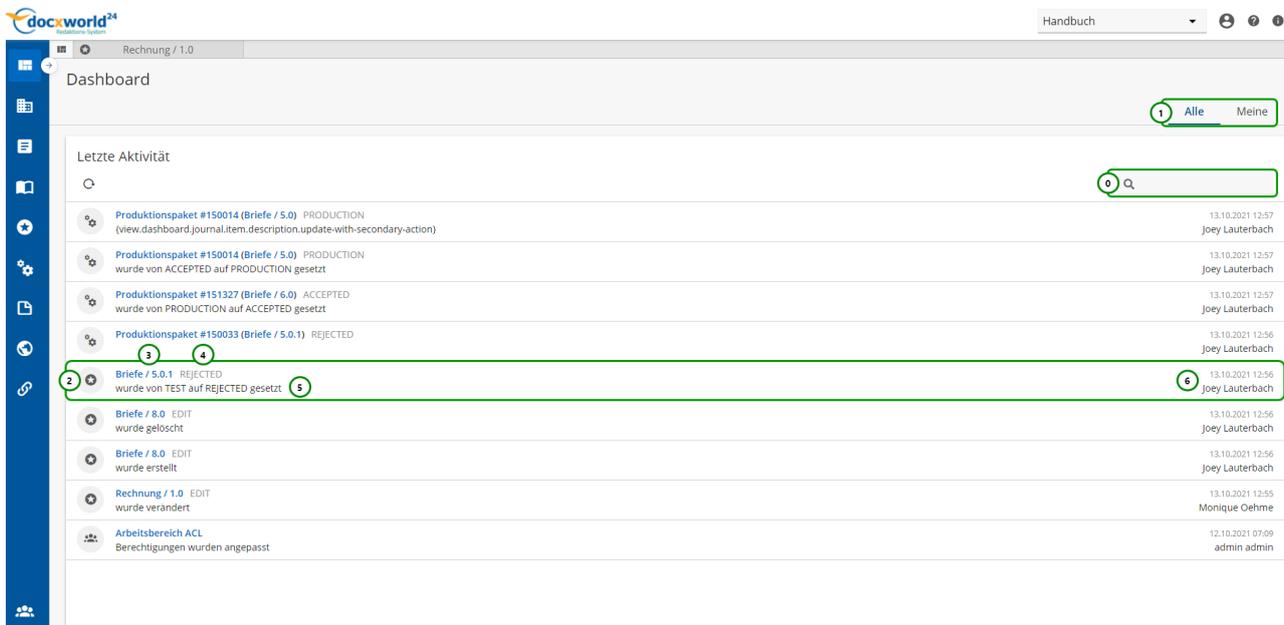


Abbildung A

Im oberen Teil der Ansicht haben Sie zu einem die Möglichkeit, nach Aktivitäten zu suchen  [Abb. A](#) (0) und zwischen "Alle" und "Meine" Aktivitäten (1) zu wechseln.

Anhand des Icons  (2) können Sie nachvollziehbar, welches Basisobjekt es betrifft. Sollten Sie noch nicht mit den Basisobjekten vertraut sein, gelangen Sie [hier](#) zu den Informationen.

Zusätzlich wird Ihnen der Name des Objektes mit Version (3), der aktuelle Status des Objektes (4), eine kurze Beschreibung (5) der Veränderung und wann und wer (6) diese vorgenommen hat, angezeigt.

## Arbeiten mit Transaktions-Formularen

Ihnen wurden in den vorigen Kapiteln die theoretischen Grundlagen, die Bedienung und Navigation und die Handhabung des R-S vermittelt. In den folgenden Kapiteln lernen Sie, wie sie mit den verschiedenen R-S Objekten arbeiten. Als erstes werden die Transaktions-Formulare behandelt, da sich diese am Anfang des R-S Workflows befinden. Um zu den Übersichtstabellen des Transaktions-Formulars zu gelangen, wechseln Sie in Navigationspunkt  Transaktions-Formulare.

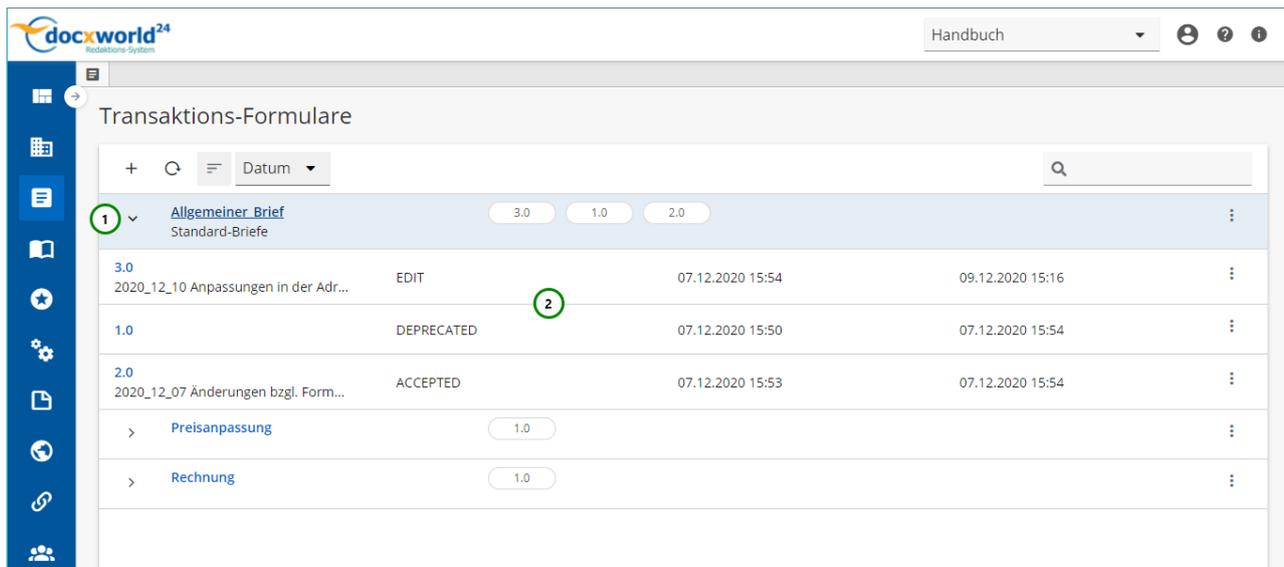


Abbildung A

Transaktions-Formulare enthalten die Rohdokumente, welche Sie später mit Schemas gestalten können. Die Dokumente in Transaktions-Formularen werden durch den externen Formular-Designer fest vorgegeben und können im R-S nicht bearbeitet, sondern nur so gestaltet werden, wie es das Transaktions-Formular erlaubt. Sobald Sie ein Formular erstellt haben, können (und müssen) Sie eine Version des Formulars anlegen. Mit diesen Formular-Versionen können Sie dann arbeiten. Falls Sie den Abschnitt über Transaktions-Formulare noch nicht gelesen haben, tun Sie dies bitte [hier](#), bevor Sie beginnen, mit Formularen zu arbeiten.

Allgemeiner Brief Standard-Briefe		3.0	1.0	2.0	
3.0	2020_12_10 Anpassungen in der Adr...	EDIT	07.12.2020 15:54	09.12.2020 15:16	
1.0		DEPRECATED	07.12.2020 15:50	07.12.2020 15:54	
2.0	2020_12_07 Änderungen bzgl. Form...	ACCEPTED	07.12.2020 15:53	07.12.2020 15:54	
>	Preis Anpassung	1.0			
>	Rechnung	1.0			

Abbildung B

Sie sehen in **Abb. B**:

- (0) Die Suche, in der Sie nach Transaktions-Formular-Versionen suchen können.
- (1) Name des Transaktions-Formulars inklusive Beschreibung.
- (2) Versions-Nummer des Transaktions-Formulars inklusive Beschreibung.
- (3) Den aktuellen Status der Transaktions-Formular-Version.
- (4) Das Erstelldatum der Transaktions-Formular-Version.
- (5) Das letzte Änderungsdatum der Transaktions-Formular-Version.
- (6) Das Kontextmenü der ausgewählten Transaktions-Formular-Version.
- (7) Das Kontextmenü des ausgewählten Transaktions-Formulars.
- (8) Quicklinks zu den letzten drei verwendeten Transaktions-Formular-Versionen des Transaktions-Formulars.

## Formular erstellen, ändern und löschen

In der Formular-Übersichtstabelle stehen Ihnen folgende Aktionen zur Verfügung, um ein Formular neu zu erstellen oder ein bereits vorhandenes Formular zu bearbeiten:

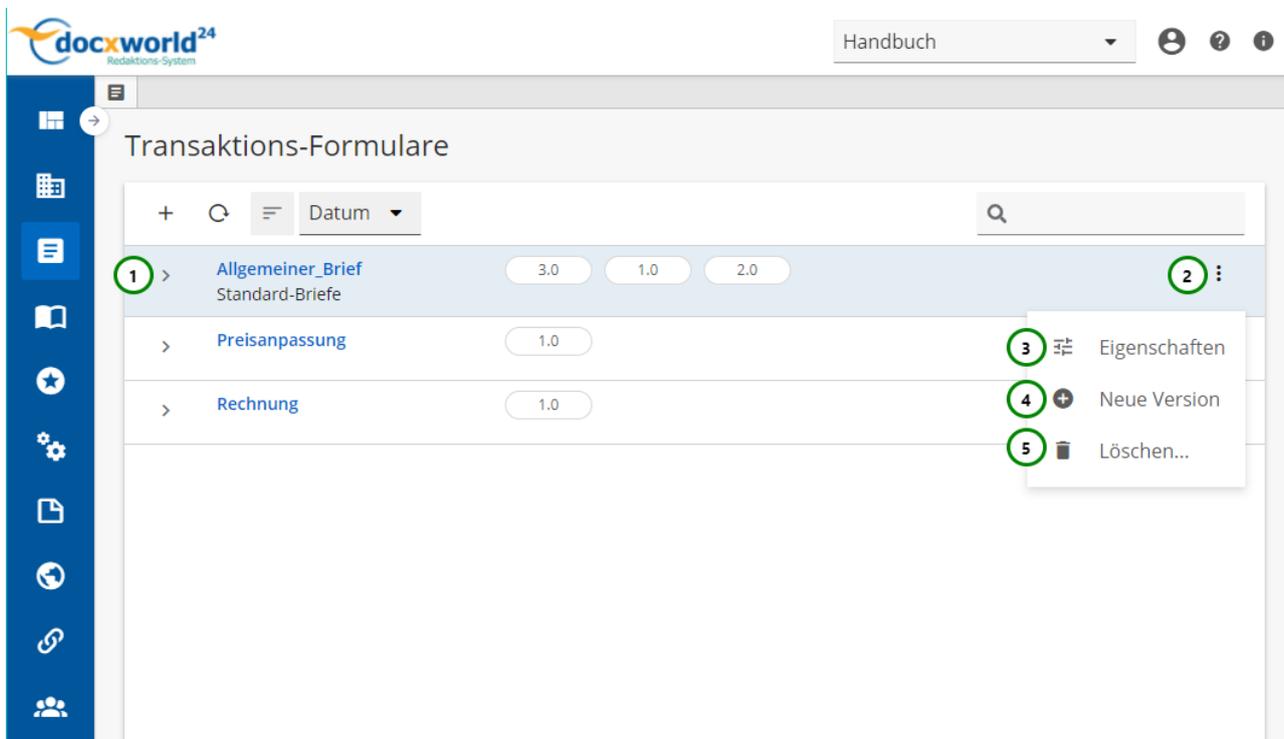


Abbildung A

Um ein Formular zu erstellen, klicken Sie bitte auf **+** Abb. A (1). Im weiteren öffnet sich ein Popup-Fenster, in welchem Sie den Namen und die Beschreibung erstellen können (*siehe Abb. B*).

Mit Abb. A (2) öffnen Sie das **:** Kontextmenü. Hier stehen Ihnen **≡** Eigenschaften (3), **+** Neue Version (4), **🗑️** Löschen (5) und **⬇️** Importieren (6) zur Auswahl.

Möchten Sie Änderungen vornehmen, wählen Sie das **:** Kontextmenü (2) des gewünschten Formulars in der Übersicht aus und klicken Sie anschließend auf **≡** (3). Auch hier öffnet sich ein Popup-Fenster, in welchem Sie das Formular aktualisieren können (*siehe Abb. B*).

Abbildung B

**Sie können dem Formular:**

- einen **Namen** geben.  
Beim Erstellen müssen Sie in das Feld etwas eingeben, da das Formular noch keinen Namen hat. Beim Ändern muss das Feld nicht zwingend geändert werden.
- eine **Beschreibung** geben.  
Die Beschreibung ist optional. Sie wird z.B. in Übersichten angezeigt.

Um das Formular zu erstellen oder die Änderungen zu speichern, verwenden Sie Ok *Abb. B*. Um das Formular zu verwerfen oder die Änderungen zu verwerfen, klicken Sie bitte auf Abbrechen *Abb. B*.

Nachdem Sie ein Transaktions-Formular erstellt haben, müssen Sie eine Version des Formulars anlegen.

## Formular-Version

Hier sehen Sie die Übersichtstabelle, mit der Sie Formular-Versionen erstellen, anschauen, ändern, löschen, kopieren, für sie einen Status setzen und sie filtern können.

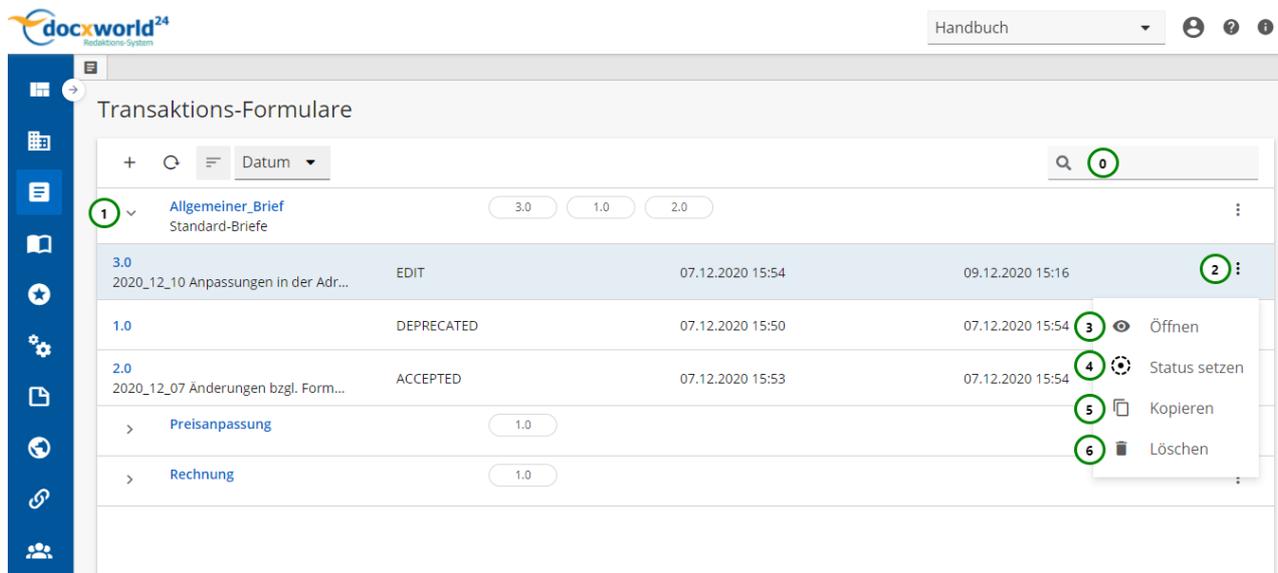


Abbildung A

### Sie sehen:

- (0) Die Suchfunktion , um die Auswahl einzuschränken.
- (1) Mit Hilfe von  können Sie Formulare aufklappen bzw.  zuklappen.
- (2) Das Kontextmenü .
- (3) Öffnen , um die Formular-Version zu öffnen.
- (4) Status setzen , um den Workflow der Formular-Version zu ändern.
- (5) Kopieren , um eine Kopie der Formular-Version mit einer neuen Versionsnummer zu erstellen.
- (6) Löschen , um die Formular-Version zu löschen.
- (7) Exportieren , um die Formular-Version als Zip zu exportieren.

Durch einen Klick auf  (1) öffnen Sie die Versionsübersicht zu dem ausgewählten Formular. Nachdem Sie ein Formular neu erstellt haben, müssen Sie eine Version für dieses Formular anlegen (siehe [hier](#)  *Neue Version (4)*), mit der Sie dann arbeiten können. Diese Formular-Version können Sie dann in einem Schema verwenden (*Mehr Informationen über Versionierung und den R-S Workflow finden Sie [hier](#).*). Außerdem können Sie die Schema-Versionen-Liste mit Hilfe von  (0) durchsuchen. Ausführliche Informationen zur Suchfunktion finden Sie [hier](#).

## Formular-Version erstellen, ändern und löschen

In der Formular-Versionen Übersichtstabelle stehen Ihnen folgende Aktionen zur Verfügung, um eine Formular-Version neu zu erstellen oder eine bereits vorhandene Formular-Version zu bearbeiten:

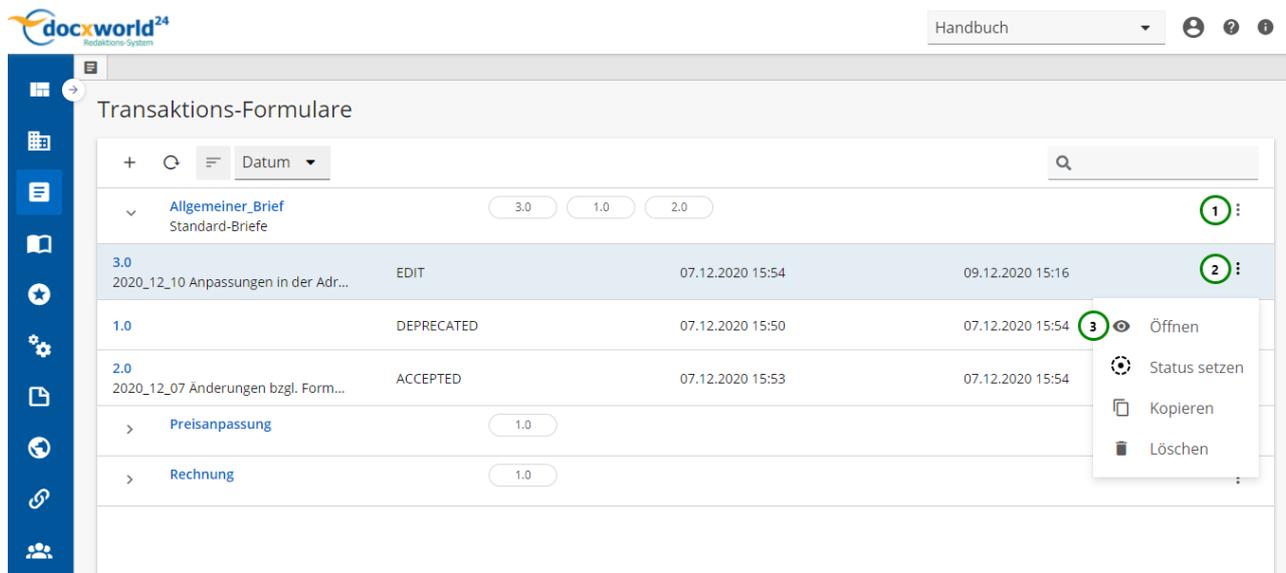


Abbildung A

Um eine Formular -Version zu erstellen, klicken Sie bitte auf **⋮** [Abb. A \(1\)](#) und anschließend auf **+** Neue Version (Abbildung dazu [hier](#)). Um eine bestehende Formular-Version zu öffnen, wählen Sie bitte **⋮** (2) der Formular-Version, die Sie öffnen möchten, aus und klicken Sie anschließend auf **👁** Öffnen (3).

Wenn Sie eine Formular-Version neu erstellen oder eine bereits vorhandene Formular-Version öffnen und bearbeiten möchten, öffnet sich folgende Übersicht:

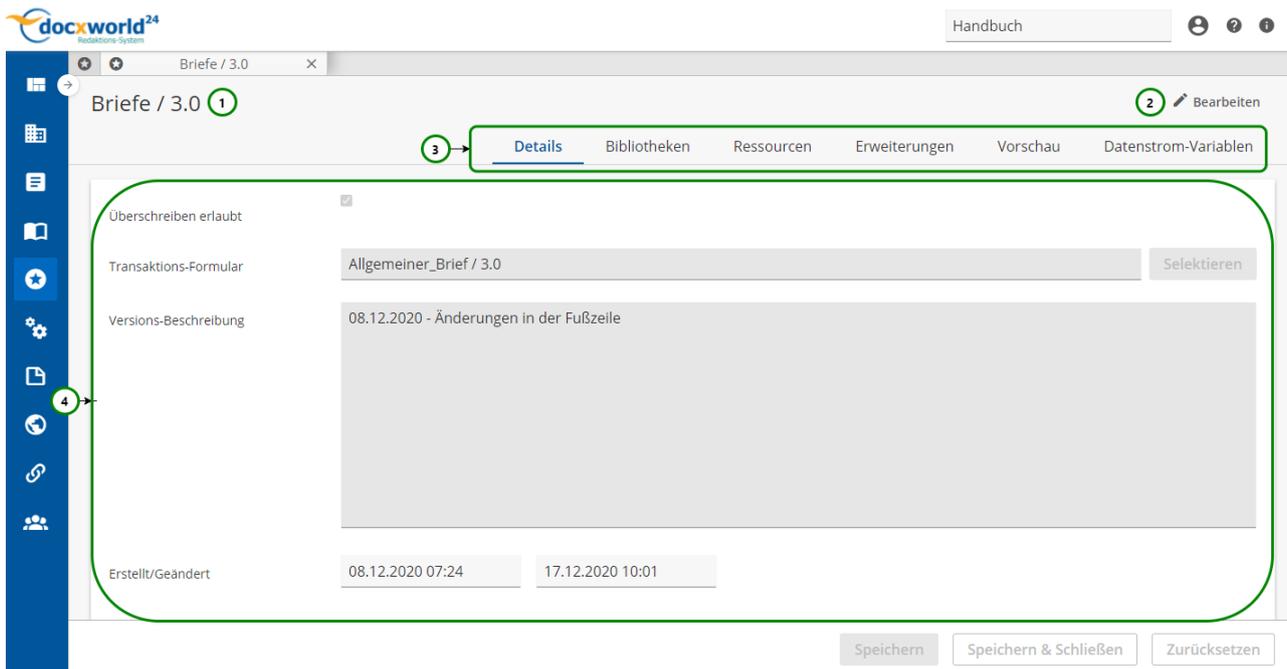


Abbildung B:

⚠ Sie befinden sich immer zuerst im **View-Modus**. Die Bearbeitung kann mit  Abb. B (2) begonnen werden.

Sie sehen in Abb. B :

- (1) Name / Version des Formulars
- (2) Bearbeitungsmodus aktivieren
- (3) Das Register der Formular-Version
- (4) Inhalt des ausgewählten Registers

## Formular-Version kopieren

In der Formular-Versionen-Übersichtstabelle steht Ihnen folgende Aktion zur Verfügung, um eine Formular-Version zu kopieren (duplizieren):

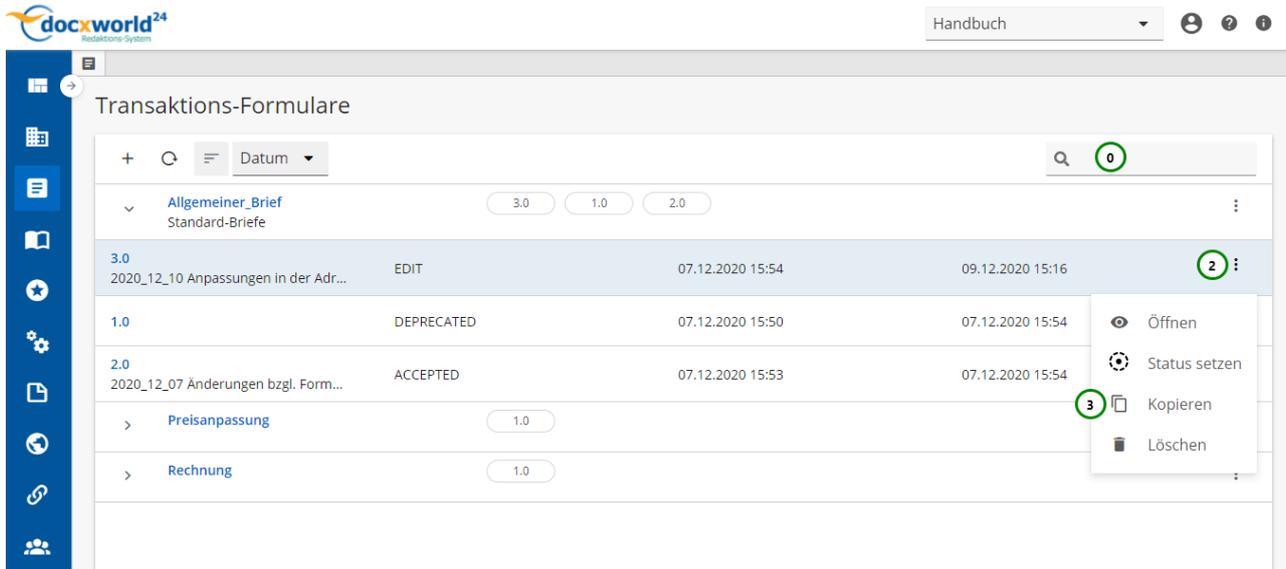


Abbildung A

Um eine Formular-Version zu kopieren, wählen Sie bitte die Formular-Version aus, die Sie kopieren möchten. Klicken Sie hierzu auf das  Kontextmenü Abb. A (1) und anschließend auf  Kopieren (2). Folgendes Fenster öffnet sich:



Abbildung B

Geben Sie für die neue Formular-Version eine Versionsnummer an Abb. B (1). Diese muss sich von der alten Version unterscheiden und darf für dieses Formular noch nicht existieren. Klicken Sie anschließend auf (2), um den Kopiervorgang abzuschließen oder auf (3), um ihn abubrechen. Sie finden die kopierte Formular-Version jetzt in der Formular-Versionen-Liste:

Allgemeiner_Brief Standard-Briefe		3.0_Copy	3.0	1.0	
3.0_Copy	2020_12_10 Anpassungen in der Adr...	EDIT	31.12.2020 10:30	31.12.2020 10:30	⋮
3.0	2020_12_10 Anpassungen in der Adr...	EDIT	07.12.2020 15:54	09.12.2020 15:16	⋮
1.0		DEPRECATED	07.12.2020 15:50	07.12.2020 15:54	⋮
2.0	2020_12_07 Änderungen bzgl. Form...	ACCEPTED	07.12.2020 15:53	07.12.2020 15:54	⋮

Abbildung C

 Wenn Sie eine Formular-Version kopieren, hat die Kopie immer den Status EDIT.

Eine kopierte Formular-Version besitzt alle Eigenschaften, die das Original zum Kopierzeitpunkt besaß. Ändert sich das Original oder die Kopie, hat dies aber keine Auswirkungen auf die andere Formular-Version, bei der Kopie handelt es sich um eine neues, eigenständiges Objekt. Da Kopien den Status EDIT haben, ist es durch Kopieren möglich, mit geschlossene Formular-Versionen weiterzuarbeiten, ohne sie komplett nachbilden zu müssen.

Sie können die Formular-Versionen-Liste mit Hilfe von  Abb. A (0) durchsuchen. Ausführliche Informationen zur Suchfunktion finden Sie [hier](#).

## Status einer Formular-Version setzen

In der Formular-Versionen-Übersichtstabelle steht Ihnen folgende Aktion zur Verfügung, um den Status einer Formular-Version zu ändern:

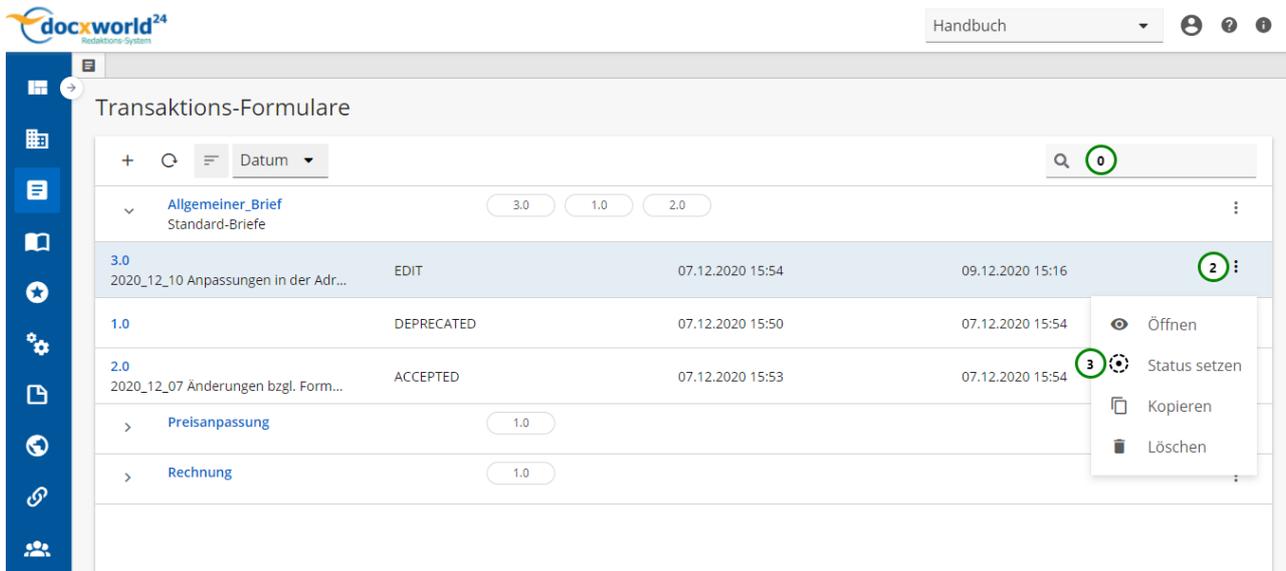


Abbildung A

Formulare sind wie **Bibliotheken**, **Schemas** oder **Produktions-Pakete** ein Teil des R-S Status-Workflows. Sie sind deshalb der Versionierung unterworfen und besitzen einen **Status**. Lesen Sie dazu bitte den Abschnitt über den **R-S Workflow**.

Mit **Abb. A** (1) öffnen Sie das Kontextmenü der ausgewählte Schema-Version. Wählen Sie hier "Status setzen" (2). Folgendes Fenster öffnet sich:

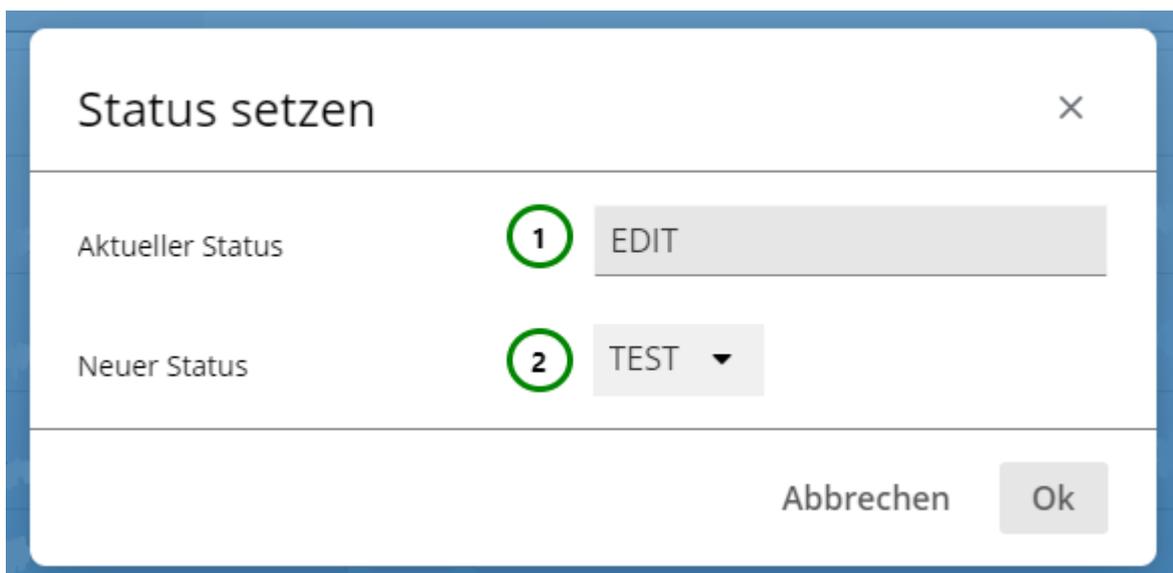


Abbildung B

⚠ Der Status kann nur verändert werden, wenn kein Benutzer im Bearbeitungsmodus der ausgewählten Formular-Version ist.

Sie sehen **Abb. B**:

- den aktuellen **Status** der Formular-Version
- den neuen **Status** der Formular-Version.

Wählen Sie in **Abb. B (2)** den neuen Status der Formular-Version. Die Auswahl in **Abb. B (2)** wird durch den aktuellen Status bestimmt:

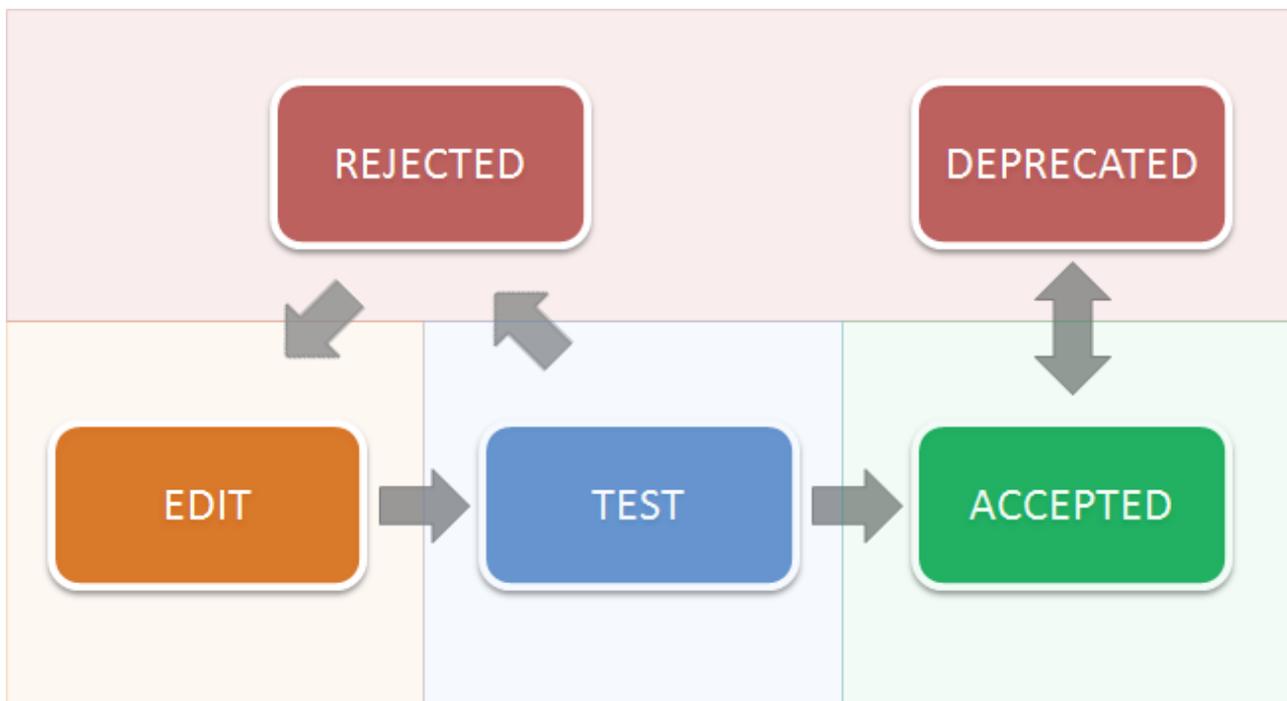


Abbildung C

Klicken Sie auf Ok **Abb. B**, um den neuen Status zu setzen oder auf Abbrechen **Abb. B**, um den aktuellen Status beizubehalten. Der Status der Formular-Version ist jetzt geändert. Beachten Sie bitte, dass Sie vom ACCEPTED-Status nicht wieder zurück in den EDIT-Status wechseln können. Wechseln Sie daher den Status von TEST zu ACCEPTED erst dann, wenn die Formular-Version wirklich zum Release freigegeben werden soll.

⚠ Für eine genaue Beschreibung des Status-Workflows bei Formular-Versionen lesen Sie bitte unbedingt den **Abschnitt über den R-S Workflow**.

## Registerkarten bei Formular-Versionen

Bei den Formular-Versionen gibt es folgende **Registerkarten**:

Details	Seiten	Testdaten	Datenstrom-Variablen	Beileger	WhiteSpaces	Schriftarten	Umgebungen
---------	--------	-----------	----------------------	----------	-------------	--------------	------------

Sie beinhalten die Eigenschaften und Inhalte der Formular-Version.

### Kurzbeschreibung der Registerkarten und ihrer Inhalte:

Registerkarte	Beschreibung
<b>Details</b>	Sie beinhaltet allgemeine Informationen über die Formular-Version und ermöglicht das Hochladen des Import-Pakets, das die Rohdokumente und Daten für das Formular enthält.
<b>Seiten</b>	Hier sehen Sie das Design der Formular-Seiten (der Rohdokumente aus dem Import-Paket).
<b>Testdaten</b>	Hier sehen Sie, welche Testdaten die Formular-Version zur Verfügung stellt, um später eine Testvorschau zu generieren.
<b>Datenstrom-Variablen</b>	Zeigt die Datenstrom-Variablen, die die Formular-Version besitzt.
<b>Beileger</b>	Zeigt die durch das Formular gelieferten Materialien und welche Beileger das Formular erlaubt.
<b>WhiteSpaces</b>	Zeigt die Formular-Teile für WhiteSpaces und ob sie erlaubt sind.
<b>Schriftarten</b>	Hier sehen Sie, welche Schriftarten und Schrifttypen das Formular erlaubt.
<b>Umgebungen</b>	Hier sehen Sie, welche Software-Versionen das Formular im <b>Binär-Paket</b> voraussetzt.

## Registerkarte - Details in Formular-Versionen

Die Registerkarte Details enthält allgemeine Informationen über die Formular-Version und ermöglicht das Hochladen eines Import-Pakets. Wenn Sie eine Formular-Version neu erstellen oder öffnen befinden Sie sich automatisch in der Registerkarte Details.

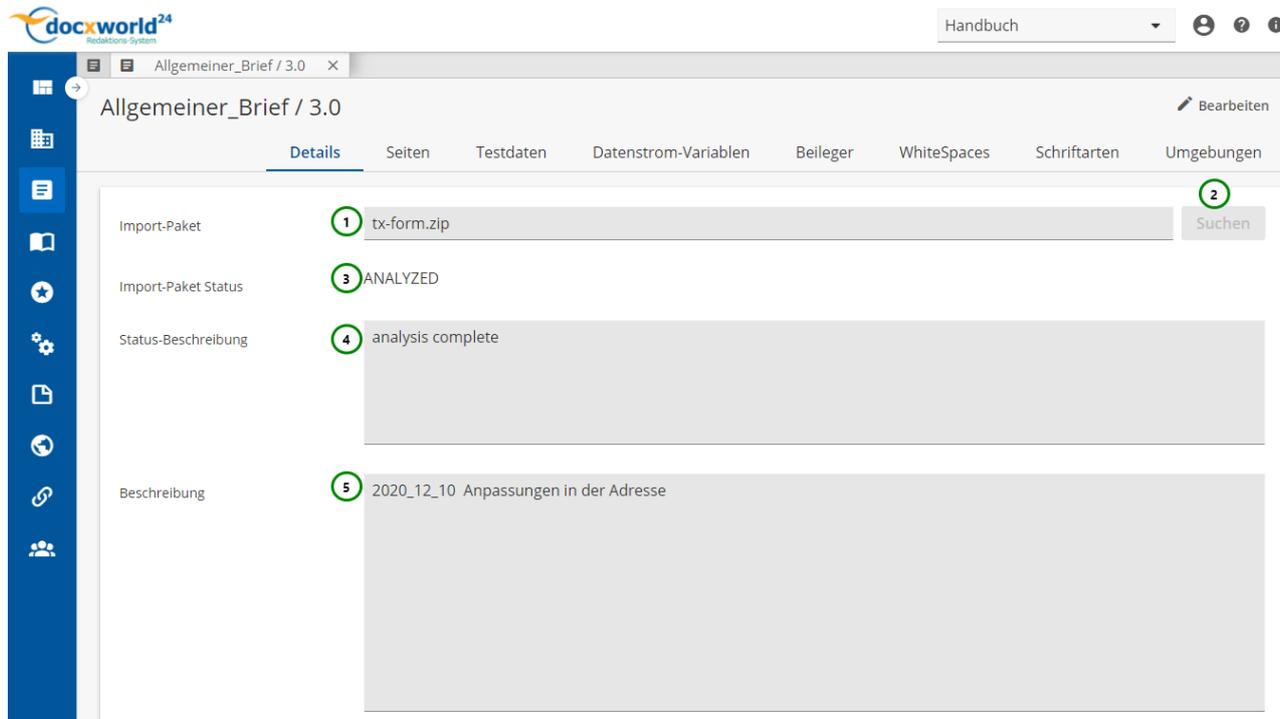


Abbildung A

### Sie können:

- ★ Ein **Import-Paket** hochladen [Abb. A \(2\)](#).  
Das Import-Paket beinhaltet die Dokument-Rohdaten, die z.B. mit Hilfe des Entwickler-Clients erstellt wurden. Sie bilden die Basis des Transaktions-Formulars. Ohne gültiges Import-Paket enthält das Formular keine Daten. Mehr über Import-Pakete erfahren Sie [hier](#).  
Ist schon ein Import-Paket vorhanden, sehen Sie dieses unter (1).
- ★ Der Formular-Version eine **Beschreibung** geben (5).  
Die Beschreibung ist optional und wird z.B. in Auswahldialogen angezeigt.

In [Abb. A \(3\)](#) sehen Sie den Status des Import-Pakets (hier ANALYZED). Tritt bei dem Import-Vorgang ein Fehler auf, ist der Status ERROR. Unter der Status-Beschreibung (4) finden Sie dann genauere Informationen zum Status.

⚠ Zum Verändern der Inhalte müssen Sie vorher den Bearbeitungsmodus ( Bearbeiten) aktivieren (nur im Status Edit möglich).

## Import eines Formulars

Ein Transaktions-Formular benötigt ein Import-Paket, das die Daten (u.a. Rohdokumente), mit denen das Transaktions-Formular gefüllt werden soll, enthält. Transaktions-Formulare enthalten also eigentlich den Inhalt des mit dem DeveloperClient erstellten Import-Pakets. Import-Pakete werden in der Registerkarte Details hochgeladen. Wechseln Sie zu dieser Ansicht. Sie finden dort folgende Aktionsleiste:

The image shows a search bar with the label 'Import-Paket' on the left. The search input field contains the text 'tx-form.zip'. To the right of the input field is a small 'x' icon for clearing the search. Further right is a blue button with the white text 'Suchen'.

Um ein Formular zu importieren, klicken Sie bitte auf "Suche". Es öffnet sich der Windows-Explorer. Wählen Sie nun eine Datei aus, die Sie importieren möchten. Import-Pakete enthalten alle nötigen Rohdokumente, Daten und Informationen zur Generierung eines Transaktions-Formulars. Import-Pakete werden im DeveloperClient generiert, mit dem der Formular-Entwickler die Rohdokumente erstellen und weitere Konfigurationen und Inhalte für die Dokumente festlegen. Import-Pakete sind also die Schnittstelle zwischen dem Formular-Designer und dem R-S. Zur Erstellung von Dokumenten und zur Generierung eines Import-Pakets innerhalb des Entwickler Client wird der Adobe LifeCycle Designer genutzt.

Ist der Import des Paktes abgeschlossen und der Inhalt wurde erfolgreich überprüft, wechselt der Status des Import-Pakets zu *ANAYLZED*. Sonst sehen Sie in der Status-Beschreibung aufgetretene Ereignisse beim Ladevorgang.

Mit **X** können Sie das Import-Paket wieder entbinden. Beachten Sie jedoch, dass ein Formular ohne ein Import-Paket keinerlei Daten enthält! Sie können das Import-Paket auch wieder exportieren. Da Import-Pakete im ZIP-Format vorliegen, können Sie das exportierte Paket mit einem ZIP-Viewer öffnen.

Adobe XDP und OMS - Import-Paket

Verwenden Sie den Adobe LiveCycle-Designer als Formular-Design-Software und die Output-Management-Software von profiforms (**ReportWriter, Logistics** ...) zur Dokument-Aufbereitung, so erfolgt der Import des Transaktions-Formulars in Form einer ZIP-Datei. Diese ZIP-Datei ist das Import-Paket der Transaktions-Formulare für das R-S.

**Ein Import-Paket umfasst im wesentlichen folgende Bestandteile:**

- eine Beschreibungsdatei für das Import-Paket
- das Transaktions-Formular (XDP) selbst, sowie weitere benötigte Teil/Unter-Formulare
- alle für die Aufbereitung des Transaktions-Formulars notwendigen Fonts, Logos (welche vom Transaktions-Formular selbst verwendet werden)
- alle Basis-Konfigurationen des Transaktions-Formulars wie TCI's, INI's

Alle zu einem Import-Paket gehörenden Bestandteile werden nicht einzeln als Datei in das R-S importiert, sondern in Form einer ZIP-Datei, welche in Unterordnern die entsprechenden o.g. Dateien und Konfigurationen enthält. Der Name der ZIP-Datei ist für den Import in das R-S nicht maßgeblich, sollte aber aus Gründen der Übersichtlichkeit den Namen des Transaktions-Formulars enthalten. Werden Änderungen am Transaktions-Formular oder an dessen Konfigurationen etc. vorgenommen, so muss für diese Änderungen ein neues Import-Paket erstellt werden, welches im R-S als neue Version des Transaktions-Formulars importiert werden kann.

## Aufbau des ZIP-Import-Paketes

Generell sei vorausgeschickt, dass das Import-Paket zwingend nur das Unterverzeichnis **etc** zur Strukturierung der Dateien eines Transaktions-Formulars benötigt. Nutzer der Output-Management-Suite (Spooler) und des DeveloperClient haben sich in der Praxis jedoch so organisiert, dass die Dateien zu einem Transaktions-Formular thematisch in Unterverzeichnissen abgelegt werden. Im Folgenden sehen Sie einen typischen Verzeichnis-Aufbau eines ZIP-Import-Paketes als Beispiel:

Verzeichnisname	Inhalt des Verzeichnisses
etc	Das Verzeichnis enthält die Paket-Beschreibungs-Datei <b>rs_package_configuration.xml</b> des Transaktions-Formulars, welche für den erfolgreichen Import eines Import-Paketes in das R-S unbedingt notwendig ist. Weiterhin werden im etc-Verzeichnis alle Konfigurationsdateien für die Programme aus dem bin-Verzeichnis abgelegt. Dazu zählen typischerweise TCI-Dateien und INI-Dateien.  Detaillierte Informationen zum Aufbau der Paket-Beschreibungs-Datei finden Sie hier: <a href="#">Aufbau der Paket-Beschreibungs-Datei</a>
fonts	Das Verzeichnis enthält alle zur Aufbereitung des Transaktions-Formulars notwendigen Font-Dateien.
forms	Alle Formular-Dateien des Adobe LiveCycle Designers (XDP) sowie Overlays, Anhänge z.B. im PDF-Format werden in diesem Verzeichnis abgelegt.
logos	Die für die Aufbereitung des Transaktions-Formulars selbst benötigten grafischen Elemente wie Logos, Hintergründe (TIFF, BMP, JPG ...) sind in diesem Verzeichnis abgelegt
testdata	Sollen für die Druckvorschau im R-S Testdaten zur Verfügung gestellt werden, so werden diese im testdata-Verzeichnis abgelegt



### Bitte beachten Sie:

- Dass die Verzeichnisse mit dem Namen rs und rw im Standard des R-S verwendet werden und als Verzeichnis-Name im Import-Paket nicht benutzt werden dürfen.
- Dass Dateien / Verzeichnisse innerhalb des ZIP-Import-Containers keine Umlaute / Sonderzeichen enthalten dürfen (Optimal: a-z,A-Z,\_)



### Achten Sie bei der Erstellung der ZIP-Datei des Import-Paketes auf folgende Hinweise:

- Vermeiden Sie bei der Vergabe der Verzeichnisnamen Leerzeichen.
- Achten Sie darauf, dass die Transaktions-Formulare auf mehreren Betriebssystemen ausgeführt werden können
- Dass die genannten Verzeichnisnamen beim ZIP-Vorgang nicht wiederum in einem Verzeichnis enthalten sind
- Dass Das ZIP-Programm die ZIP-Spezifikation erfüllt (siehe WIKIPEDIA: <http://www.iana.org/assignments/media-types/application/zip>)

## Aufbau der Paket-Beschreibungs-Datei

Liest das R-S beim Import eines Transaktions-Formulars das ZIP-Import-Paket, so wird im Unterverzeichnis **etc** nach einer XML-Datei mit dem Namen **rs\_package\_configuration.xml** gesucht. Diese XML-Datei beschreibt den Inhalt und den Aufbau des ZIP-Import-Paketes genauer und ermöglicht eine inhaltliche Prüfung.

### In der Paket-Beschreibungs-Datei sind u.a. folgende Informationen abgelegt:

- Name des Transaktions-Formulars.
- Informationen zum Aufruf der Laufzeitumgebung zur Generierung von Dokumenten auf der Basis dieses Transaktions-Formulars (pro Betriebssystem).
- Den Dateinamen des Transaktions-Formulars von Adobe (welches nach LayoutBereichen analysiert wird).
- Materialinformationen zu den verwendeten Papieren des Transaktionsformulars.
- Hintergrundbilder für die Anzeige der Seiten des Transaktions-Formulars im R-S.
- Referenzen auf Testdaten, welche dem Nutzer des R-S für Dokumentvorschau angeboten werden
- Namen von Variablen aus dem Datenstrom des Transaktions-Formulars und exemplarische Werte dazu.  
**Hinweis:** Diese Variablen werden im R-S als sogenannte DS-Variablen angeboten, welche R-S Nutzer zum Beispiel in Texten als Platzhalter oder bei Regeln als Suchwert verwenden können. Aber auch ohne explizite Deklaration der DS-Variablen in der Paket-Beschreibungs-Datei kann mit dieser Syntax/Notation auf globale Datenelemente des Datenstroms zugegriffen werden.
- Definitionen ob logische Beileger genutzt werden und ob diese auf dem Papier des Transaktions-Formular gedruckt werden müssen.
- Definitionen ob physikalische Beileger genutzt werden können.
- Definition ob logische/physikalische Beileger ggf. einen Sprung im Porto verursachen können.
- Definitionen ob WhiteSpace-Bereiche auf den Dokumenten genutzt werden können.
- Wie viel Platz maximal auf den Seiten für Werbung zur Verfügung steht.
- Auf welchen Teilen des Transaktions-Formulars WhiteSpace platziert werden kann.

Die Paket-Beschreibungs-Datei **rs\_package\_configuration.xml** wird mit einem geeigneten XML-Editor durch den Designer des Transaktions-Formulars manuell erstellt. Beispiel einer Paket-Beschreibungs-Datei:

```

<?pte APIVersion="0.0.02"?>
<package type="transaction" name="V-0002" description="fly-ticket invoices 01.01.2011"
version="1.1">
<configurationSet>
  <runtimeEnvironments>
    <runtimeEnvironment platform="Windows">
      <program name="ReportWriter" version="5.6.*"></program>
      <cmdLine>-alp ${home}/my_special_logos</cmdLine>
      <programResult>
        <log>*.log,*.docref</log>
        <preview>*.pdf</preview>
        <result type="ReturnCode">0</result>
      </programResult>
    </runtimeEnvironment>
    <runtimeEnvironment platform="Linux">
      <program name="ReportWriter" version="5.6.1014"></program>
      <cmdLine>-alp ${home}/my_special_logos</cmdLine>
      <programResult>
        <log>process.log</log>
        <preview>*.pdf</preview>
        <result type="ReturnCode">0</result>
      </programResult>
    </runtimeEnvironment>
  </runtimeEnvironments>
  <transactionForm>${home}/forms/M-0044-V-0002.xdp</transactionForm>
  <transactionFormMaterials>
    <material width="210mm" height=297mm" thickness="0.5mm" weight="1.2g" name="A4LOG
0" description="Einzelblatt weiss mit Logo"/>
    <material width="210mm" height=297mm" thickness="0.8mm" weight="3.6g" name="ZSPAP
ER" description="Einzelblatt Zahlschein u. Perforation"/>
  </transactionFormMaterials>
  <transactionFormPageBackgrounds>
    <transactionFormPageBackground pageName="ErsteSeite">${home}/backgrounds/
firstpage.gif</transactionFormPageBackground>
    <transactionFormPageBackground pageName="Folgeseiten">${home}/backgrounds/
nextpage.gif</transactionFormPageBackground>
  </transactionFormPageBackgrounds>
  <fontDef>
    <location>${home}/etc/fonts.ini</location>
  </fontDef>
  <testDataSet>
    <testData>

```

```

    <name>Fly-Testabrechnung</name>
    <description>5 Dokumente in allen Varianten</description>
    <value>${home}/testdata/fly.xml</value>
  </testData>
</testDataSet>
<inputVariableSet>
  <inputVariable>
    <name>PARTNER_ID</name>
    <description>System-ID des Werbe-Partners.</description>
    <values>
      <value>
        <name>RYANA</name>
        <description>ID for Airline Ryan Air</description>
      </value>
      <value>
        <name>LH</name>
        <description>ID for Lufthansa</description>
      </value>
      <value>
        <name>FLYDBA</name>
        <description>ID for DBA</description>
      </value>
    </values>
  </inputVariable>
  <inputVariable>
    <name>NL_KEY</name>
    <description>System-ID der Länderniederlassung. Werte: '01 - Deutschland' '02 -
    Österreich'</description>
  </inputVariable>
  <inputVariable>
    <name>SUM_VALUE</name>
    <description>Rechnungs-Gesamtsumme</description>
  </inputVariable>
</inputVariableSet>
</configurationSet>
<supplement>
  <logicalSupplement>
    <allowed>true</allowed>
    <useOnlyTransactionFormPaper>true</useOnlyTransactionFormPaper>
  </logicalSupplement>
  <physicalSupplement>
    <allowed>true</allowed>
  </physicalSupplement>

```

```

</supplement>
<shipment>
  <postage>
    <optionalSupplementCanExceedPostage>true</optionalSupplementCanExceedPostage>
  </postage>
</shipment>
<whitespace>
  <allowed>true</allowed>
  <maxSpace>50</maxSpace>
  <overflow>ALLOWTOCREATEBACKPAGE</overflow>
  <!--NOTALLOWED # standard value -->
  <!--ALLOWTOCREATEBACKPAGE # create whitespace-content on empty backpages in case of
duplex-printing -->
  <partSet>
    <part>
      <name>Rechnung</name>
      <description>Rechnung</description>
    </part>
    <part>
      <name>Dunning</name>
      <description>Dunning</description>
    </part>
  </partSet>
</whitespace>
</package>

```

**i** Bei fehlerhafter XML-Syntax oder fehlerhaft definierten Objekt-Referenzen bricht der Import des Paketes im R-S mit Fehlermeldungen ab.

**i** Die Paket-Beschreibungs-Datei ist Teil des ZIP-Import-Paketes. Bei der Analyse des ZIP-Import-Paketes wird dieses temporär entpackt und auf notwendige Bestandteile untersucht. Zur Untersuchung zählen u.a. die Analyse der Fonts, die mitgegebenen Testdaten etc. Um die Position dieser Informationen im Import-Paket variabel zu halten, können Pfade angegeben werden, in welchen das R-S bei der Analyse des Import-Paketes die entsprechenden Informationen findet. Der erste Teil des Pfades ist immer variabel/dynamisch und hängt von den technischen Installationsbedingungen des R-S ab. Damit die Nutzer des R-S mit diesen technischen Informationen nicht konfrontiert werden müssen, wird eine Variable \$home verwendet, welche den tatsächlichen Pfad des Import-Paketes nach dem Entpacken des ZIP-Import-Paketes beschreibt. Verwenden Sie \$home immer direkt vor den Angabe von relativen Pfaden zu Dateien/Ressourcen aus dem Import-Paket.

⚠ Input-Variablen (DS.\*) werden **NICHT** durch das Redaktions-System auf Korrektheit geprüft! Ist zum Beispiel eine DS-Variable falsch geschrieben, so wird dies erst durch die Output Management Produkte geprüft, welche die Variablen bei der Dokument-Generierung nutzen (Late Binding). So können diese Art von Fehlern erst bei der Ausführung der Preview-Funktion im Redaktions-System evaluiert werden.

Download einer Muster-Paket-Beschreibungs-Datei: [rs\\_package\\_configuration.xml](#)

## Layout-Bereiche im Adobe XDP

Um den Nutzern des R-S die Möglichkeit zu geben, einzelne Formular-Bereiche (Layout-Bereiche) mit dynamischen Inhalten füllen zu können, muss der Designer des Transaktions-Formulars diese Layout-Bereiche definieren. Die Definition der Layout-Bereiche erfolgt über speziell gekennzeichnete Objekte im Adobe LiveCycle-Designer (XDP).

### Die Definition von Layout-Bereichen erfolgt in folgenden Schritten:

- Definieren Sie mit den verantwortlichen Mitarbeitern die Layout-Bereiche auf den Transaktions-Formularen, welche zur Bearbeitung freigegeben werden sollen.
- Bestimmen Sie für jeden Layout-Bereich mit welchem Inhalt dieser gefüllt werden soll (TEXT oder BILD).
- Platzieren Sie für Layout-Bereiche mit TEXT als Inhalt ein Adobe LiveCycle-Objekt vom Typ 'Textfeld' auf dem Transaktions-Formular.
- Platzieren Sie für Layout-Bereiche mit BILD als Inhalt ein Adobe LiveCycle-Objekt vom Typ 'Bild' auf dem Transaktions-Formular.
- Definieren Sie die Position/Höhe/Breite für die entsprechenden Objekte.
- Klicken Sie nun auf jedes Feld, welches ein Layout-Bereich im R-S darstellt und wechseln Sie in die Ansicht 'XML-Quelle' im Adobe LiveCycle Designer.
- Ergänzen Sie nun im XML-Tag 'ui' folgende Zeile `<?omsTpl type="OVERLAYSTATIC" version="0.2"?>` (ein Beispiel sehen Sie unten).
- Geben Sie dem Feld über die Objekteigenschaften 'Bindung' einen Namen, so wie das Feld als Layout-Bereich im R-S bezeichnet werden soll.

#### Beispiel

```
<field name="PARTNER_LOGO" w="68mm" h="13mm" x="124mm" y="23mm">
  <ui>
    <?omsTpl type="OVERLAYSTATIC" version="0.2"?>
    <tpStaticOverlay/>
    <imageEdit/>
  </ui>
  <bind match="none"/>
</field>
```

### **Maßeinheiten der Layout-Bereiche**

Der Adobe Livecycle Designer unterstützt eine Vielzahl von Maßeinheiten bei der Eingabe von Objektpositionen, Abständen usw. Das R-S unterstützt beim Import und bei der Analyse von XDP-Dateien derzeit nur die Maßeinheit Millimeter ('mm'). Bitte stellen Sie sicher, dass in allen Elementen für das R-S (Layout-Bereiche) die Angaben in Millimetern erfolgt.

 Auf der grafischen Oberfläche des Adobe LiveCycle-Designers werden alle Angaben in der gewünschten Maßeinheit des Anwenders angezeigt. Intern (je nach Standard-Einstellung des XDP's) kann es jedoch vorkommen, dass die Werte in Inch o.ä. gespeichert werden. Prüfen Sie dies bitte in der XML-Ansicht des XDP-Formulars.

 Nutzen Sie vielfach die Layout-Bereiche im Adobe XDP-Format, so ist es von Vorteil sich die oft genutzten Feld-Typen für die Weiterverarbeitung durch das R-S als Adobe Objekt-Bibliothek zu erstellen.

### **Arbeiten mit Adobe Objekt-Bibliotheken**

Der Adobe LiveCycle Designer bietet die Möglichkeit, immer wieder benötigte Design-Objekte in einer Objekt-Bibliothek abzulegen. Benötigen Sie Layout-Bereiche für das R-S ständig in Ihrer Arbeit, so erstellen Sie ein R-S Objekt im Adobe LiveCycle Designer und legen dieses Element in der Objekt-Bibliothek 'Eigene' ab (Drag & Drop).

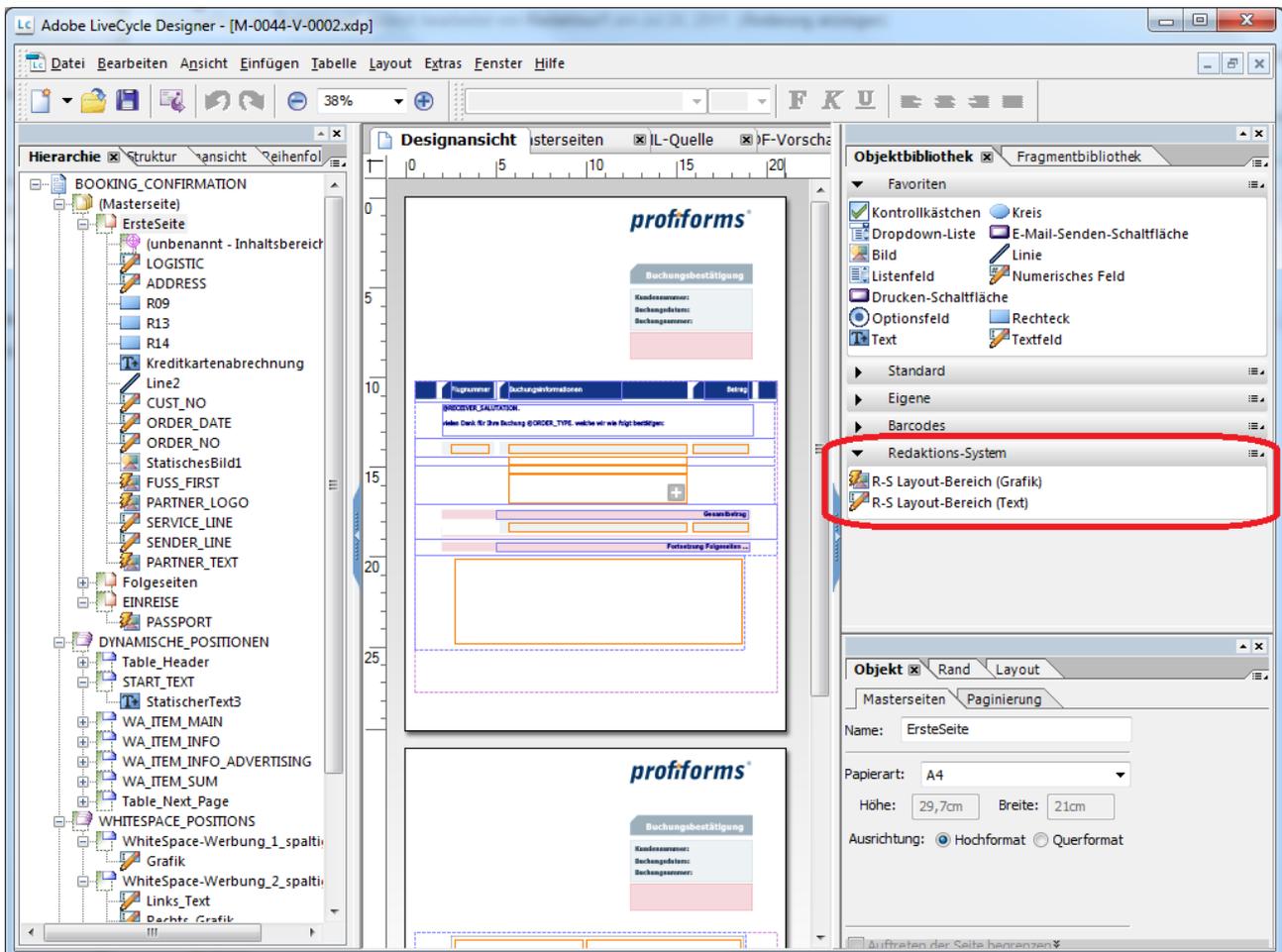


Bild: Adobe LiveCycle Designer mit einer Objekt-Bibliothek für Designobjekte des R-S

Die zwei dargestellten Objekte aus der Objekt-Bibliothek 'Redaktions-System' finden Sie als Beispiel hier:

- [R-S Layout-Bereich \(Grafik\).xfo](#)
- [R-S Layout-Bereich \(Text\).xfo](#)

**i** Neu angelegte Bibliotheken im Adobe LiveCycle Designer finden Sie meist im Windows-Nutzerverzeichnis an folgender Position:  
 C:\Users\%username%\AppData\Roaming\Adobe\Designer\8.2\DE\Objects\R-S (Beispiel für Windows 7 und Adobe LiveCycle Designer 8.2)

## Produkte - Konfiguration

Werden für die Aufbereitung der Dokumente die Produkte (ReportWriter/Logistics) der Firma profiforms verwendet, so sind in den jeweiligen Konfigurations-Dateien (TCI-Dateien) spezifische Einstellungen für das R-S vorzusehen. Diese Einstellungen in der TCI-Datei sind notwendig für folgende Funktionalitäten:

- Anzeige des Materials für physikalische Beileger im PDF-Inhaltsverzeichnis
- Anzeige von logischen Beilegern
- Realitätsnahe Berechnung von Gewicht und Porto zur Analyse der Verwendung optionaler Beileger und WhiteSpaces

### **Anzeige des Materials für physikalische Beileger im PDF-Inhaltsverzeichnis**

Physikalische Beileger für Dokumente aus dem Redaktions-System werden erst im Produktions-Prozess in den Druckzentren als Beileger über die Kuvertieranlage beigesteuert. Im Gegensatz zu logischen Beilegern können die physikalischen Beileger bei einer Vorschau aus dem R-S nicht als Dokument-Bestandteil im Layout angezeigt werden. Um die Korrektheit der Beisteuerung von physikalischen Beilegern prüfen zu können, wird in den PDF-Dateien der Druckvorschau des R-S ein Inhaltsverzeichnis erzeugt, welches durch das PDF-Anzeigeprogramm (z.B. Adobe Acrobat Reader) dargestellt werden kann. In diesem PDF-Inhaltsverzeichnis werden pro Dokument die dafür erzeugten physikalischen Beileger als 'Insertion' mit solchen Informationen wie der MaterialID und der Beschreibung ('Description') angezeigt. Die MaterialID und die Beschreibung für die physikalischen Beileger sind in der Material-Definition im R-S hinterlegt.

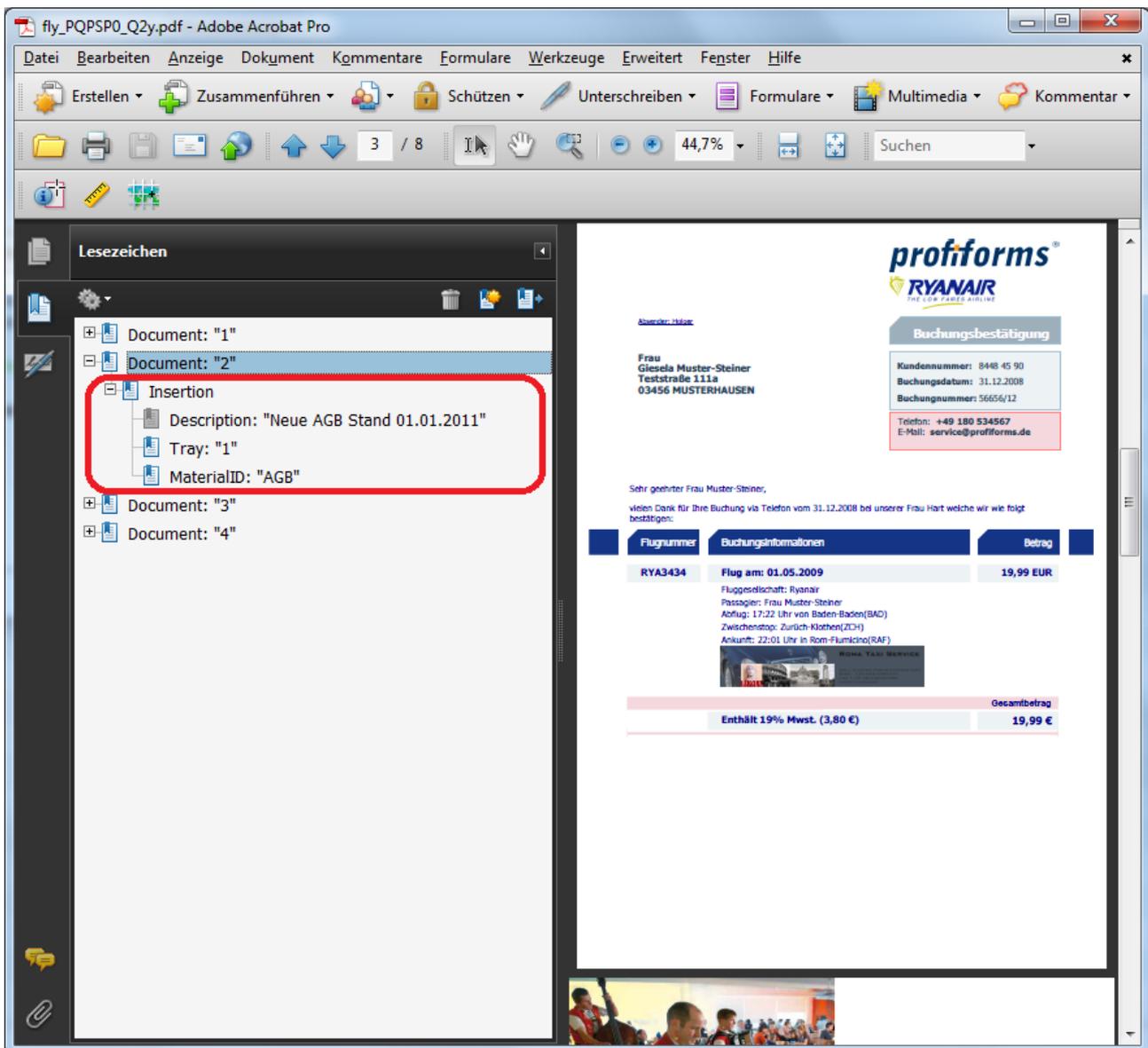


Bild: Inhaltsverzeichnis (rot umrandet) für verwendete physikalische Beileger im Adobe Acrobat Reader.

Damit dieses Inhaltsverzeichnis erzeugt wird, muss in den Konfigurationsdateien 'reportw.ini/logistics.ini' im verwendeten PDF-Profil (meist Default) die Eigenschaft 'MaterialReport' auf den Wert '1' gesetzt werden.

**Beispiel**

```

...
PDF {
  Profiles {
    Default {
      ...
      MaterialReport 1
    }
  }
}
...

```

**Anzeige von logischen Belegern**

Logische Beleger werden durch das R-S erzeugt und können schon während des Formular-Designs im R-S bei der Dokument-Vorschau angezeigt werden. Grundlage der Anzeige von logischen Belegern ist die Definition eines speziellen 'Supplement-Workitems' in den TCI-Dateien, welches für die Platzierung von logischen Belegern verwendet wird. Über dieses Workitem können außerdem vielfältige Einstellungen für die technische Verarbeitung von logischen Belegern wie z.B. die Archivierung oder der mehrkanalige Druck eingestellt werden. Der Verweis auf das Workitem, welches die Einstellungen für die logischen Beleger trägt, erfolgt über die Eigenschaft 'SupplementWorkitem' im Docdef-Objekt für das Transaktions-Formular.

**Beispiel**

```

DocDef booking_confirmation {

  ...
  SupplementWorkItem LogicalSupplements
}

```

Das dort definierte SupplementWorkItem verweist auf ein Workitem 'LogicalSupplement' welches im Folgenden exemplarisch dargestellt ist:

```

WorkItem LogicalSupplements {
  FormFileName Supplement
  Central      1
  FirstPage   SupplementPage
  Archive     0
  EnvelopeSortSystem {
    Type      NONE
    Channel   1
  }
}

```

### **Berechnung von Gewicht und Porto zur Analyse der Verwendung optionaler Beileger und WhiteSpaces**

In den TCI-Dateien der Transaktionsformulare muss in dem führenden WORKITEM ein Objekt 'EnvelopeSortSystem' wie folgt definiert sein:

```

...
EnvelopeSortSystem {
  Type      NONE
  Channel   1
  PSP       docxworld
  Class     Brief
  ZIPCode   71296
  CountryCode D
}
...

```

Im Konfigurations-Verzeichnis der Produkte ('etc' - Verzeichnis) muss eine Datei 'psp.ini' mit den Angaben des Briefversenders vorliegen. Es sollte folgendem Inhalt angelegt sein:

```
PostalServiceProvider docxworld {
  Products {
    Standard {
      Class          BRIEF
      Weight         20gramm
      Postage        0.55
    }
    Kompakt {
      Class          BRIEF
      Weight         50gramm
      Postage        0.90
    }
    Gross {
      Class          BRIEF
      Weight         500gramm
      Postage        1.45
    }
    Maxi {
      Class          BRIEF
      Weight         1000gramm
      Postage        2.20
    }
    MaxiPlus {
      Class          BRIEF
      Weight         2000gramm
      Postage        4.40
    }
  }
}
```

Eine Musterdatei für eine 'psp.ini' können Sie mit folgendem Link downloaden: [psp.ini](#)

## Registerkarte - Seiten

In dieser Registerkarte sehen Sie das Design der Rohdokumente aus dem Import-Paket. In den Seiten sehen Sie außerdem die Erweiterungs-Punkte des Dokuments, in die Sie mit Erweiterungen in Schemas Ressourcen einfügen können. Die Layout-Bereiche sind farbig hervorgehoben und enthalten den Namen des jeweiligen Erweiterungs-Punkts.

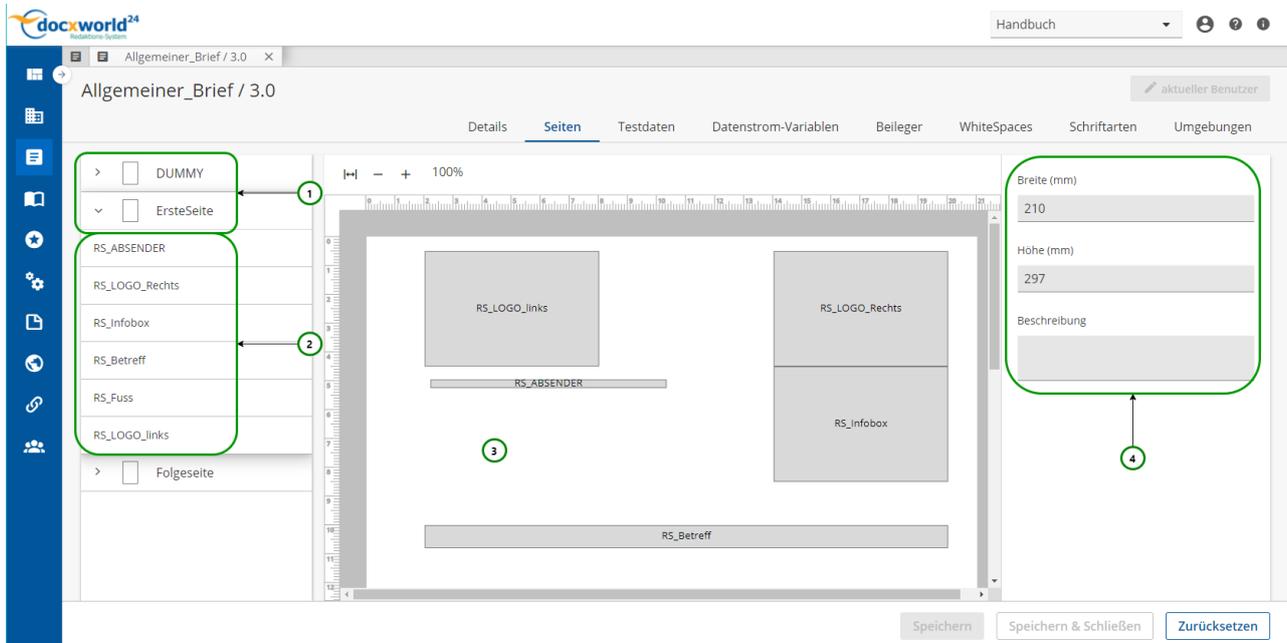


Abbildung A

### Sie sehen:

- welche **Seiten** es gibt *Abb. A (1)*,
- **Erweiterungspunkte** der Seite (2)
- das dazugehörige **Design der Seite** (3) und
- die **Beschreibung**, also die Höhe und Breite (Angaben in mm), der Seite bzw. der Erweiterungspunkte (4).

## Registerkarte - Testdaten

In dieser Registerkarte sehen Sie die vom Formular zur Verfügung gestellten Testdaten. Mit Ihnen können Sie später in der Schema-Version (oder im Produktions-Paket) eine Dokument-Vorschau generieren lassen, um die Schema-Version zu testen. Neben den vorgegebenen Testdaten des Formulars können Sie auch in der Schema-Version eigene Testdaten anlegen (**Testdaten bei Schema-Versionen**).

Details	Seiten	Testdaten	Datenstrom-Variablen	Beileger	WhiteSpaces	Schriftarten	Umgebungen
<input type="text" value=""/>							
Testdaten_21_12_2020	1	Datenstrom vom 21.12.2020	2				
Testdaten_alle_Briefe		Datenstrom zum Testen von allen Brief-Varianten					

Abbildung A

### Sie sehen:

- den **Namen** der Testdaten Abb. A (1)
- die dazu gehörige **Beschreibung** (2).

## Registerkarte - Datenstrom-Variablen in Formular-Versionen

In der Registerkarte sehen Sie die vom Transaktions-Formular zur Verfügung gestellten Datenstrom-Variablen. Diese Variablen bilden eine Schnittstelle zwischen den Datensystemen (ERP-/HOST-/CRM-Systeme, etc.) und dem Transaktions-Formular des R-S. Sie enthalten formularspezifische Daten, die durch das Datensystem zur Verfügung gestellt werden. Lesen Sie dazu den [Abschnitt über Datenstrom-Variablen](#).

Details	Seiten	Testdaten	<b>Datenstrom-Variablen</b>	Belegger	WhiteSpaces	Schriftarten	Umgebungen
>	DS.Tarif	Neuer Tarif des Kunden					
>	DS.Vorname	Vorname des Endkunden					
>	DS.Zaehlernr	Zaehlernummer Kunde					
∨	DS.advertisementindicator	Möchte Kunde Werbung erhalten					
	false	nein					
	true	Ja					
>	DS.gender	Geschlecht des Kunden					

Abbildung A

### Sie sehen:

- den **Namen** und die **Beschreibung** der Variablen (Datenstrom-Variablen beginnen immer mit 'DS.'). *Abb. A (1)*,
- den **Wert** der Variablen.

Wählen Sie dazu eine Datenstrom-Variable aus der Übersichtstabelle (1) aus und klappen Sie diese mit > auf. Die dazugehörigen Werte werden Ihnen dann angezeigt (3).

⚠ Diese Werte sind nur exemplarische Werte. Sie wurden durch den Formular-Designer festgelegt und sollen anzeigen, welche Werte die Variable annehmen kann (oder wahrscheinlich annimmt).

- die **Beschreibungen** der möglichen Werte (4).

## Registerkarte - Beileger

In dieser Registerkarte sehen Sie die vom Formular zur Verfügung gestellten Materialien, die Sie in Beilegern verwenden können *Abb. A (4)*. Sie können auch eigene Materialien definieren. In Logischen Beileger können Sie eigene Materialien aber nur verwenden, wenn dies erlaubt ist (2).

Abbildung A

Wenn durch das Import-Paket keine formulareigenen Materialien geliefert wurden, ist die Übersicht leer und Sie müssen für Beileger zwingend eigene Materialien erstellen.

### Sie sehen außerdem:

- ob **physikalische Beileger erlaubt** sind *Abb. A (1)*. Wenn sie erlaubt sind, dann ist ein Haken gesetzt.
- ob **logische Beileger erlaubt** sind (2). Wenn sie erlaubt sind, dann ist ein Haken gesetzt.
- ob logische Beileger **eigene Materialien** verwenden dürfen, die nicht durch das Formular bereitgestellt werden (2). Wenn eigene Materialien verwendet werden dürfen, ist der Haken nicht gesetzt.
- ob ein **Porto-Sprung erlaubt** ist (3). Wenn er erlaubt ist, dann ist ein Haken gesetzt.
- welche **Materialien** durch das Formular bereitgestellt werden (mit Name, Breite, Höhe, Stärke, Gewicht und Beschreibung) (4).

## Registerkarte - WhiteSpace

In dieser Registerkarte sehen Sie die Einschränkungen, die das Formular bezüglich **WhiteSpaces** definiert und die vorgegebenen Formular-Teile, die für WhiteSpaces freigegeben sind.

Details	Seiten	Testdaten	Datenstrom-Variablen	Beileger	WhiteSpaces	Schriftarten	Umgebungen
WhiteSpace erlaubt	1	<input checked="" type="checkbox"/>					
Maximum	2	50%					
Seitenüberlauf	3	ALLOWTOCREATEBACKPAGE					
Formular-Teile	4	letter Seiten					

Abbildung A

### Sie sehen außerdem:

- ob **WhiteSpaces erlaubt** sind Abb. A (1). Wenn sie erlaubt sind, dann ist ein Haken gesetzt.  
Sind WhiteSpaces verboten, können Sie zwar WhiteSpaces erstellen, das R-S wird diese allerdings nicht anwenden und produziert eine Fehlermeldung.
- wie groß der **maximale Werbe-Bereich** ist (2).  
Liegt der Wert bspw. bei 70%, so dürfen max. 70% der freien Fläche durch WhiteSpaces eingenommen werden.
- ob der **Seitenüberlauf erlaubt** ist oder nicht (3).  
Legt fest, ob WhiteSpaces so platziert werden dürfen, dass der Inhalt einer Seite dadurch auf eine neue Seite verschoben werden darf.
- auf welchen Formular-Teilen des Transaktions-Formulars **White-Spaces erlaubt** sind (mit Name und Beschreibung) (4).  
Sind Formular-Teile definiert, dürfen WhiteSpaces nur auf diesen Formular-Bereichen angewendet werden. Sind WhiteSpaces erlaubt und keine Formular-Teile definiert, können WhiteSpaces auf das komplette Formular angewandt werden.

## Registerkarte - Schriftarten

Ein Transaktions-Formular legt auch fest, welche Schriftarten verwendet werden dürfen. In Schemas dürfen nur die Schriftarten verwendet werden, die in dem Formular, das die Schema-Version als Vorlage nutzt, freigegeben wurden. Bei der Erstellung eines RichTextes können Sie dann nur die Schriftarten auswählen, die durch das Formular definiert worden sind. Einsehen (aber nicht ändern) können Sie die erlaubten Schriftarten in der Registerkarte Schriftarten.

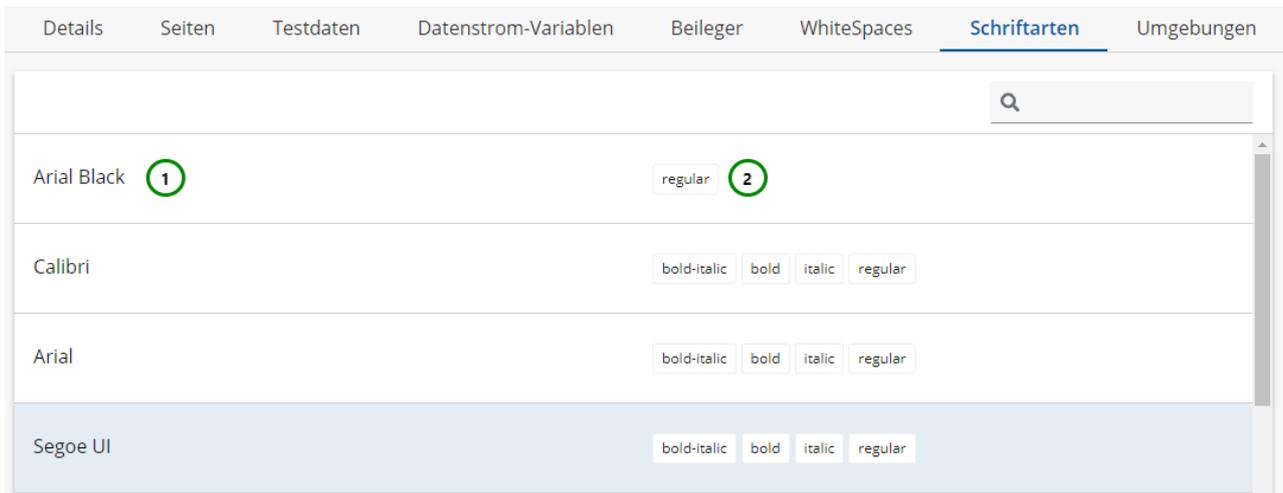


Abbildung A

### Sie sehen:

- den **Namen** der Schriftart *Abb. A (1)*,
- die erlaubten **Stile** der Schriftart (2): regular, **bold**, *italic*, **bold-italic**.

## Registerkarte - Umgebungen

In dieser Registerkarte sehen Sie, welche Programme das Formular für das Binär-Paket voraussetzt. Das Binär-Paket enthält Programme, welche die Dokumente und Ressourcen aus dem R-S zu fertigen Dokumenten zusammenfügen. Mehr dazu erfahren Sie [hier](#). Um später aus dem Produktions-Paket Dokumente generieren zu können, müssen die im Binär-Paket enthaltenen Programme den hier gemachten Vorgaben entsprechen.

Details	Seiten	Testdaten	Datenstrom-Variablen	Beileger	WhiteSpaces	Schriftarten	Umgebungen
<input type="text" value=""/>							
linux	①		Operations_binary	②	01.5.10.*	③	

Abbildung A

### Sie sehen:

- die **Plattform der Umgebung** *Abb. A (1)*,
- das **Programm, das vorausgesetzt wird** (2),
- die erlaubten **Versionen des Programms**.

01.5.10.\* bedeutet, dass alle Binär-Pakete mit dem Namen Operations\_binary und der Hauptnummer 01.5.10 erlaubt sind, egal welche weiteren Unternummern die Version hat (3). Es wird immer die höchste vorhandene Version genutzt.

## Arbeiten mit Schemas

Nachdem Sie im vorigen Kapitel das Arbeiten mit Formularen, also den Rohdokumenten, kennengelernt haben, wird in diesem Kapitel das Arbeiten mit Schemas und Schema-Versionen behandelt, also die Gestaltung der Rohdokumente. Um zu der Übersicht des Schemas zu gelangen, wechseln Sie in den Navigationspunkt Schemas .

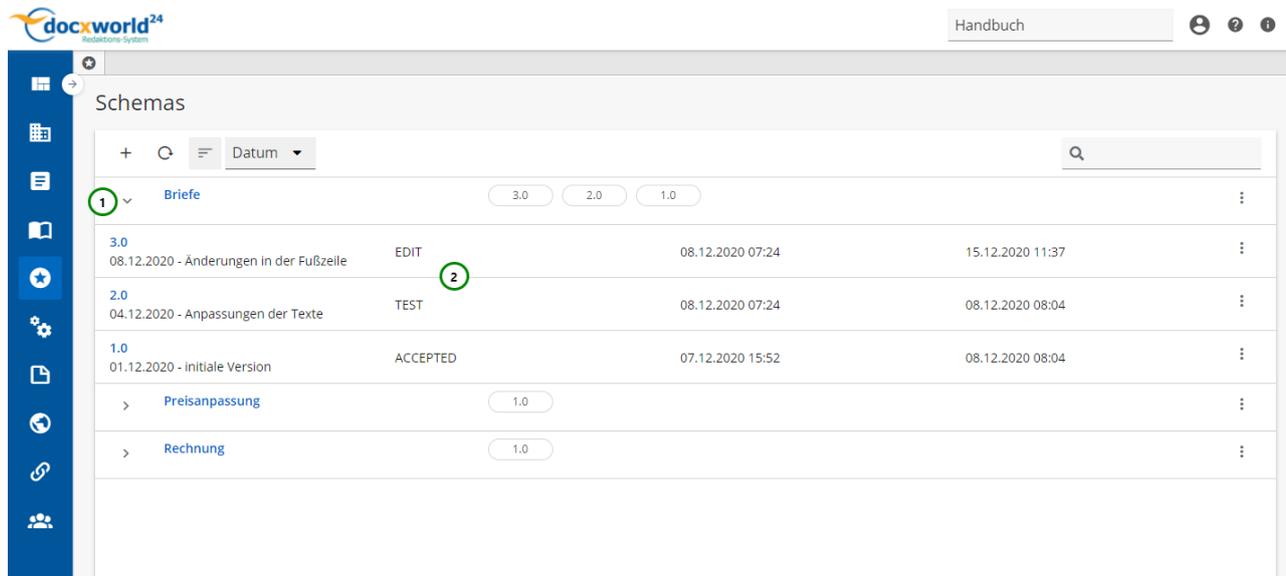


Abbildung A

Schemas dienen zur Gestaltung von Transaktions-Formularen. Mit Schemas gestalten Sie die Rohdokumente aus dem Formular, füllen die vorgegebenen Layout-Bereiche mit Inhalt, definieren Beileger, WhiteSpaces und etc.

Die Bearbeitungsmöglichkeiten des Schemas hängen dabei vom Transaktions-Formular ab. Sobald Sie ein Schema *Abb. A (1)* erstellt haben, können Sie eine Version des Schemas (2) anlegen. Mit diesen Schema-Versionen können Sie dann arbeiten (siehe: [R-S Workflow und Versionierung](#)).

Falls Sie den Abschnitt über Schemas noch nicht gelesen haben, tun Sie dies bitte [hier](#), bevor Sie beginnen, mit Schemas zu arbeiten.

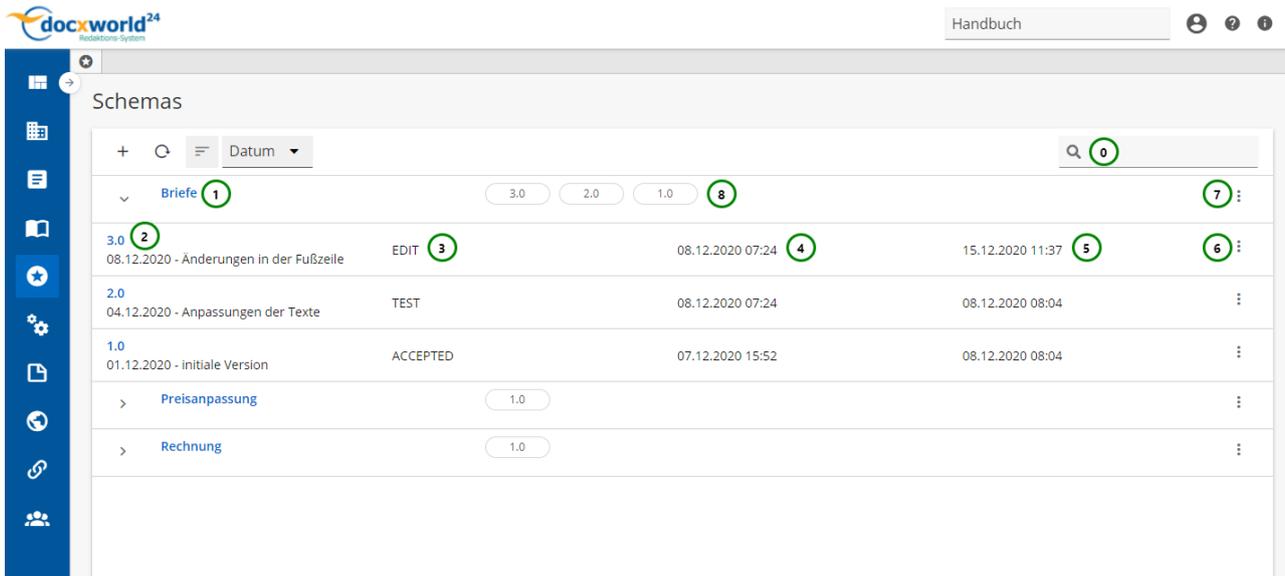


Abbildung B

Sie sehen in **Abb. B**:

- (0) Die Suche, in der Sie nach Schema-Versionen suchen können
- (1) Name des Schemas
- (2) Versions-Nummer des Schemas inklusive Beschreibung
- (3) Den aktuellen Status der Schema-Version
- (4) Das Erstelldatum der Schema Version
- (5) Das letzte Änderungsdatum der Schema-Version
- (6) Das Kontextmenü der ausgewählten Schema-Version
- (7) Das Kontextmenü des ausgewählten Schemas
- (8) Quicklinks zu den letzten drei verwendeten Schema-Versionen des Schemas

## Schema erstellen, ändern und löschen

In der Schema-Übersicht stehen Ihnen folgende Aktionen zur Verfügung, um ein Schema neu zu erstellen oder ein bereits vorhandenes Schema zu bearbeiten:

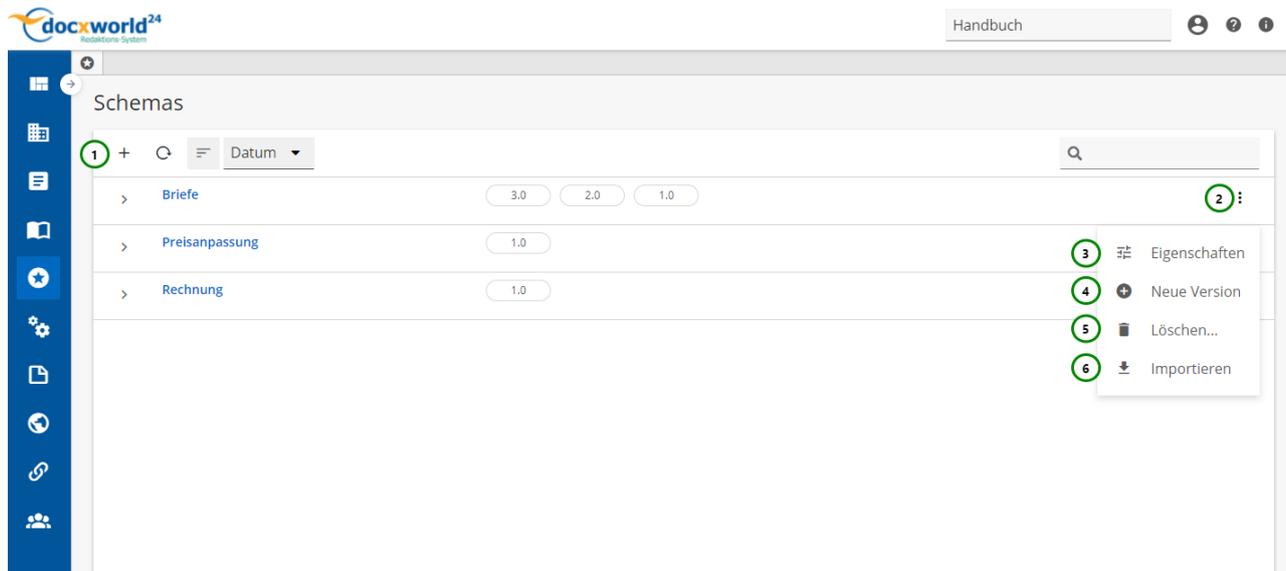


Abbildung A

Um ein Schema zu erstellen, klicken Sie bitte auf **+** [Abb. A](#) (1). Im weiteren öffnet sich ein Popup-Fenster, in welchem Sie den Namen und die Beschreibung erstellen können ([siehe Abb. B](#)).

Mit (2) öffnen Sie das **⋮** Kontextmenü. Hier stehen Ihnen **≡** Eigenschaften (3), **+** Neue Version (4), **🗑** Löschen (5) und **↓** Importieren (6) zur Auswahl.

Wollen Sie ein vorhandenes Schema ändern, wählen Sie das **⋮** Kontextmenü (2) des gewünschten Schemas in der Übersicht aus und klicken Sie anschließend auf **≡** (3). Auch hier öffnet sich ein Popup-Fenster, in welchem Sie das Schema aktualisieren können ([siehe Abb. B](#)).

Abbildung B

⚠ Beim Löschen des Schema müssen dessen Versionen bereits gelöscht sein. Beachten Sie dabei, dass das Objekt durch das Löschen nur "inaktiv" gesetzt wird (es ist weder für den Nutzer noch für andere Objekte sichtbar oder verwendbar) und durch einen Administrator auch wiederhergestellt werden kann. Mehr Informationen zum Löschen von Objekten finden Sie [hier](#).

## Schema-Version

Hier sehen Sie die Übersicht, mit der Sie Schema-Versionen erstellen, öffnen, löschen, kopieren, für sie einen Status setzen, exportieren und importieren können.

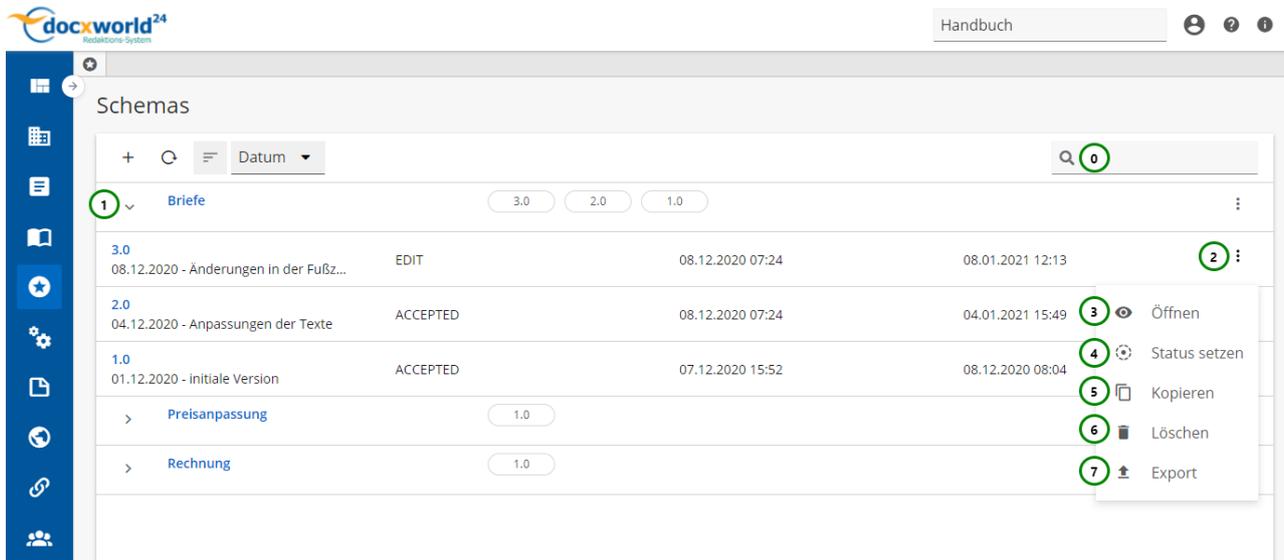


Abbildung A

Sie sehen:

- (0) Die Suchfunktion  , um die Auswahl einzuschränken.
- (1) Mit Hilfe von  können Sie Schemas aufklappen bzw.  zuklappen.
- (2) Das Kontextmenü  .
- (3) Öffnen  , um die Schema-Version zu öffnen.
- (4) Status setzen  , um den Workflow der Schema-Version zu ändern.
- (5) Kopieren  , um eine Kopie der Schema-Version mit einer neuen Versionsnummer zu erstellen.
- (6) Löschen  , um die Schema-Version zu löschen.
- (7) Exportieren  , um die Schema-Version als Zip zu exportieren.

Durch einen Klick auf  [Abb. A \(1\)](#) öffnen Sie die Versionsübersicht zu dem ausgewählten Schema. Nachdem Sie ein Schema neu erstellt haben, müssen Sie eine Version dieses Schemas anlegen (*siehe [hier](#)  Neue Version (4)*), mit der Sie dann arbeiten können. Mit dieser Schema-Version können Sie dann ein Formular gestalten. (*Mehr Informationen über Versionierung und den R-S Workflow finden Sie [hier](#)*). Außerdem können Sie die Schema-Versionen-Liste mit Hilfe von  (0) durchsuchen. Ausführliche Informationen zur Suchfunktion finden Sie [hier](#).

## Schema-Version erstellen, ändern und löschen

In der Schema-Versionen Übersicht stehen Ihnen folgende Aktionen zur Verfügung, um eine Schema-Version neu zu erstellen oder eine bereits vorhandene Schema-Version zu bearbeiten:

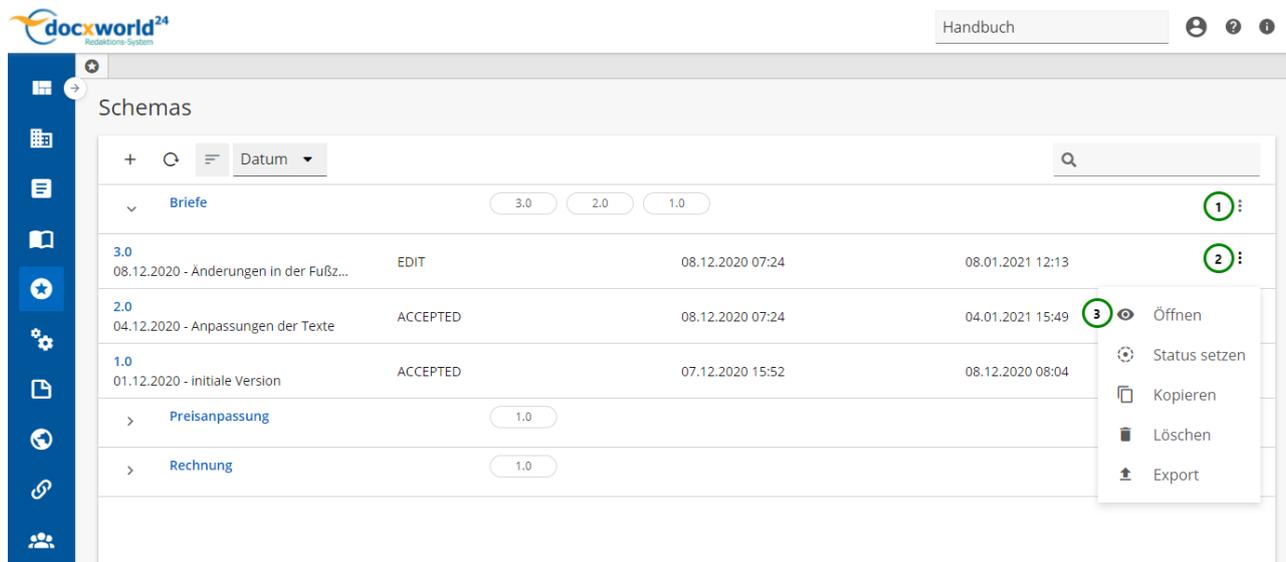


Abbildung A

Um eine Schema-Version zu erstellen, klicken Sie bitte auf [Abb. A \(1\)](#) und anschließend auf Neue Version (Abbildung dazu [hier](#)).

Um eine bestehende Schema-Version zu öffnen, wählen Sie bitte (2) der Schema-Version, die Sie öffnen möchten, aus und klicken Sie anschließend auf Öffnen (3).

Wenn Sie eine Schema-Version neu erstellen oder eine bereits vorhandene Schema-Version öffnen und bearbeiten möchten, öffnet sich folgende Übersicht:

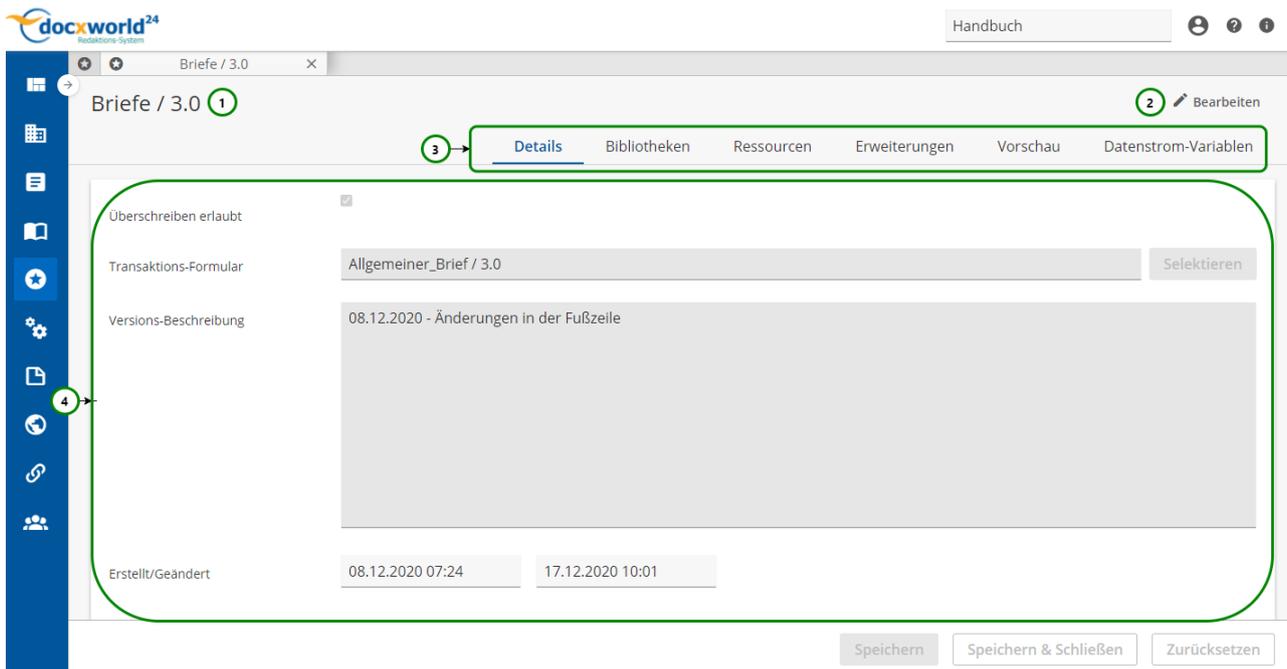


Abbildung B

⚠ Sie befinden sich immer zuerst im **View-Modus**. Die Bearbeitung kann mit  Abb. B (2) begonnen werden.

Sie sehen in Abb. B:

- (1) Name / Version des Schemas
- (2) Bearbeitungsmodus aktivieren
- (3) Das Register der Schema-Version
- (4) Inhalt des ausgewählten Registers

## Schema-Version kopieren

In der Schema-Versionen Übersicht steht Ihnen folgende Aktion zur Verfügung, um eine Schema-Version zu kopieren:

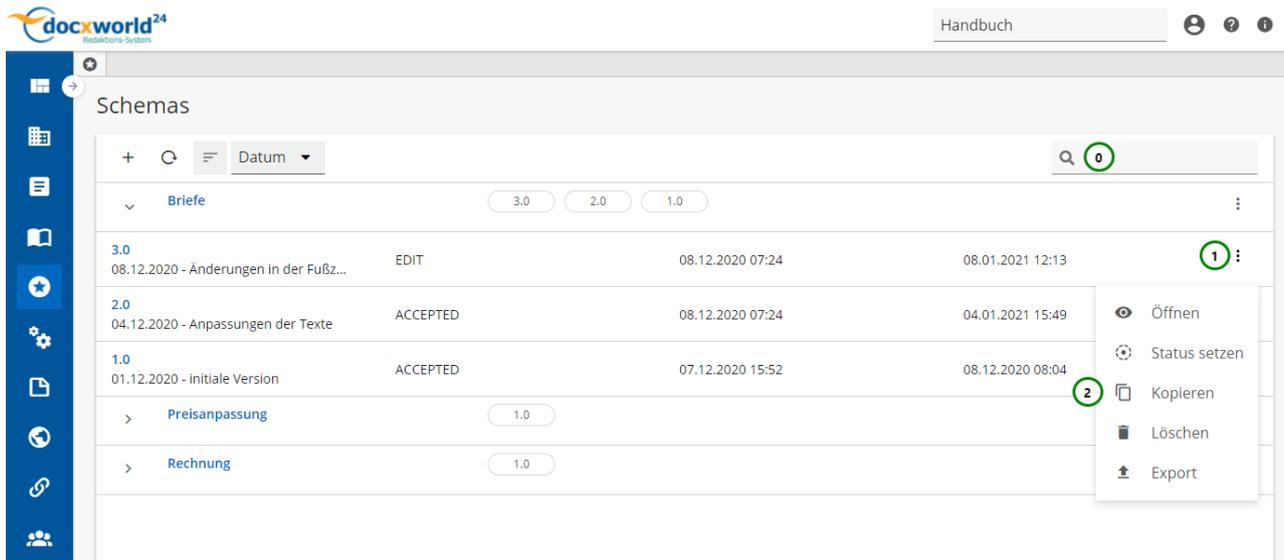


Abbildung A

Um eine Schema-Version zu kopieren, wählen Sie bitte die Schema-Version aus, die Sie kopieren möchten und klicken Sie auf das  Kontextmenü Abb. A (1) und anschließend auf  Kopieren (2). Folgendes Fenster öffnet sich:



Abbildung B

Geben Sie für die neue Schema-Version eine Versionsnummer an Abb. B (1). Diese muss sich von der alten Version unterscheiden und darf für dieses Schema noch nicht existieren. Klicken Sie anschließend auf (2), um den Kopiervorgang abzuschließen oder auf (3), um ihn abubrechen. Sie finden die kopierte Schema-Version jetzt in der Schema-Versionen-Liste:

Briefe		3.0_Copy	3.0	2.0	
<b>3.0_Copy</b>	08.12.2020 - Änderungen in der Fußzeile	EDIT	17.12.2020 12:45	17.12.2020 12:45	⋮
<b>3.0</b>	08.12.2020 - Änderungen in der Fußzeile	EDIT	08.12.2020 07:24	17.12.2020 12:24	⋮
<b>2.0</b>	04.12.2020 - Anpassungen der Texte	TEST	08.12.2020 07:24	08.12.2020 08:04	⋮
<b>1.0</b>	01.12.2020 - initiale Version	ACCEPTED	07.12.2020 15:52	08.12.2020 08:04	⋮

Abbildung C



Wenn Sie eine Schema-Version kopieren, hat die Kopie immer den Status EDIT.

Eine kopierte Schema-Version besitzt alle Eigenschaften und Ressourcen, die das Original zum Kopierzeitpunkt besaß. Ändert sich das Original oder die Kopie, hat dies aber keine Auswirkungen auf die andere Schema-Version, bei der Kopie handelt es sich um eine neues, eigenständiges Objekt. Da Kopien den Status EDIT haben, ist es durch Kopieren möglich, mit geschlossene Schema-Versionen weiterzuarbeiten, ohne sie komplett nachbilden zu müssen.

Sie können die Schema-Versionen-Liste mit Hilfe von  Abb. A (0) durchsuchen. Ausführliche Informationen zur Suchfunktion finden Sie [hier](#).

## Status einer Schema-Version setzen

In der Schema-Versionen Übersicht steht Ihnen folgende Aktion zur Verfügung, um den Status einer Schema-Version zu ändern:

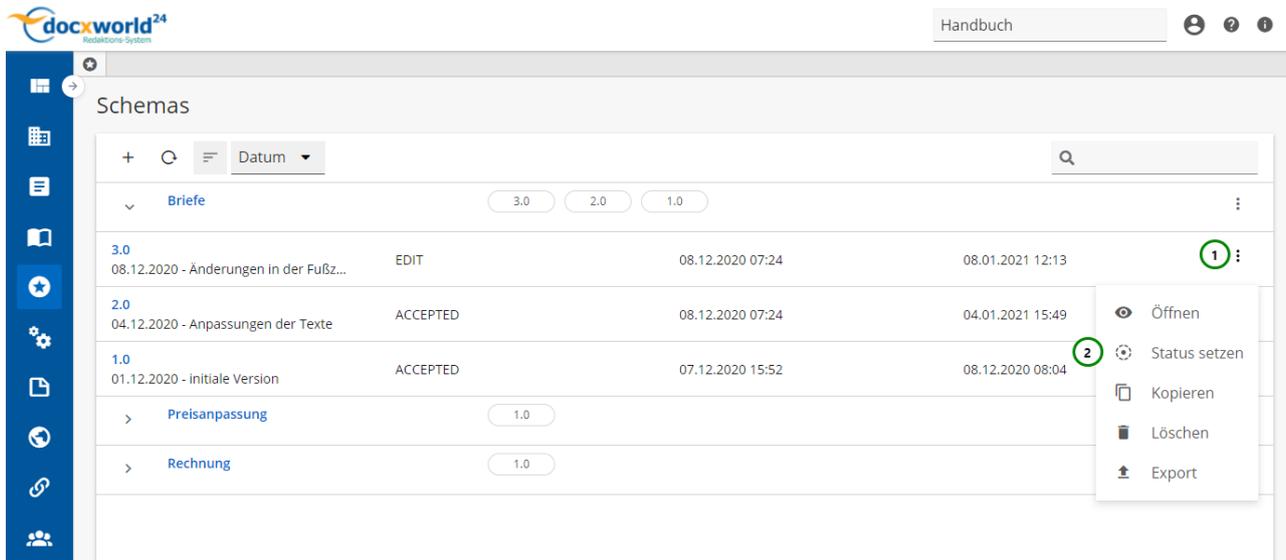
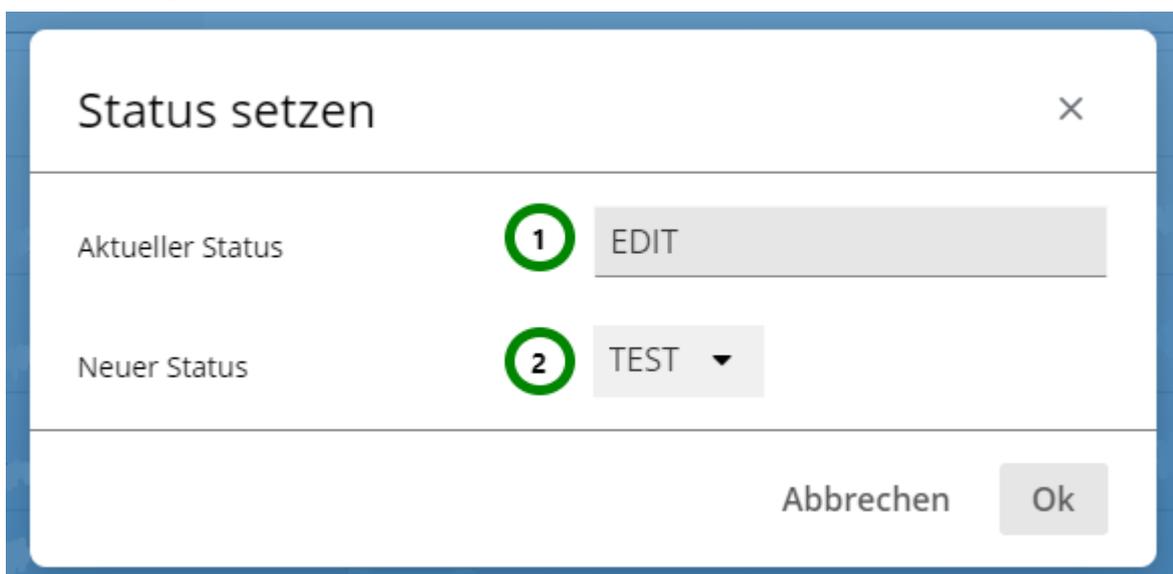


Abbildung A

Schemas sind wie **Transaktions-Formulare**, **Bibliotheken** oder **Produktions-Pakete** ein Teil des R-S Status-Workflows. Sie sind deshalb der Versionierung unterworfen und besitzen einen **Status**. Lesen Sie dazu bitte den Abschnitt über den **R-S Workflow**.

Mit **Abb. A (1)** öffnen Sie das Kontextmenü der ausgewählte Schema-Version. Wählen Sie hier **Status setzen (2)**. Folgendes Fenster öffnet sich:



Abbildung

Sie sehen **Abb. B**:



- den aktuellen **Status** der Schema-Version
- den neuen **Status** der Schema-Version



- Der Status kann nur verändert werden, wenn kein Benutzer im Bearbeitungsmodus der ausgewählten Schema-Version ist.
- Die Auswahl in **Abb. B (2)** wird durch den aktuellen Status bestimmt, siehe hierzu **Abb. C**.

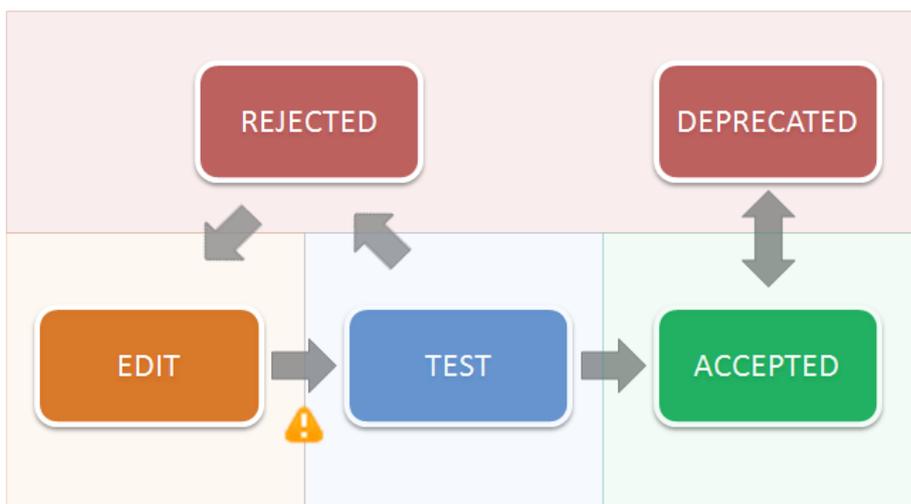


Abbildung C



Sie können den Status der Schema-Version erst zu TEST ändern, wenn das Transaktions-Formular und alle eingebundenen Bibliotheken den Status ACCEPTED haben.

Klicken Sie auf Ok (**Abb. B**), um den neuen Status zu setzen oder auf Abbrechen (**Abb. B**), um den aktuellen Status beizubehalten. Der Status der Schema-Version ist jetzt geändert.



Für eine genaue Beschreibung des Status-Workflows bei Schema-Versionen lesen Sie bitte unbedingt den **Abschnitt über den R-S Workflow**.

## Schema-Version exportieren

In der Schema-Versionen Übersicht steht Ihnen folgende Aktion zur Verfügung, mit der Sie Schema-Versionen exportieren können:

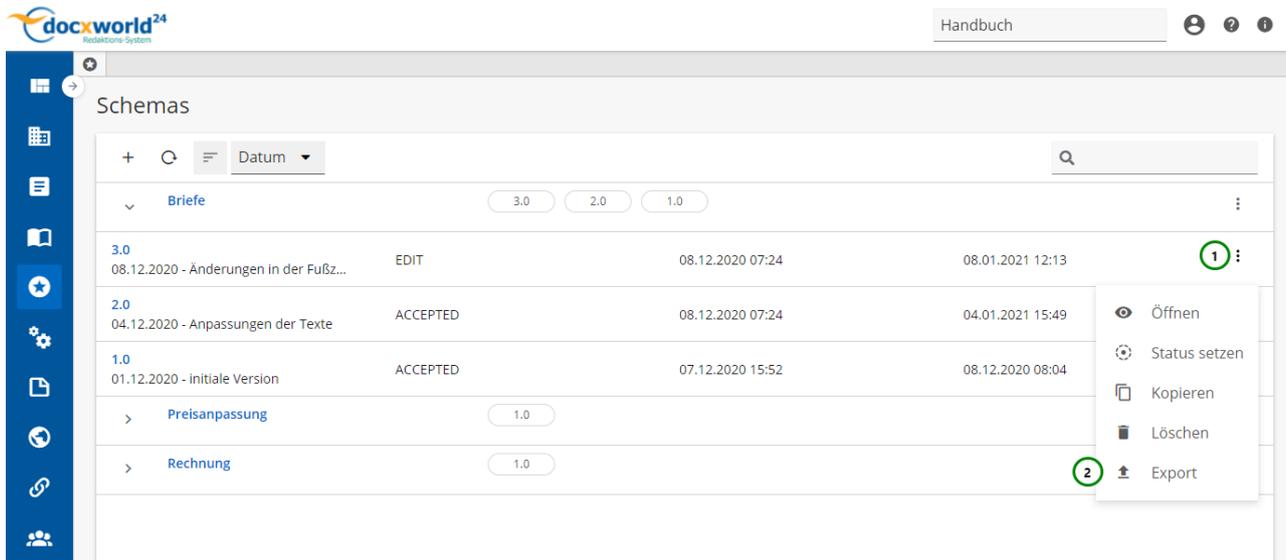


Abbildung A

Sie können eine Schema-Version exportieren, um Sie außerhalb des Arbeitsbereichs verfügbar zu machen und sie dann z.B. in einem anderen Arbeitsbereich wieder zu importieren und dort zu verwenden. Wählen Sie dazu die Schema-Version aus, die Sie exportieren möchten und öffnen Sie das Kontextmenü [Abb. A](#) (1) und klicken Sie anschließend zum Exportieren auf  (2). Der Export startet dann voll automatisch und die Schema-Version wird in Ihr Standard-Download-Verzeichnis exportiert (*Unter Windows: User-Verzeichnis/Downloads*). Sie finden die Datei, indem Sie zum Download-Verzeichnis wechseln oder sich den Download in Ihrem Browser anzeigen lassen. Die Datei wird standardmäßig unter dem Namen "schema-revision-export" abgespeichert und besitzt keine Dateiendung, liegt aber im ZIP-Format vor.

 Wenn Sie eine exportierte Schema-Version wieder importieren, besitzt diese immer den Status EDIT.

## Schema-Version importieren

In der Schema-Versionen Übersicht steht Ihnen folgende Aktion zur Verfügung, mit der Sie Schema-Version importieren können:

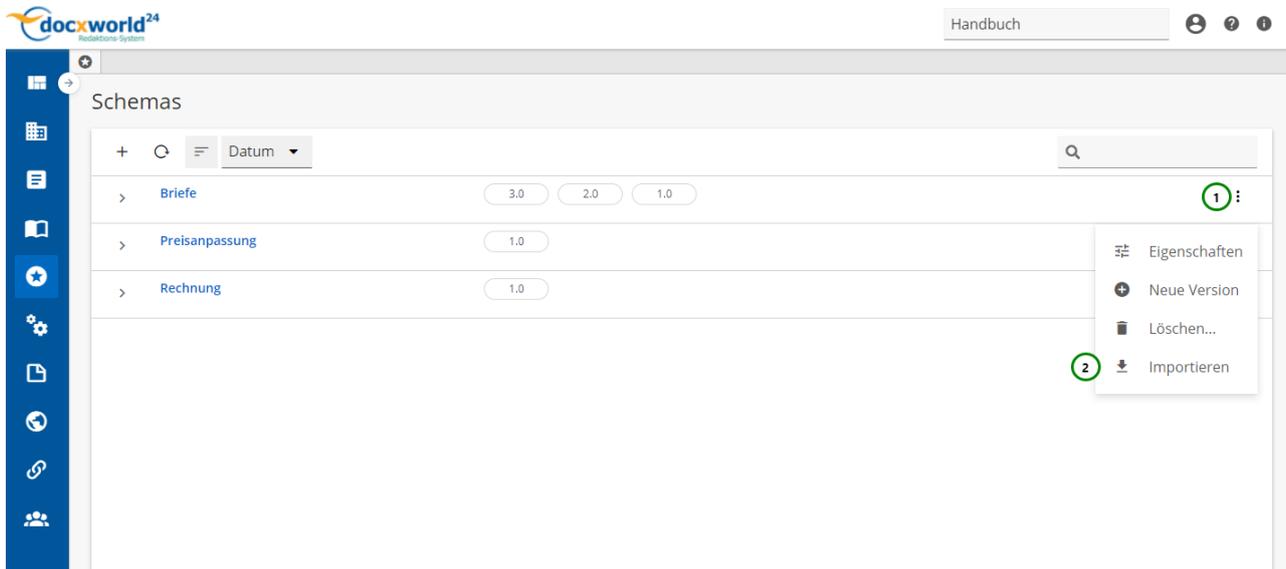


Abbildung A

Sie können eine zuvor **exportierte** Schema-Version wieder importieren. Beachten Sie jedoch, dass in dem Arbeitsbereich, in den Sie die Schema-Version importieren, das dazugehörige Transaktions-Formular vorhanden sein muss, da die Verweise in der Schema-Version sonst ungültig werden. Beim Import einer Schema-Version verliert diese außerdem ihre Bindung zum Transaktions-Formular, weshalb Sie das Formular erneut einbinden müssen (siehe [hier](#)). Außerdem müssen Sie Materialien für Beileger gegebenenfalls neu Verknüpfen.

Um eine Schema-Version zu **importieren**, klicken Sie auf das Kontextmenü [Abb. A \(1\)](#) und wählen Sie Importieren  (2) aus. Folgender Dialog öffnet sich:



Abbildung B

Geben Sie in Abb. B (1) eine Versionsnummer für die importierte Schema-Version an. Diese darf im Schema noch nicht vorhanden sein. Klicken Sie anschließend auf (3), um die zu importierende Schema-Version auszuwählen. Nachdem Sie eine Datei ausgewählt haben, erscheint der Button (4), mit dem Sie Ihre Auswahl wieder entfernen können. Ist die Datei bereit zum Import, so sehen Sie in (2) den Dateinamen. Ob es sich um eine gültige Schema-Versions-Datei handelt wird erst beim Importieren überprüft. Klicken Sie zum Importieren der Datei auf (5) oder zum Abbruch des Importvorgangs auf (6). Falls die Datei ungültig ist, bricht das R-S mit einer Fehlermeldung ab. Andernfalls schließt sich der Dialog automatisch und die Schema-Version ist nun in das gewählte Schema Abb. A importiert.



Wenn Sie eine Schema-Version importieren, hat diese immer den Status EDIT.

Eine importierte Schema-Version ist, wie eine **Kopie**, unabhängig von der Original-Schema-Version. Auch wenn das Original in dem selben Schema vorhanden ist, können Sie mit der importierten Schema-Version arbeiten. Eine Änderung an der einen Schema-Version wirkt sich nicht auf die Andere aus.

## Registerkarten bei Schema-Versionen

Bei Schema-Versionen gibt es folgende **Registerkarten**:

Details    Bibliotheken    Ressourcen    Erweiterungen    Vorschau    Datenstrom-Variablen

Sie beinhalten die Eigenschaften und Inhalte der Schema-Version.

### Kurzbeschreibung der Registerkarten und ihrer Inhalte:

Registerkarte	Beschreibung
<b>Details</b>	Sie beinhaltet allgemeine Informationen über die Schema-Version und ermöglicht die Auswahl eines Transaktions-Formulars, das die Rohdokumente und Daten, die im Schema gestaltet werden sollen, enthält.
<b>Bibliotheken</b>	Ermöglicht die Verwaltung der in die Schema-Version eingebundenen Bibliotheken.
<b>Ressourcen</b>	Stellt die <b>Ressourcenverwaltung</b> der Schema-Version zur Verfügung. Hier können Sie die Ressourcen der Schema-Version erstellen, ändern, löschen und verwalten. Da es Ressourcenverwaltung nahezu in der gleichen Form auch bei Bibliotheks-Versionen gibt, wird sie in einem gesonderten Kapitel behandelt. Es empfiehlt sich, zuerst die Abschnitte über Schema-Versionen und über Bibliotheken zu lesen, bevor Sie sich mit der Ressourcenverwaltung vertraut machen.
<b>Erweiterungen</b>	Ermöglicht die Erstellung und Verwaltung von Erweiterungen, mit denen Sie Ressourcen in die Dokumente einfügen und Beileger an das Formular anheften können.
<b>Vorschau</b>	Ermöglicht das Erstellen einer Testvorschau für die Schema-Version.
<b>Datenstrom-Variablen</b>	Zeigt die Datenstrom-Variablen, die von der Formular-Version zur Verfügung gestellt werden.

## Registerkarte - Details in Schema-Versionen

Die Registerkarte Details enthält allgemeine Informationen über die Schema-Version. Außerdem geben Sie hier das verwendete Transaktions-Formular an.

Wenn Sie eine Schema-Version neu erstellen oder öffnen befinden Sie sich automatisch in der Registerkarte Details.

The screenshot shows the 'Briefe / 3.0' details page in the Profiforms system. The page is titled 'Briefe / 3.0' and has a 'Handbuch' search bar in the top right. The main content area is divided into several sections:

- Überschreiben erlaubt:** A checkbox is checked, indicated by a circled '1'.
- Transaktions-Formular:** A dropdown menu shows 'Allgemeiner\_Brief / 3.0' selected, with a 'Selektieren' button to the right. A circled '2' is next to the dropdown.
- Beschreibung:** A text area contains '08.12.2020 - Änderungen in der Fußzeile', with a circled '3' next to it.
- Erstellt / Geändert:** Two date-time fields are shown: '08.12.2020 07:24' and '17.12.2020 14:18', with a circled '4' next to the first field.

At the bottom of the page, there are three buttons: 'Speichern', 'Speichern & Schließen', and 'Zurücksetzen'. A vertical sidebar on the left contains various navigation icons.

Abbildung A

**Sie können:**

- ★ Zulassen, ob **Überschreibungen** von **Ressourcen** erlaubt sein sollen. Wenn ja, setzen Sie den Haken bei *Abb. A (1)*.  
Ist diese Option aktiviert, können Ressourcen gleichnamige Ressourcen mit geringerer Priorität überschreiben (siehe **Bibliotheken**).
- ★ Eine **Version** des **Transaktions-Formulars** selektieren.  
Beim Erstellen klicken Sie bitte auf Selektieren (2). Es öffnet sich eine Übersichtstabelle mit allen Transaktions-Formularen. Wählen Sie eine Formular-Version aus. Mit ✕ (2) können Sie ein ausgewähltes Transaktions-Formular wieder entfernen. Um das Formular zu ändern, benutzen Sie ebenfalls Selektieren (2). Ohne Transaktions-Formular können im Schema keine Dokumente gestaltet werden!
- ★ Der Schema-Version eine **Beschreibung** geben (3).  
Die Beschreibung ist optional und wird z.B. in Auswahldialogen angezeigt.
- ★ Hier wird automatisch das Datum und die Zeit der Schemaerstellung und -änderung eingetragen (4).  
⚠ Diese Felder sind nicht beschreibbar.

- ⚠ Zum Verändern der Inhalte müssen Sie vorher den Bearbeitungsmodus (  Bearbeiten) aktivieren (nur im Status Edit möglich).

## Registerkarte - Bibliotheken

In dieser Registerkarte können Sie die Bibliotheken, welche Ihre Schema-Version verwendet, verwalten. Bevor Sie eine Bibliothek einbinden können, müssen Sie diese erst erstellen. Lesen Sie dazu bitte den Abschnitt [Arbeiten mit Bibliotheken](#).

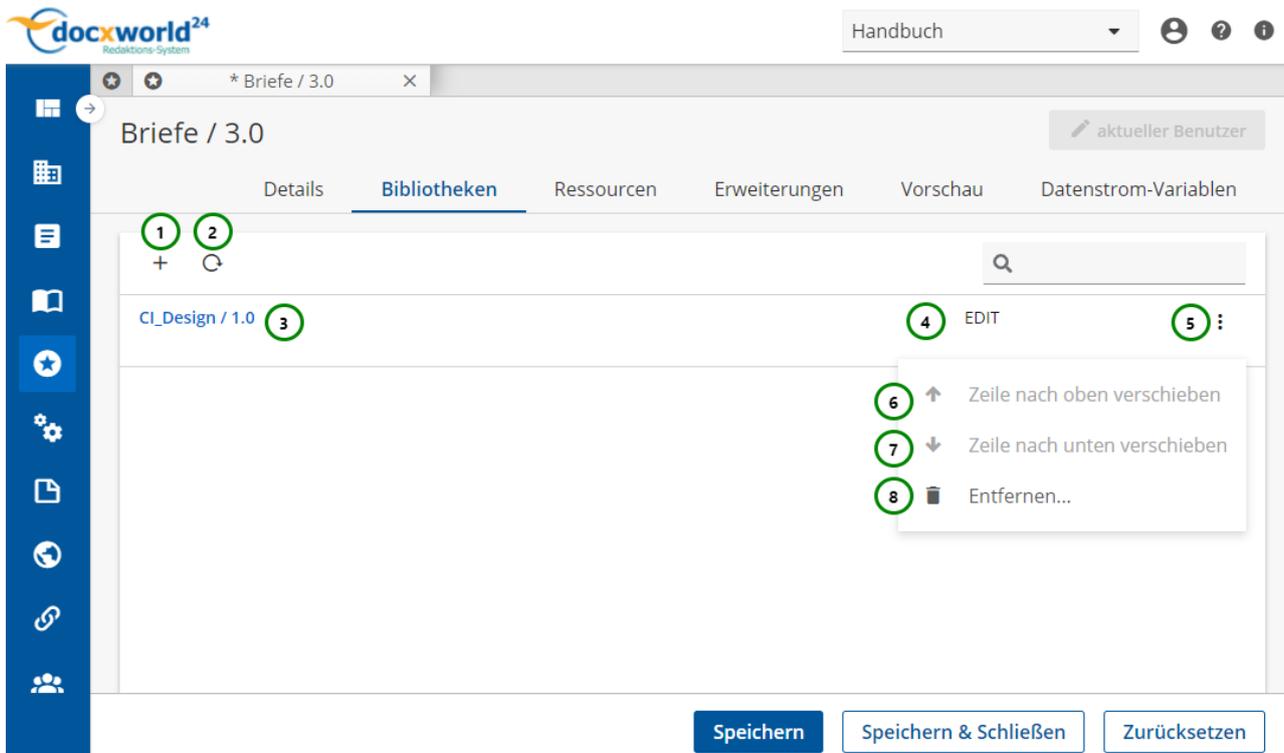


Abbildung A

### In der Registerkarte sehen Sie:

- Welche Bibliotheks-Versionen in das Schema eingebunden sind [Abb. A](#) (3).
- Den Status der eingebundenen Bibliotheks-Version (4).

### Sie können:

- Eine Bibliothek zur Schema-Version hinzufügen [Abb. A](#) (1).
- Eine Bibliothek aus der Schema-Version entfernen (8).
- Bibliotheken in der Liste auf- und abwärts verschieben (6 & 7).
- Die Bibliotheksliste neu laden (2)

### **Bibliotheks-Version hinzufügen**

Um eine neue Bibliotheks-Version in das Schema einzubinden, drück Sie **+** Abb. A (1).  
Anschließend öffnet sich ein neues Fenster:

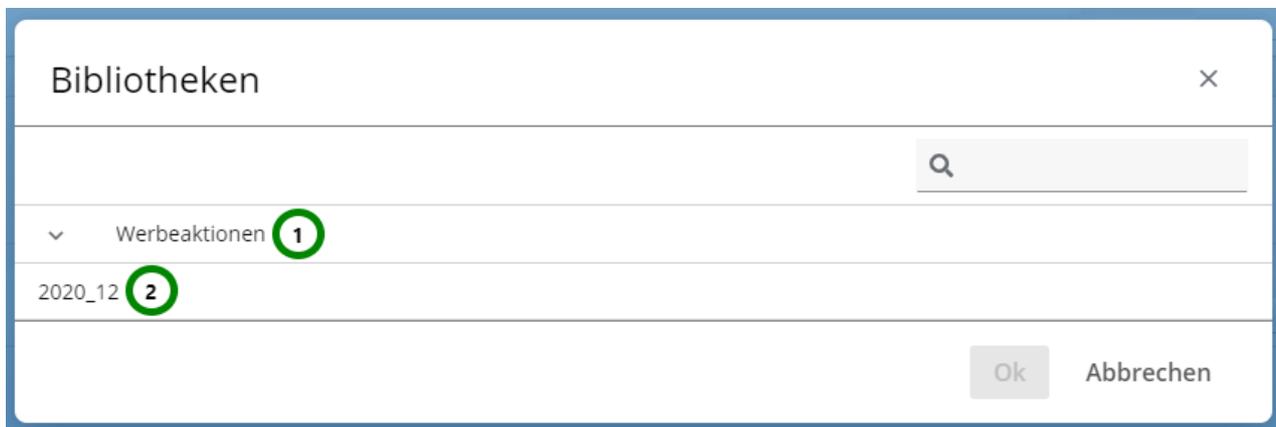


Abbildung B

In diesem Fenster sehen Sie vorhandene Bibliotheken Abb. B (1) und die dazugehörigen Bibliotheks-Versionen (2). Wählen Sie die gewünschte Bibliotheks-Version (2) aus und bestätigen Sie mit Ok.

### **Bibliotheks-Version entfernen**

Zum Entfernen einer Bibliotheks-Version, öffnen Sie das **⋮** Kontextmenü Abb. A (5) der Bibliotheks-Version, die Sie entfernen möchten. Hier steht Ihnen die Option Entfernen (8) zur Verfügung.

**⚠** Das Entfernen einer Bibliotheks-Version aus dem Schema entfernt lediglich die Verlinkung. Die Bibliotheks-Version bleibt weiterhin erhalten.

### **Bibliotheks-Version verschieben**

Zum Verschieben einer Bibliotheks-Version können Sie über das **⋮** Kontextmenü Abb. A (5) die Buttons Zeile nach oben verschieben oder Zeile nach unten verschieben verwenden.

Sie können die Priorität von Bibliotheken festlegen, indem Sie die Einträge in der Übersichtstabelle Abb. A verschieben. Die Position einer Bibliothek in der Liste Abb. A legt zugleich ihre Priorität fest. Diese nimmt von oben nach unten zu. Umso weiter unten eine Bibliothek also in der Liste steht, desto höher ist ihre Priorität.

Beinhalten zwei oder mehr Bibliotheken Ressourcen mit dem gleichen Namen, so überschreibt die Ressource aus der Bibliothek mit der höchsten Priorität die Ressourcen der anderen Bibliotheken.

Ressourcen, die von anderen Ressourcen überschrieben werden, werden als durchgestrichen gekennzeichnet.

**⚠** Schema-Ressourcen haben immer Vorrang vor Bibliotheks-Ressourcen.

## Registerkarte - Ressourcen

In der Registerkarte Ressourcen verwalten Sie die **Ressourcen** der Schema-Version. Hier sehen Sie die Ressourcenverwaltung:

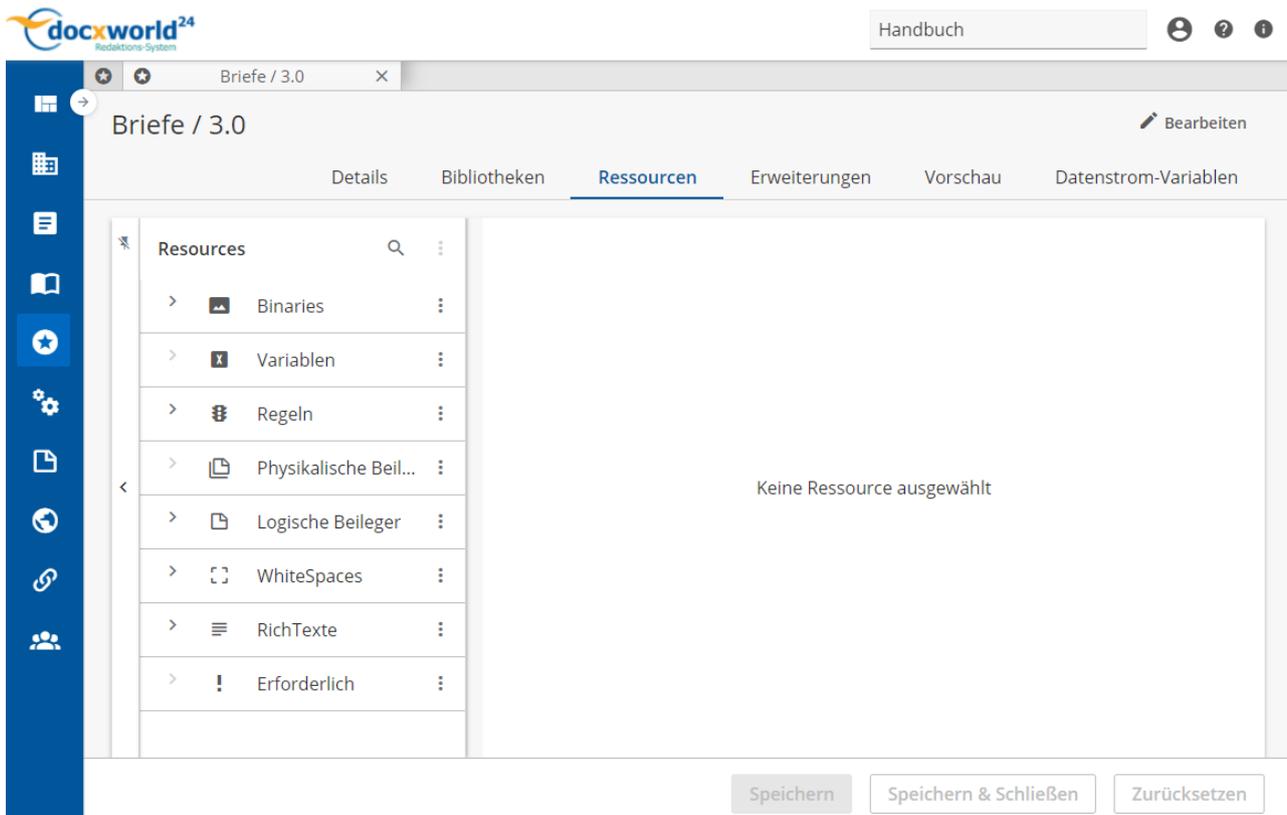


Abbildung A

In der Ressourcenverwaltung haben Sie folgende Möglichkeiten:

- Sie können Ressourcen folgenden Typs  **erstellen**:
  -  **-Binäre Ressourcen**
  -  **Variablen**
  -  **Regeln**
  -  **Physikalische Beileger**
  -  **Logische Beileger**
  -  **WhiteSpaces**
  -  **RichTexte**
- Sie können Ressourcen vom folgenden Typ **einsehen**:
  -  **Datenstrom-Variablen**
  -  **Erforderliche Ressourcen (der eingebundenen Bibliotheken)**
- Sie können bestehende Ressourcen **bearbeiten**,  **duplizieren** und  **löschen**.
- Sie können Ressourcen  **exportieren** und  **importieren**.

Die Ressourcenverwaltung wird in einem gesonderten Kapitel behandelt, da sie in Bibliotheken und Schemas nahezu identisch ist. Das Kapitel über die Ressourcenverwaltung finden Sie hier (wir empfehlen allerdings, erst die Kapitel über Schema- und Bibliotheksversionen zu lesen):

[Zur Ressourcenverwaltung](#)

 Zum Verändern der Inhalte müssen Sie vorher den Bearbeitungsmodus (  Bearbeiten) aktivieren (nur im Status Edit möglich).

In der Registerkarte Ressourcen verwalten Sie die **Ressourcen** der Schema-Version. Hier sehen Sie die Ressourcenverwaltung:

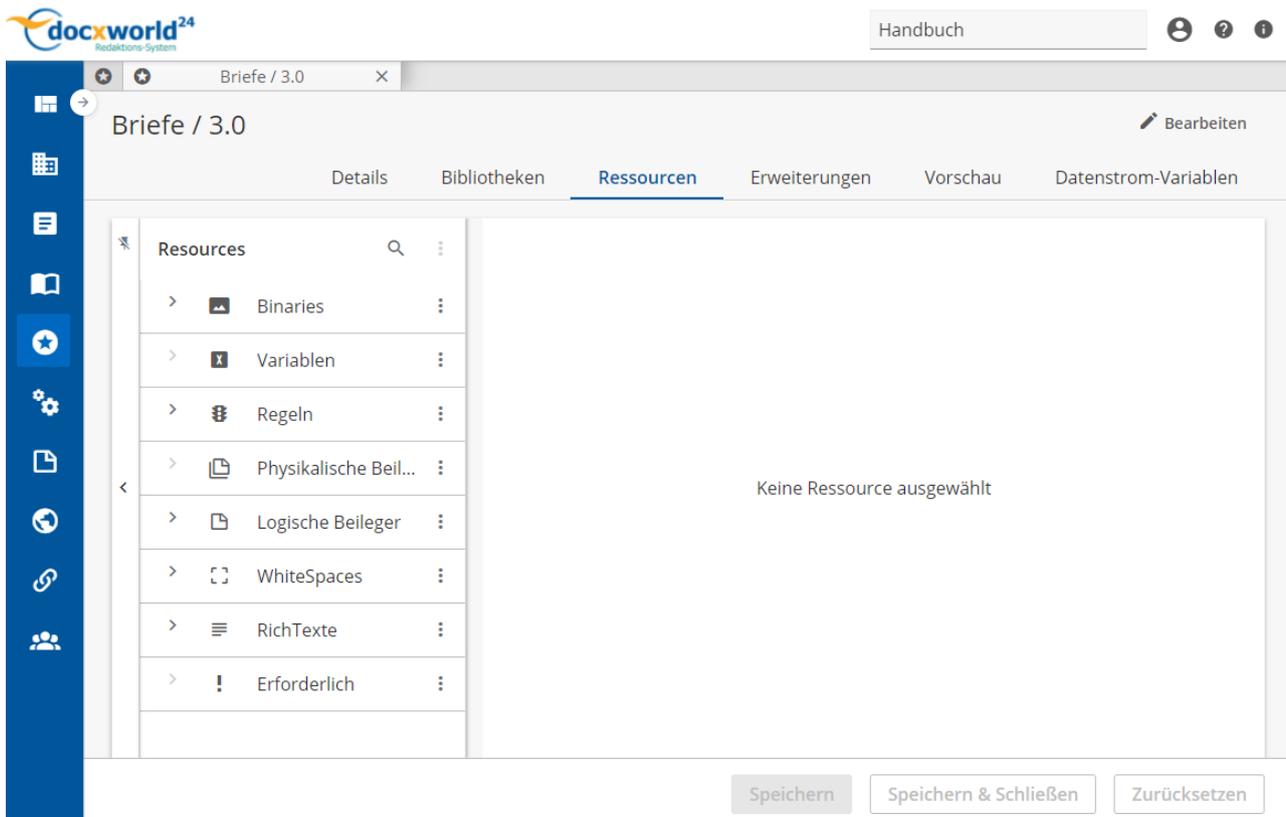


Abbildung A

In der Ressourcenverwaltung haben Sie folgende Möglichkeiten:

- Sie können Ressourcen folgenden Typs  **erstellen**:
  -  **-Binäre Ressourcen**
  -  **Variablen**
  -  **Regeln**
  -  **Physikalische Beileger**
  -  **Logische Beileger**
  -  **WhiteSpaces**
  -  **RichTexte**
- Sie können Ressourcen vom folgenden Typ **einsehen**:
  -  **Datenstrom-Variablen**
  -  **Erforderliche Ressourcen (der eingebundenen Bibliotheken)**
- Sie können bestehende Ressourcen **bearbeiten**,  **duplizieren** und  **löschen**.
- Sie können Ressourcen  **exportieren** und  **importieren**.

Die Ressourcenverwaltung wird in einem gesonderten Kapitel behandelt, da sie in Bibliotheken und Schemas nahezu identisch ist. Das Kapitel über die Ressourcenverwaltung finden Sie hier (wir empfehlen allerdings, erst die Kapitel über Schema- und Bibliotheksversionen zu lesen):

[Zur Ressourcenverwaltung](#)

 Zum Verändern der Inhalte müssen Sie vorher den Bearbeitungsmodus (  Bearbeiten) aktivieren (nur im Status Edit möglich).

## Registerkarte - Erweiterungen

In dieser Registerkarte können Sie **Erweiterungen** erstellen, die es Ihnen ermöglichen, Ressourcen und Materialien in das Dokument einzubinden. Die Erweiterungen bestehen aus den Seiten des zugeordneten Transaktions-Formulars, physikalischen Belegern, logischen Belegern und WhiteSpaces *Abb. A (1)*.

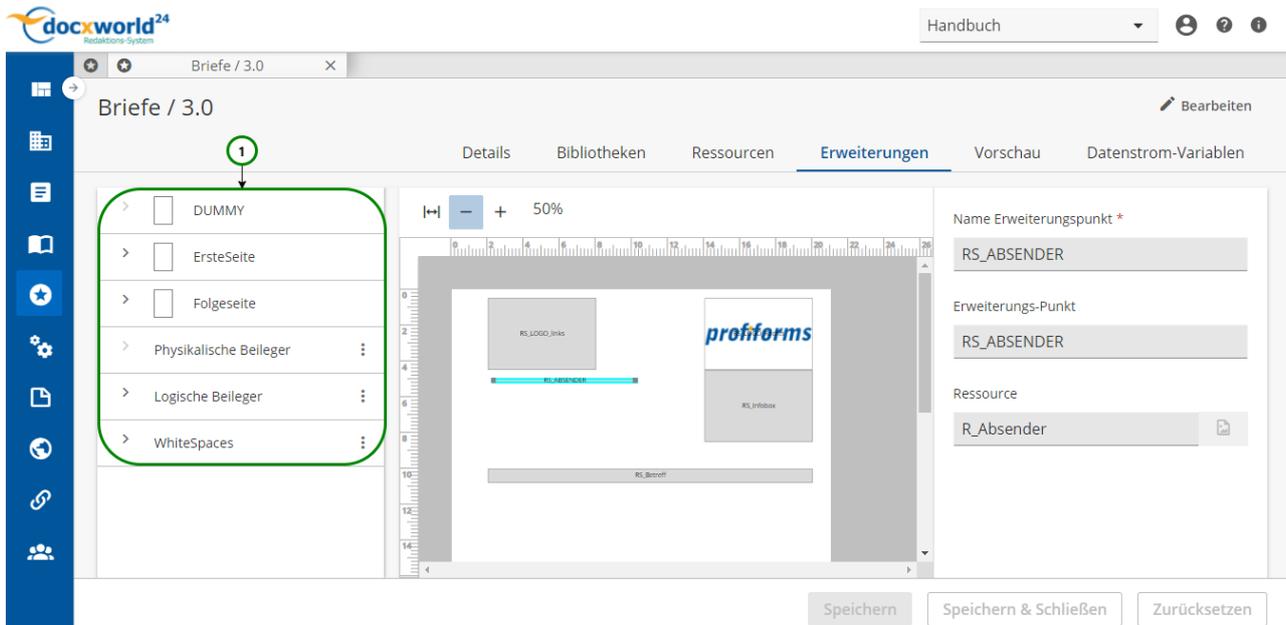


Abbildung A

⚠️ Zum Verändern der Inhalte müssen Sie vorher den Bearbeitungsmodus (  Bearbeiten) aktivieren (nur im Status Edit möglich).

## Seitendesign-Ansicht

Sie haben in der Erweiterungs-Ansicht die Möglichkeit, das Design der Dokumentenseiten inklusive deren Erweiterungs-Punkte einzusehen. Sie finden dazu auf der linken Seite der Registerkarte Erweiterungen die Seiten-Übersicht *Abb. A (1)*.

Wählen Sie die Seite in der Seiten-Übersicht (1) aus, für die Sie das Design einsehen möchten. Ihnen wird dann das Design der Seite in der rechten Übersicht (2) angezeigt:

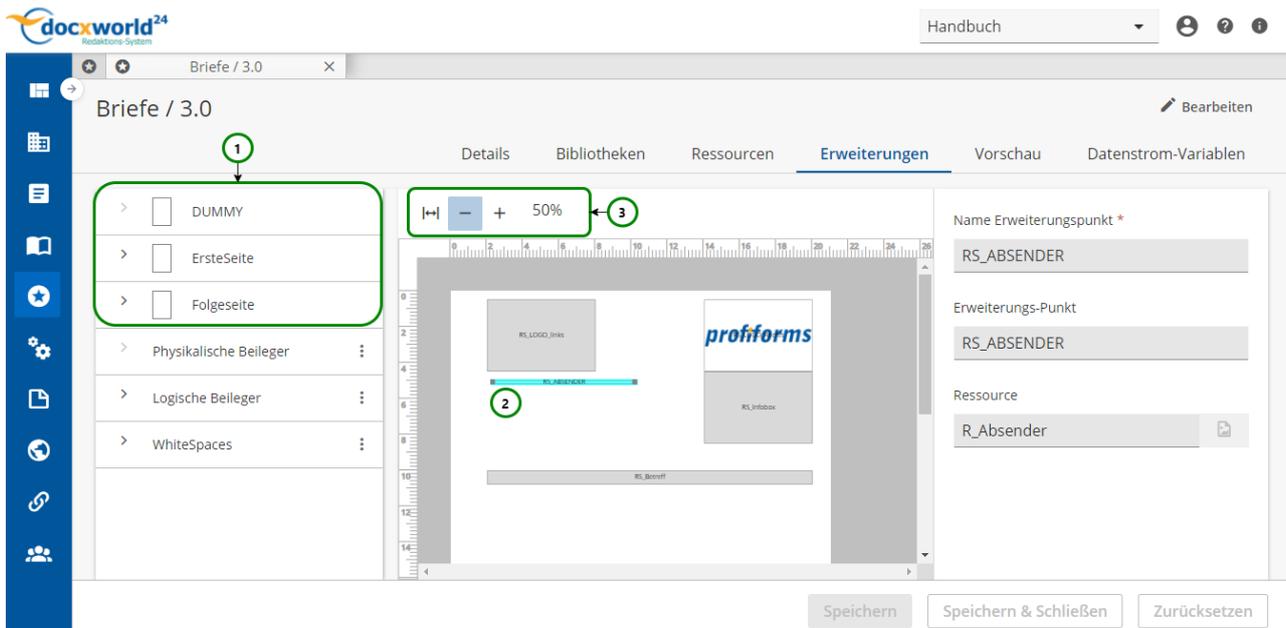


Abbildung A

### Sie sehen jetzt:

- Das **Design** der in *Abb. A (1)* ausgewählten Seite
- Die **Erweiterungs-Punkte** der Seite (2)
- Die Zoom Einstellungen (3)

Sie sehen die Position der Erweiterungs-Punkte im Dokument, in die Sie Ressourcen einfügen können. Außerdem wird der Name des Erweiterungs-Punktes, den Sie beim Erstellen einer Erweiterung angeben müssen angezeigt und, falls der Erweiterungs-Punkt bereits mit einer Ressource verknüpft worden ist, der Ressource-Typ der zugewiesenen Ressource (als **Symbol**).

Mit Hilfe von *Abb. A (3)* können Sie die Seite vergrößern und verkleinern. Um zum Design einer anderen Seite zu wechseln, klicken Sie auf die gewünschte Seite in (1).

## Erweiterung für Seiten verknüpfen, ändern und löschen

An Seitenbezogenen Erweiterungs-Punkten können Sie Ressourcen verknüpfen, die zur Laufzeit in den Erweiterungs-Punkt hineingeladen werden. Die verfügbaren Erweiterungs-Punkte, an die Sie eine Ressource binden können, werden durch das Transaktions-Formular definiert. Falls Sie sich noch nicht mit Erweiterungen vertraut gemacht haben, können Sie das [hier](#) tun.

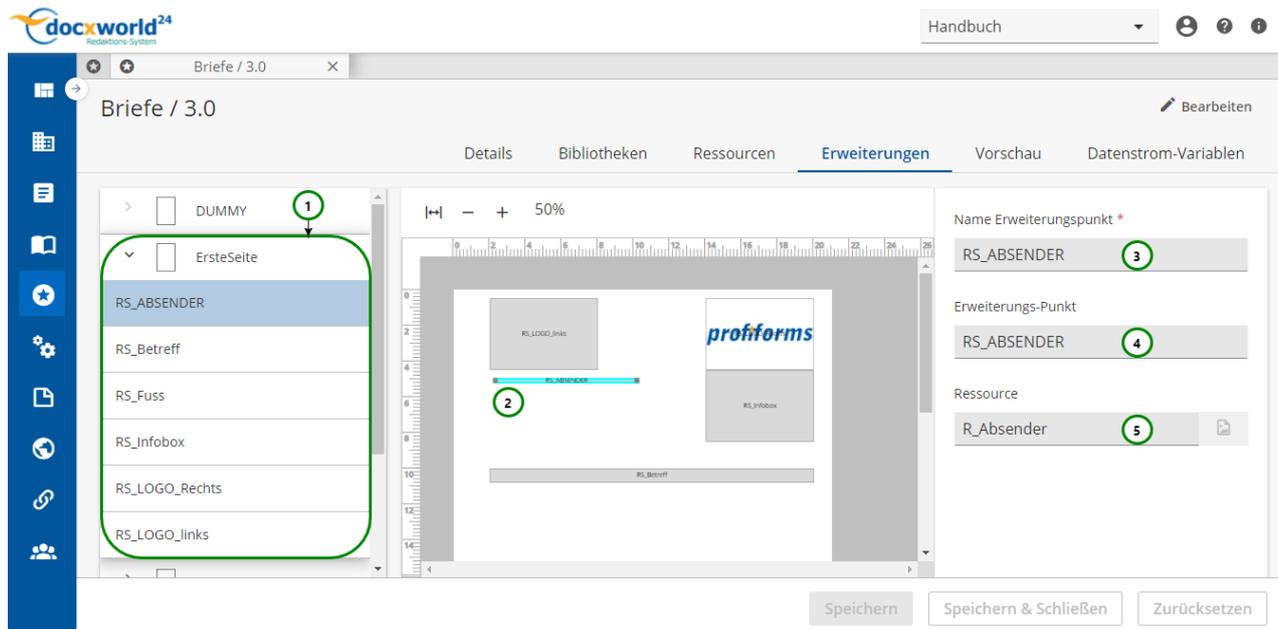


Abbildung A

Wählen Sie aus der Erweiterungsübersicht *Abb. A* (1) oder aus der Seitenansicht (2) einen Erweiterungs-Punkt aus. Auf der Rechten Seite sehen Sie nun die Informationen zu dem Erweiterungs-Punkt (3,4,5).

### Sie sehen :

- Den Namen des Erweiterungs-Punkt, der im Schema festgelegt werden kann *Abb. A* (3).
- Den Namen des Erweiterungs-Punkt, der im Transaktionsformular festgelegt wurde (4).
- Die momentan verlinkte Ressource (5).

Den Name des Erweiterungs-Punkt, der im Schema festgelegt werden kann *Abb. A* (3), können Sie nun frei definieren. Dieser kann verwendet werden, um den Erweiterungs-Punkt mit einem fachlichen Bezeichner zu versehen.

### Ressource verlinken

Sie können Ressourcen an einen Erweiterungs-Punkt auf der Seite zuweisen. Bevor Sie eine Ressource zuweisen, müssen Sie diese im Schema erst erstellen (oder in einer in das Schema eingebundenen **Bibliothek**). Klicken Sie auf *Abb. A* (5), anschließend öffnet sich folgendes Fenster:

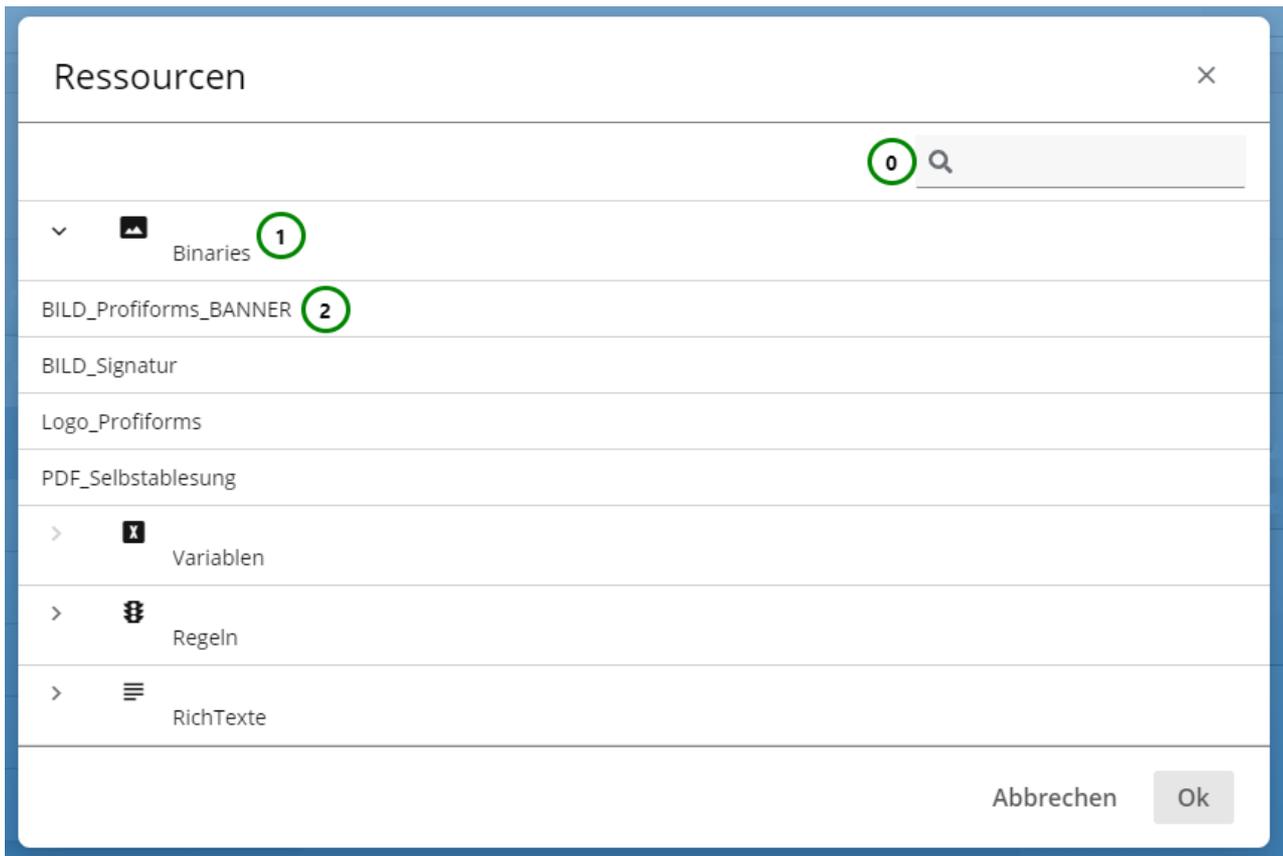


Abbildung B

**In *Abb. B* sehen Sie:**

- Die verfügbaren **Ressource-Typen** (1).
- Die verfügbaren **Ressourcen** (vom ausgewählten Ressource-Typ) (2).

Wählen Sie aus *Abb. B* (1) einen Ressource-Typ aus, um die Ressourcen (2) zu diesem Typ anzeigen zu lassen. Mit Hilfe von  (0) können Sie außerdem die Ressourcen durchsuchen.

Wählen Sie dann aus *Abb. B* (2) die Ressource aus, die Sie an den Erweiterungs-Punkt binden möchten und klicken Sie anschließend auf Ok.

**Ressourcen entfernen**

Um eine Ressourcenzuweisung zu entfernen, wählen Sie den Erweiterungs-Punkt *Abb. A* (1) oder (2) aus und benutzen Sie  (5).

Speichern Sie die Schema-Version, um die Erweiterung endgültig zu löschen oder verlassen Sie das Schema ohne zu Speichern, um die Löschung rückgängig zu machen.

## Erweiterung für physikalische Beileger erstellen, ändern und löschen

Mit Erweiterungen für physikalische Beileger weisen Sie den Dokumenten einen physikalischen Beileger zu. Um eine neue Erweiterung zu erstellen, wählen Sie aus der Übersicht links Abb. A (1) Physikalische Beileger aus und klicken Sie dann auf das Kontextmenü  (1). Anschließend können Sie mit Neu  (2), einen neuen Erweiterungs-Punkt für einen physikalischen Beileger erstellen.

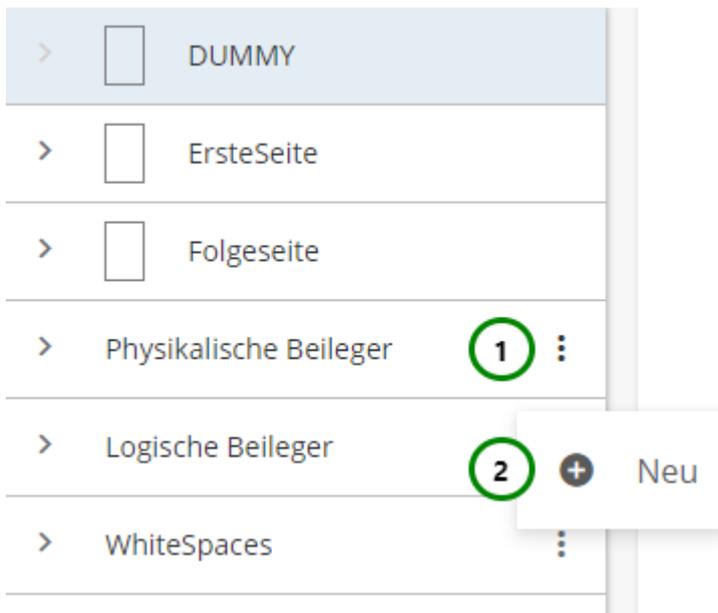


Abbildung A

### Sie können der Erweiterung für physikalische Beileger:

- ★ einen **Namen** geben Abb. B (2).  
Beim Erstellen müssen Sie in das Feld etwas eingeben, da die Erweiterung noch keinen Namen hat. Beim Ändern muss das Feld nicht zwingend geändert werden.
- ★ sie als **optionalen Beileger** festlegen (3).  
Als optional gekennzeichnete Beileger werden nur angefügt, wenn der Beileger bspw. keinen Portosprung verursacht. Andernfalls wird der Beileger weggelassen. Ist ein Beileger nicht optional, wird er in jedem Fall angefügt (außer die Bedingung ist false).
- ★ eine **Bedingung** geben (4).  
Beileger werden den Dokumenten nur dann angefügt, wenn die Bedingung wahr (= true) ist. Als Bedingung können z.B. EVAL-Anweisungen definiert werden. Bedingungen müssen einen booleschen Wert (true oder false, wahr oder falsch, 1 oder 0) zurückgeben.
- ★ eine **Ressource** zuordnen.  
Um einen Beileger zuzuordnen, nutzen Sie bitte Selektieren (5). Mit  (5) können Sie einen zugeordneten Beileger wieder entbinden. Die Ressource muss ein physikalischer Beileger sein. Es ist allerdings möglich, eine Variable oder Regel anzugeben, die einen physikalischen Beileger zurückgibt, um so z.B. durch Regeln die Beilage gezielt zu steuern.

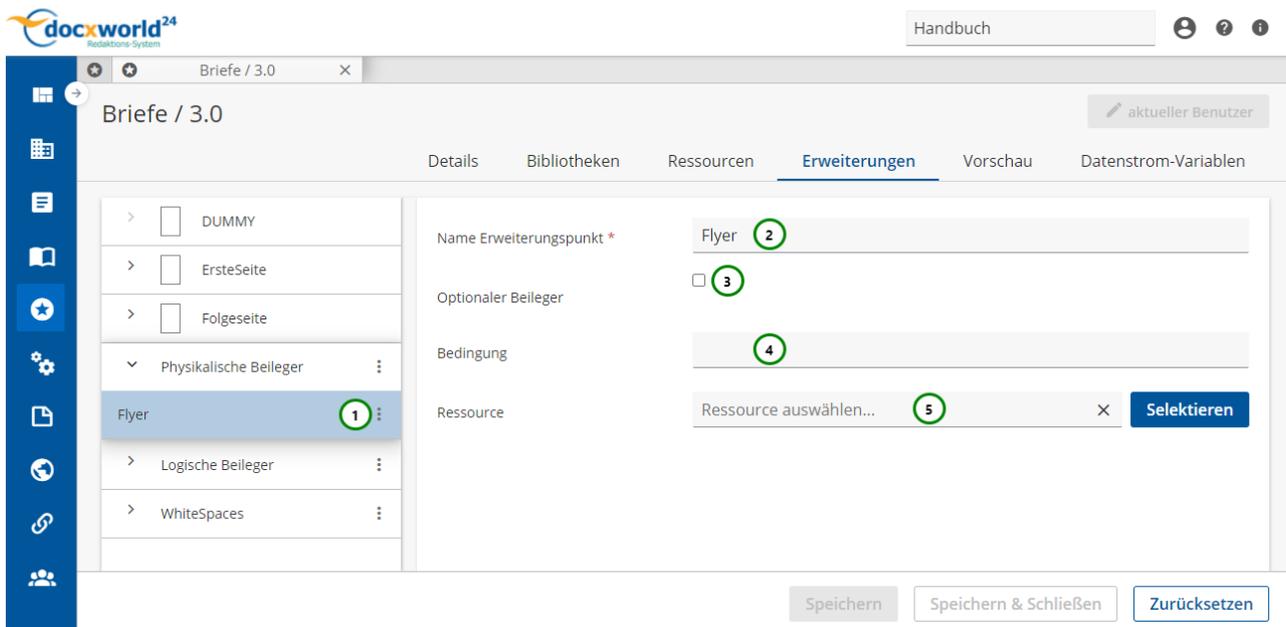


Abbildung B

Sie müssen der Erweiterung natürlich einen Beileger zuweisen, der den Dokumenten angefügt werden soll. Bevor Sie eine Beileger zuweisen, müssen Sie diesen in der Schema-Version erst **erstellen** (oder in einer in die Schema-Version eingebundenen **Bibliothek**). Wenn Sie zur Zuweisung eines Beilegers auf Selektieren Abb. B (5) klicken, öffnet sich folgender Dialog:

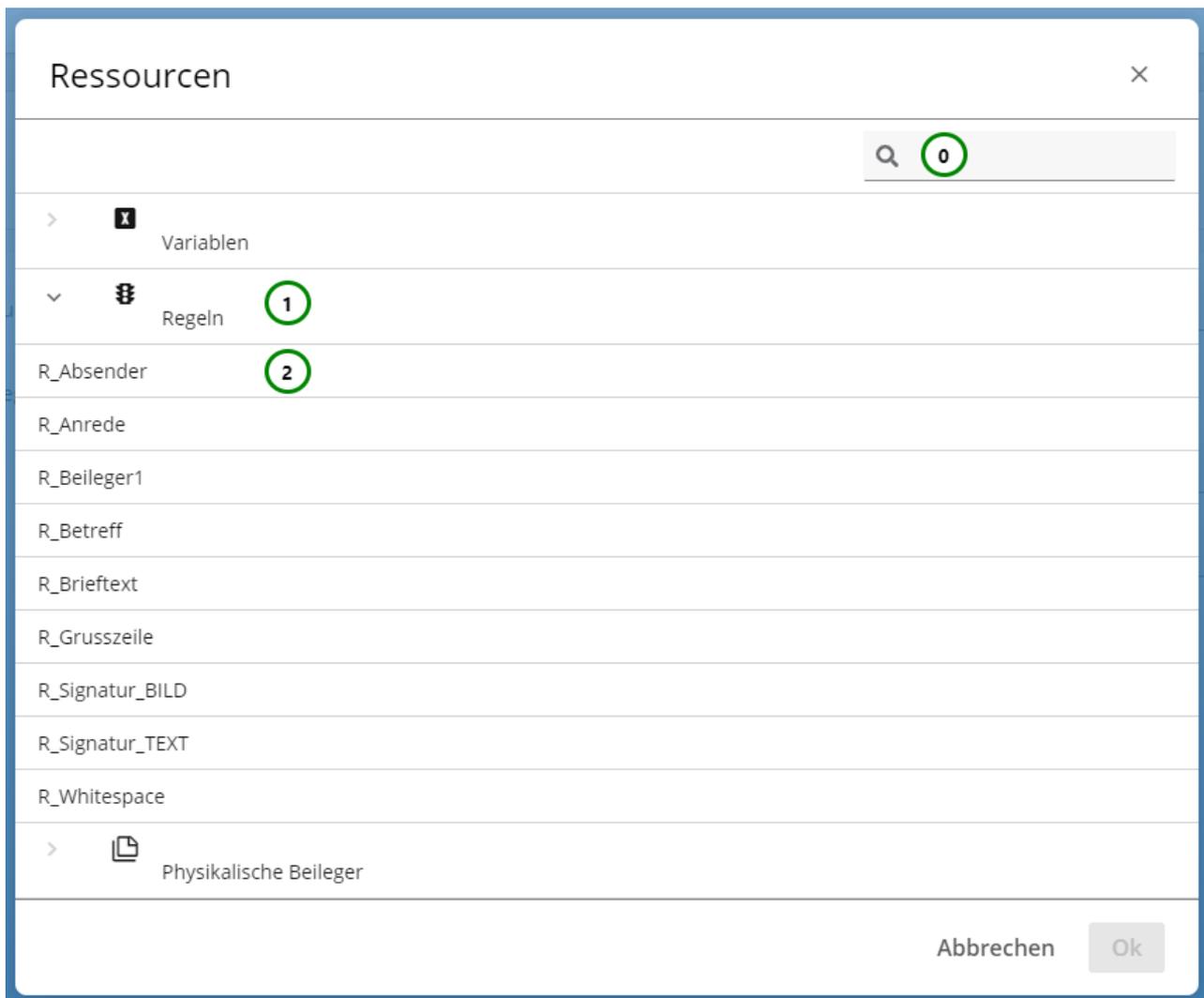


Abbildung C

In **Abb. C** sehen Sie:

- Die Suche (0)
- Die verfügbaren **Ressource-Typen** (1)
- Die verfügbaren **Ressourcen** (3) (vom ausgewählten Ressource-Typ)

Wählen Sie aus *Abb. C (1)* einen Ressource-Typ aus, um die Ressourcen (2) zu diesem Typ anzeigen zu lassen. Mit Hilfe von  (0) können Sie außerdem die Ressourcen durchsuchen.

Wählen Sie dann aus *Abb. C (2)* die Ressource aus, die Sie für die Beileger-Erweiterung verwenden möchten. Achten Sie darauf, dass es sich entweder um einen physikalischen Beileger handelt oder um eine Ressource, die einen physikalischen Beileger zurückgibt und klicken Sie anschließend auf Ok, um die Zuweisung abzuschließen oder auf Abbrechen, um den Vorgang abzubrechen.

### Erweiterung löschen

Um eine Erweiterung für einen physikalischen Beileger zu löschen, wählen Sie die gewünschte Erweiterung aus und klicken Sie anschließend auf das Kontextmenü  *Abb. D (1)*.

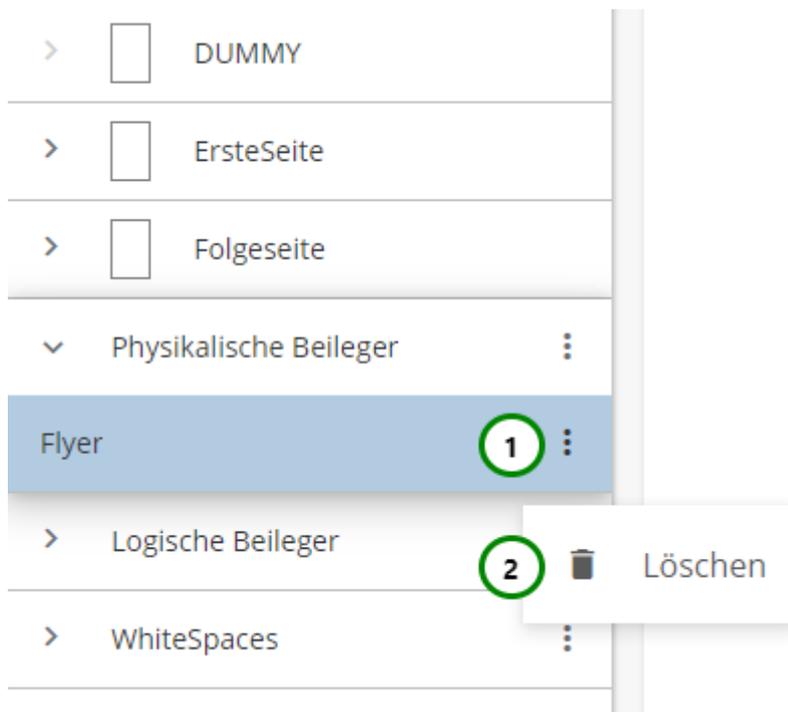


Abbildung D

Klicken Sie nun auf  (2), um das Löschen zu bestätigen.

Speichern Sie das Schema, um die Erweiterung endgültig zu löschen oder verlassen Sie das Schema ohne zu Speichern, um die Löschung rückgängig zu machen.

## Erweiterung für logische Beileger erstellen, ändern und löschen

Mit Erweiterungen für logische Beileger weisen Sie den Dokumenten einen logischen Beileger zu.

Um eine neue Erweiterung zu erstellen, wählen Sie aus der Übersicht links *Abb. A (1)* Logische Beileger aus und klicken Sie dann auf das Kontextmenü  (1). Anschließend können Sie mit Neu  (2), einen neuen Erweiterungs-Punkt für einen logischen Beileger erstellen.

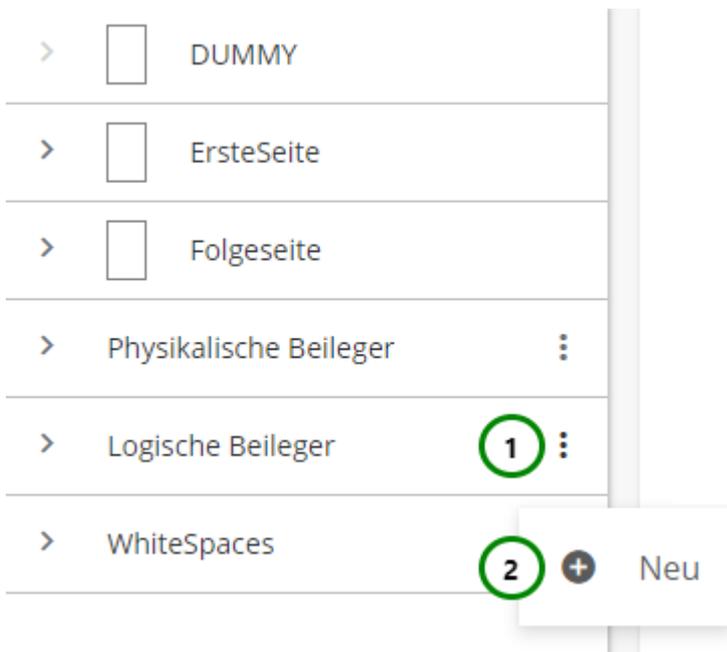


Abbildung A

### Sie können der Erweiterung für logische Beileger:

- ★ einen **Namen** geben *Abb. B (2)*.  
Beim Erstellen müssen Sie in das Feld etwas eingeben, da der Beileger noch keinen Namen hat. Beim Ändern muss das Feld nicht zwingend geändert werden
- ★ sie als **optionalen Beileger** festlegen (3).  
Als optional gekennzeichnete Beileger werden nur angefügt, wenn der Beileger bspw. keinen Portosprung verursacht. Andernfalls wird der Beileger weggelassen. Ist ein Beileger nicht optional, wird er in jedem Fall angefügt (außer die Bedingung ist false).
- ★ eine **Bedingung** geben (4).  
Beileger werden den Dokumenten nur dann angefügt, wenn die Bedingung wahr (= 1) ist. Als Bedingung können z.B. EVAL-Anweisungen definiert werden. Bedingungen müssen einen booleschen Wert (true oder false, wahr oder falsch, 1 oder 0) zurückgeben.
- ★ eine **Ressource** zuordnen.  
Um einen Beileger zuzuordnen, nutzen Sie bitte Selektieren *Abb. B (5)*. Mit  (5) können Sie einen zugeordneten Beileger wieder entbinden. Die Ressource muss ein logischer Beileger sein. Es ist allerdings möglich, eine Variable oder Regel anzugeben, die einen logischen Beileger zurückgibt, um so z.B. durch Regeln die Beilage gezielt zu steuern.

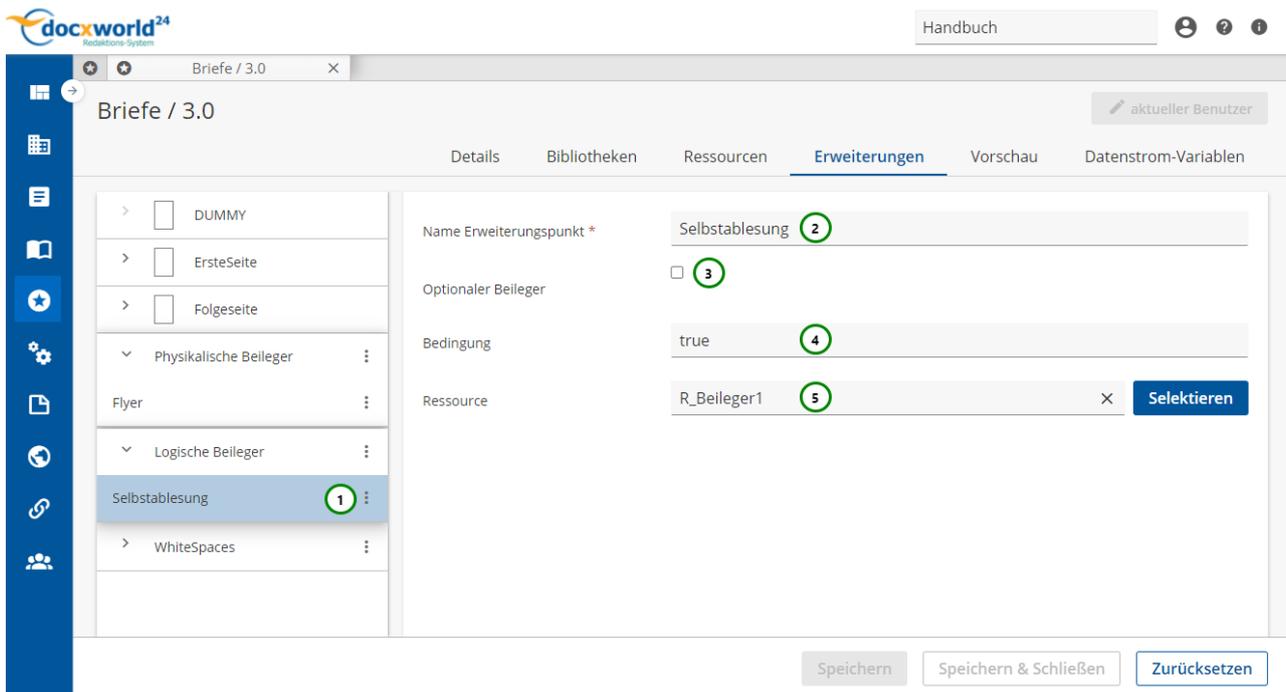


Abbildung B

Sie müssen der Erweiterung natürlich einen Beileger zuweisen, der den Dokumenten angefügt werden soll. Bevor Sie eine Beileger zuweisen, müssen Sie diesen in der Schema-Version erst **erstellen** (oder in einer in die Schema-Version eingebundenen **Bibliothek**). Wenn Sie zur Zuweisung eines Beilegers auf **Selektieren** Abb. B (5) klicken, öffnet sich folgender Dialog:

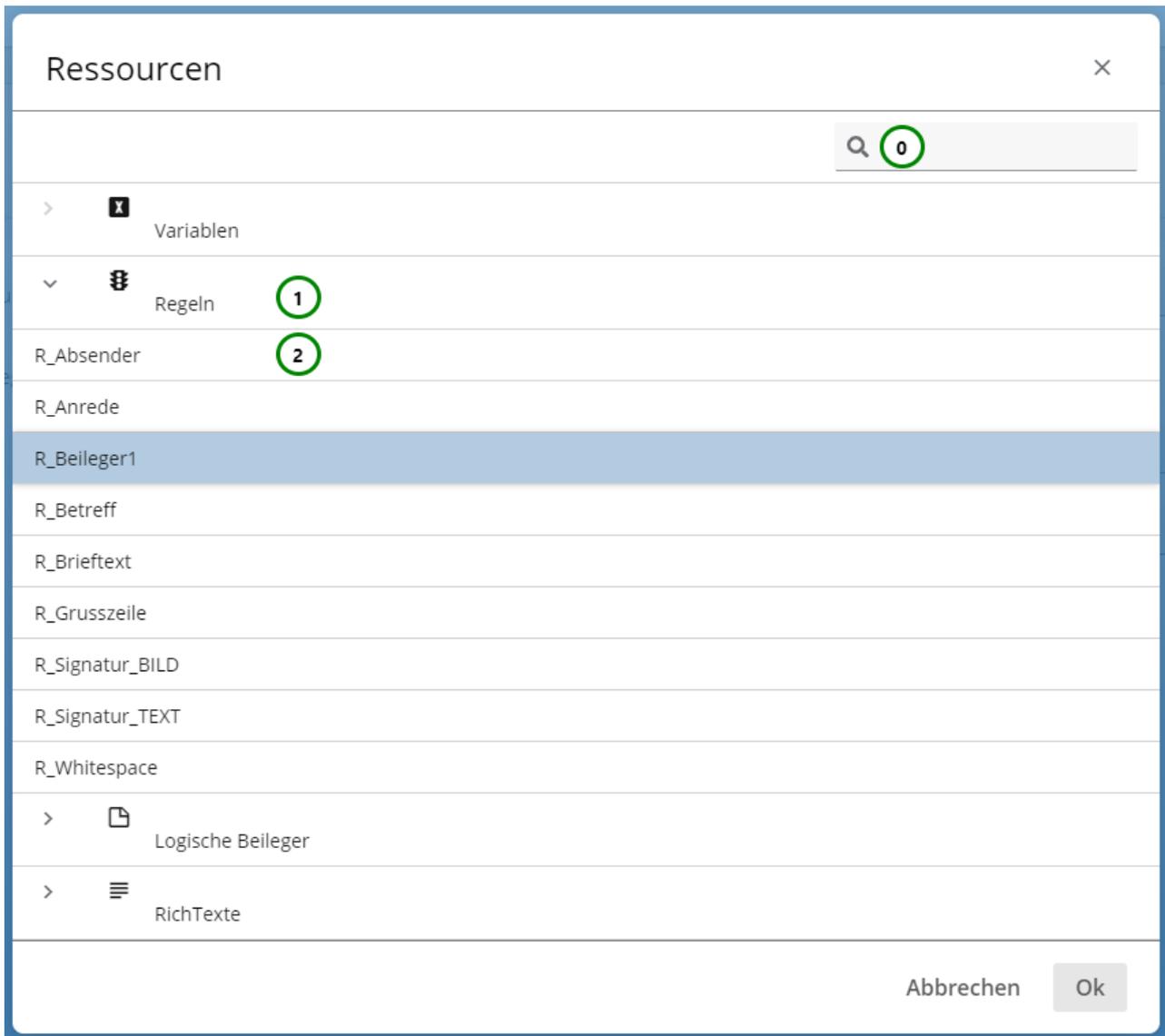


Abbildung C

In **Abb. C** sehen Sie:

- Die Suche (0)
- Die verfügbaren **Ressource-Typen** (1),
- Die verfügbaren **Ressourcen** (vom ausgewählten Ressource-Typ) (2),

Wählen Sie aus **Abb. C** (1) einen Ressource-Typ aus, um die Ressourcen (2) zu diesem Typ anzeigen zu lassen. Mit Hilfe von  (0) können Sie außerdem die Ressourcen durchsuchen.

Wählen Sie dann aus *Abb. C* (2) die Ressource aus, die Sie für die Beileger-Erweiterung verwenden möchten. Achten Sie darauf, dass es sich entweder um einen physikalischen Beileger handelt oder um eine Ressource, die einen physikalischen Beileger zurückgibt und klicken Sie anschließend auf Ok, um die Zuweisung abzuschließen oder auf Abbrechen, um den Vorgang abzubrechen.

### **Erweiterung löschen**

Um eine Erweiterung für einen physikalischen Beileger zu löschen, wählen Sie die gewünschte Erweiterung aus und klicken Sie anschließend auf das Kontextmenü  (1).

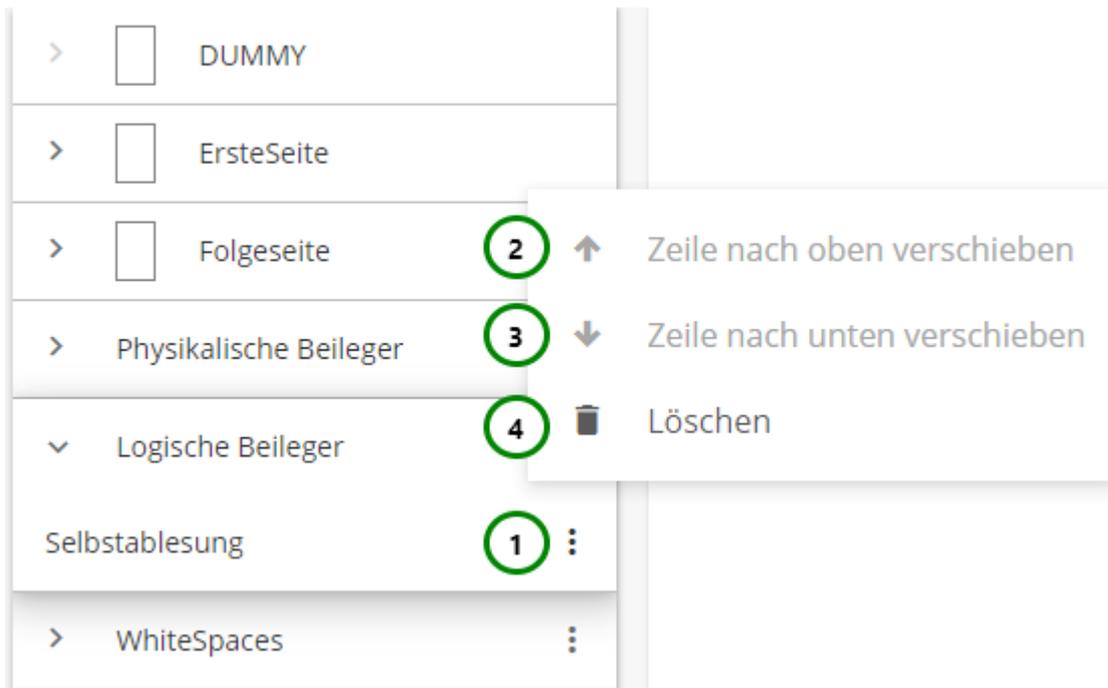


Abbildung D

Klicken Sie nun auf  (4), um das Löschen zu bestätigen. Speichern Sie das Schema, um die Erweiterung endgültig zu löschen oder verlassen Sie das Schema ohne zu Speichern, um die Löschung rückgängig zu machen.

### **Erweiterung sortieren**

Bei Logischen Beilegern bestimmt die Reihenfolge der Erweiterungs-Punkte auch dessen auftreten im Dokument. Die Prüfung geschieht von oben nach unten. Logische Beileger, die über anderen stehen, werden, wenn die Bedingung wahr ist, als erstes eingefügt.

Um die Reihenfolge zu Ändern, können Sie über das Kontextmenü  (1) Die Zeile nach oben  (2) oder nach unten  (3) verschieben.

## Erweiterung für WhiteSpaces erstellen, ändern und löschen

Mit Erweiterungen für WhiteSpaces weisen Sie den Dokumenten einen WhiteSpace zu.

Um eine neue Erweiterung zu erstellen, wählen Sie aus der Übersicht links *Abb. A (1)* WhiteSpaces aus und klicken Sie dann auf das Kontextmenü  (1). Anschließend können Sie mit Neu  (2), einen neuen Erweiterungs-Punkt für einen WhiteSpace erstellen.

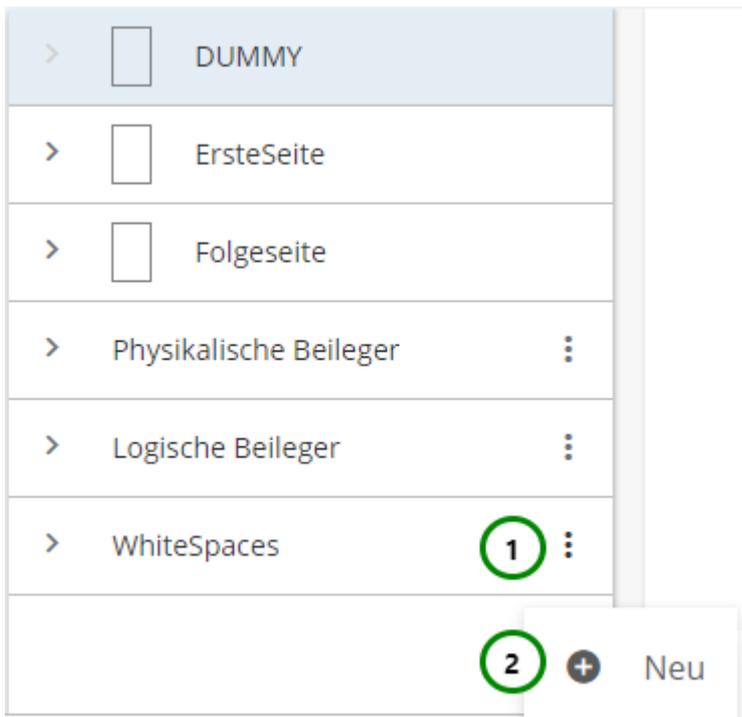


Abbildung A

**Sie können der Erweiterung für WhiteSpaces:**

- ★ einen **Namen** geben *Abb. B (2)*.

Beim Erstellen müssen Sie in das Feld etwas eingeben, da das Formular noch keinen Namen hat. Beim Ändern muss das Feld nicht zwingend geändert werden.

- ★ eine **Position** auf dem **Dokument** geben. Klicken Sie auf (3). Es klappt ein Auswahlménü aus:

Position auf Dokument \*

A dropdown menu with a grey header containing the text 'LAST' and a downward arrow. Below the header, the menu is open, showing four options: 'FIRST', 'LAST', and 'ANY'.

Wählen Sie eine Position aus.

**i** Mit der Position des WhiteSpace auf dem Dokument bestimmen Sie, auf welcher Seite des Dokuments der WhiteSpace platziert werden darf:

**FIRST** - Auf der 1. Seite.

**LAST** - Auf der letzten Seite.

**ANY** - Auf einer beliebigen Seite.

- ★ eine **Position** auf der Seite geben. Klicken Sie auf (4). Es klappt ein Auswahlménü aus:

Position auf Seite \*

A dropdown menu with a grey header containing the text 'BOTTOM' and a downward arrow. Below the header, the menu is open, showing three options: 'TOP' and 'BOTTOM'.

Wählen Sie eine Position aus.

**i** Mit der Position auf der Seite bestimmen Sie, wo auf der Seite der WhiteSpace platziert werden darf:

**TOP** - Vor dem Content.

**BOTTOM** - Nach dem Content.

- ★ eine **Bedingung** geben (5).

Als Bedingung können Sie einen beliebigen Ausdruck angeben, der einen booleschen Wert zurückgibt (true oder false). Ins Besondere können Sie hier EVAL-Anweisungen benutzen, wie Sie es auch in Regeln können. Gibt der Ausdruck true (1) zurück, wird der WhiteSpace benutzt, gibt er false (0) zurück, wird der WhiteSpace nicht benutzt.

★ eine **Ressource** zuordnen.

Um einen WhiteSpace zuzuordnen, nutzen Sie bitte  (6). Mit  (6) können Sie einen zugeordneten WhiteSpace wieder entbinden. Die Ressource muss ein WhiteSpace sein. Es ist allerdings möglich, eine Variable oder Regel anzugeben, die einen WhiteSpace zurückgibt, um so z.B. durch Regeln die WhiteSpace-Verwendung gezielt zu steuern.

★ **Formular-Teile zuordnen.**

Das Transaktions-Formular kann Formular-Teile für WhiteSpaces vorgeben. Das sind Bereiche auf dem Dokument, in denen WhiteSpaces angebracht werden dürfen. Um den WhiteSpace zu einem Bereich hinzuzufügen müssen Sie den Bereich zu den ausgewählten Formular-Teilen hinzufügen. Klicken Sie das gewünschte Objekt aus der Übersichtstabelle (7) an. Der Formular-Teil ist nun für den WhiteSpace freigegeben.

 Sind keine verfügbaren Formular-Teile vorhanden und das Transaktions-Formular erlaubt WhiteSpaces, dann können WhiteSpaces immer nur für das gesamte Dokument definiert werden.

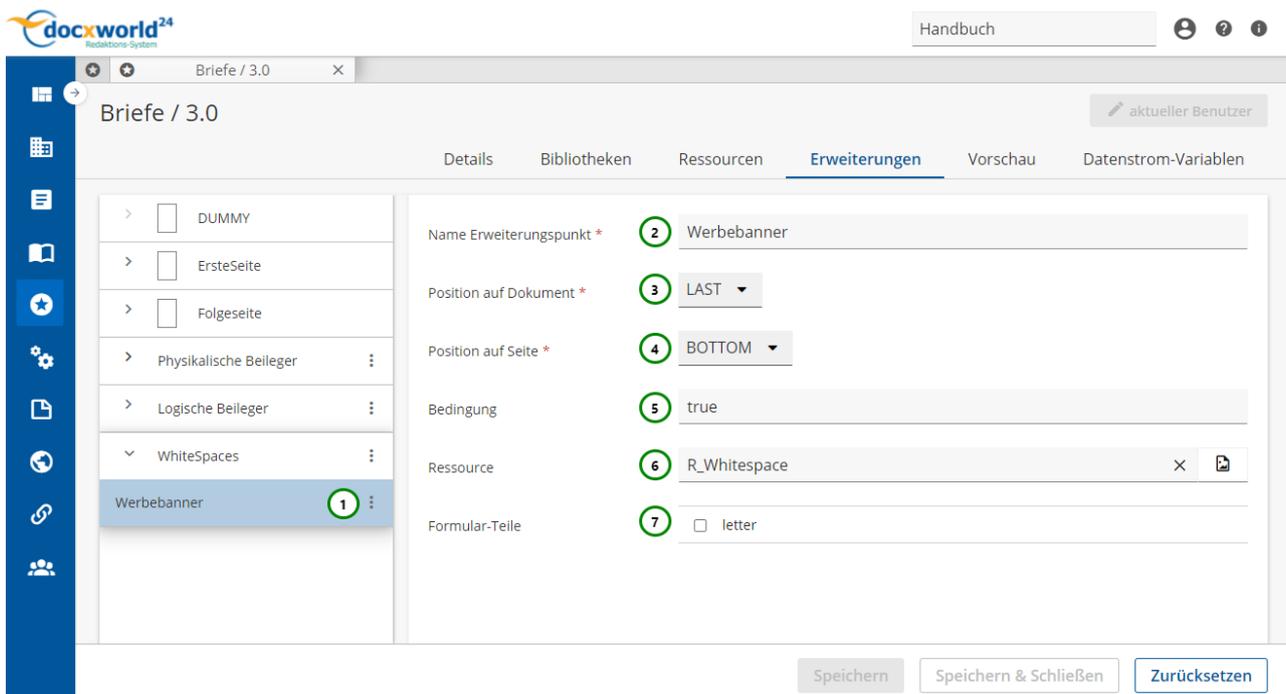


Abbildung B:

**Erweiterung für WhiteSpaces:**

Sie müssen der Erweiterung natürlich einen WhiteSpace/Ressource zuweisen, der/die in das Dokument eingefügt werden soll. Bevor Sie einen WhiteSpace/Ressource zuweisen, müssen Sie ihn in der Schema-Version erst **erstellen** (oder in einer in die Schema-Version eingebundenen **Bibliothek**). Wenn Sie zur Zuweisung einer Ressource auf  Abb. B (6) klicken, öffnet sich folgender Dialog:

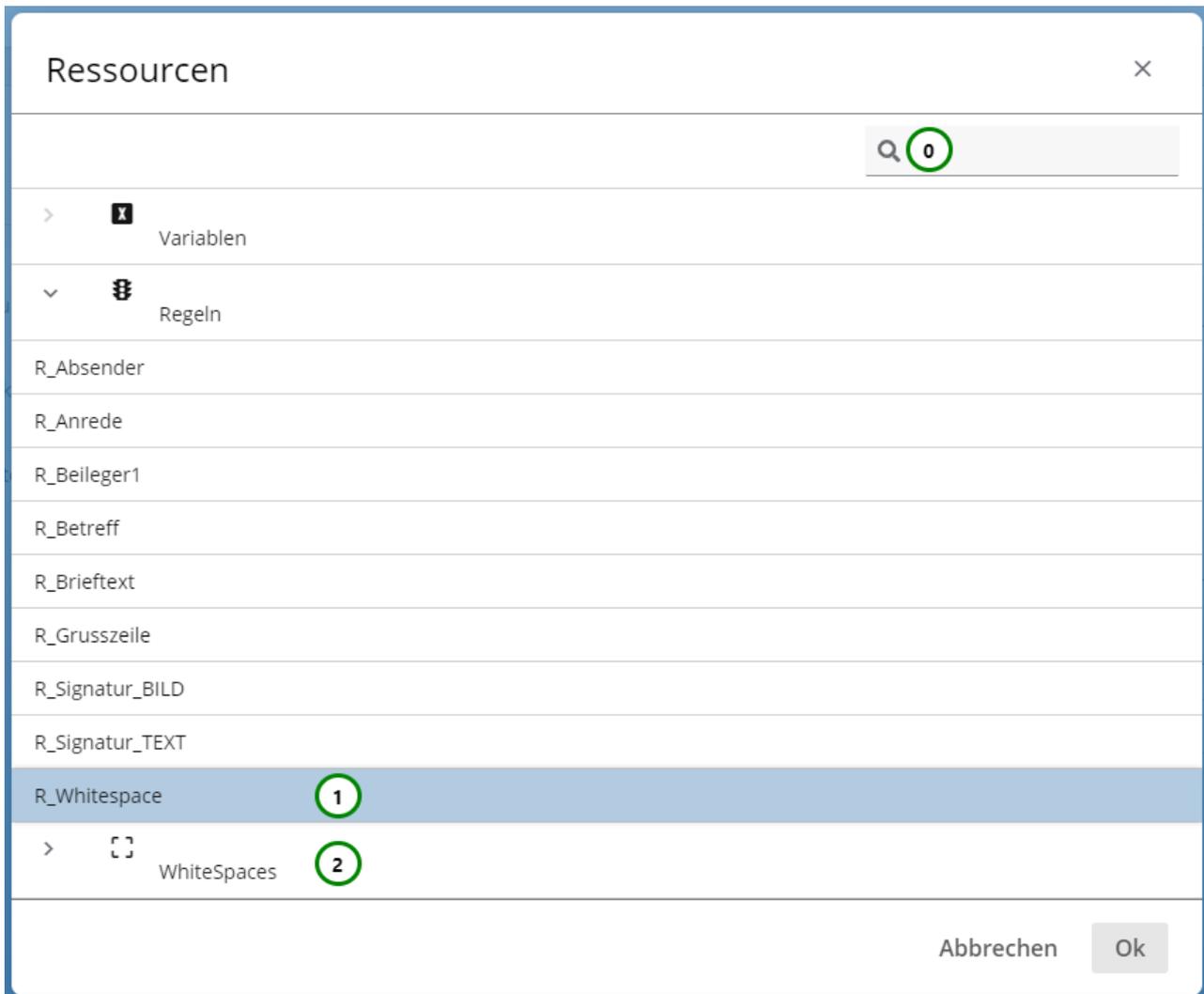


Abbildung C

**In Abb. C sehen Sie:**

- Die verfügbaren **Ressource-Typen**,
- Die verfügbaren **Ressourcen** (vom ausgewählten Ressource-Typ),
- Die **Beschreibung**, die der Ressource beim Erstellen gegeben wurde.

Wählen Sie aus Abb. C (1) einen Ressource-Typ aus, um in der Ressourcen-Liste in (2) nur Ressourcen von diesem Typ anzeigen zu lassen oder lassen Sie Ressourcen ausgewählt, um alle verfügbaren Ressourcen zu sehen. Mithilfe von  (3) können Sie außerdem die Ressourcenliste (2) durchsuchen. Detaillierte Informationen zur Suchfunktion im R-S finden Sie [hier](#). Wählen Sie dann aus (2) die Ressource aus, die Sie für die WhiteSpace-Erweiterung verwenden möchten. Achten Sie darauf, dass es sich entweder um einen WhiteSpace handelt oder um eine Ressource, die einen WhiteSpace zurückgibt und klicken Sie anschließend auf Ok, um die Zuweisung abzuschließen oder auf Abbrechen, um den Vorgang abubrechen.

### WhiteSpace löschen

Um eine Erweiterung für einen WhiteSpace zu löschen, wählen Sie die gewünschte Erweiterung aus und klicken Sie anschließend auf das Kontextmenü  *Abb. D* (1).

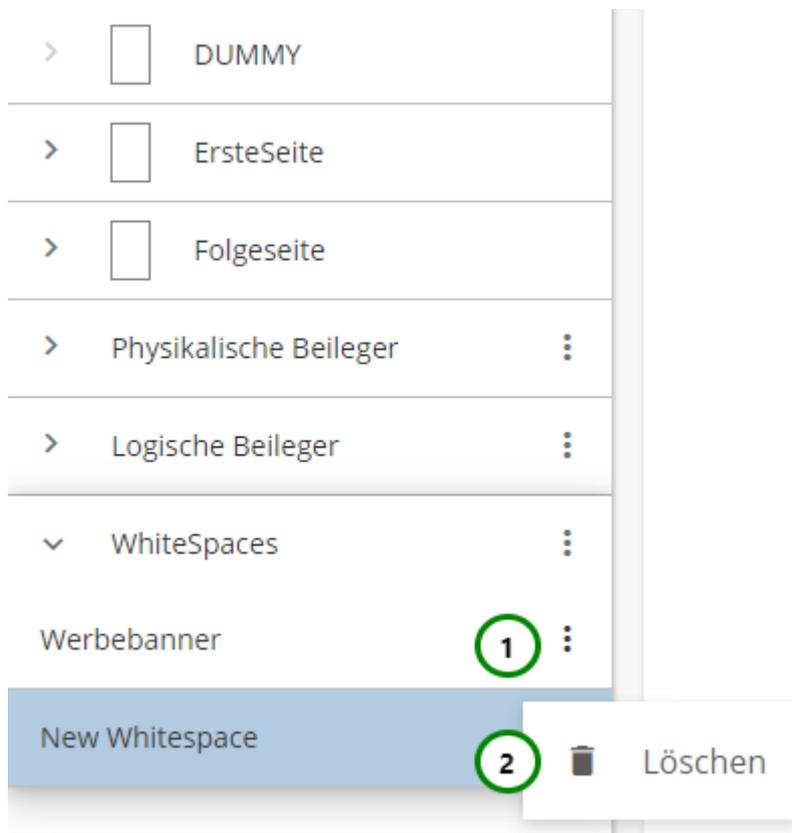


Abbildung D

Klicken Sie nun auf  (2), um das Löschen zu bestätigen. Speichern Sie das Schema, um die Erweiterung endgültig zu löschen oder verlassen Sie das Schema ohne zu Speichern, um die Löschung rückgängig zu machen.

## Konflikte und Markierungen bei Erweiterungen

Bei Erweiterungen können verschiedene Konflikte auftreten. Diese werden Ihnen entweder direkt in der Registerkarte Erweiterungen angezeigt oder per Fehlermeldung beim Speichern / der Dokumenterstellung. Im Folgenden werden die verschiedenen Konflikte und ihre Markierungen aufgelistet.

### 1. Beileger nicht erlaubt

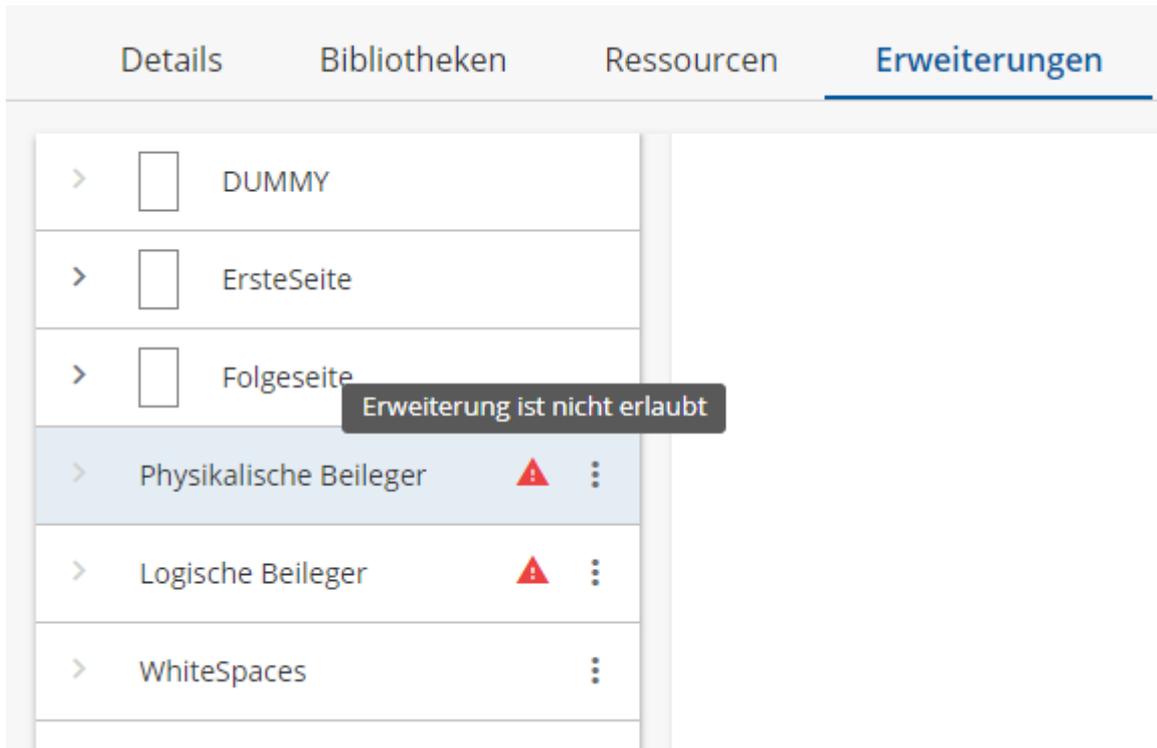


Abbildung A

Welche Beileger erlaubt sind, wird durch das Transaktions-Formular bestimmt. Dabei legt das Formular folgendes fest:

- Ob physikalische Beileger erlaubt sind. Sind sie das nicht, wird Ihnen das mit  in [Abb. A](#) signalisiert.
- Ob logische Beileger erlaubt sind. Sind sie das nicht, wird Ihnen das mit  in [Abb. A](#) signalisiert.
- Ob WhiteSpaces erlaubt sind. Sind sie das nicht, wird Ihnen das mit  in [Abb. A](#) signalisiert.

Es ist zwar möglich, eine Erweiterung für einen verbotenen Beileger zu erstellen, bei der Erstellung der Dokumentenvorschau bricht das R-S jedoch mit einer Fehlermeldung ab.

## 2. Ressource fehlt

Es ist zwar nicht möglich, einer Erweiterung eine nicht vorhandene Ressource zuzuweisen, da bei der Zuweisung nur existierende Ressourcen zur Auswahl angeboten werden und der Nutzer die Ressource nicht manuell angeben kann. Wird die Ressource allerdings nach der Zuweisung gelöscht oder die Bibliothek, in der die Ressource definiert ist, entfernt, wird der Verweis ungültig. Ein ungültiger Verweis wird in Abb. B durch  gekennzeichnet:

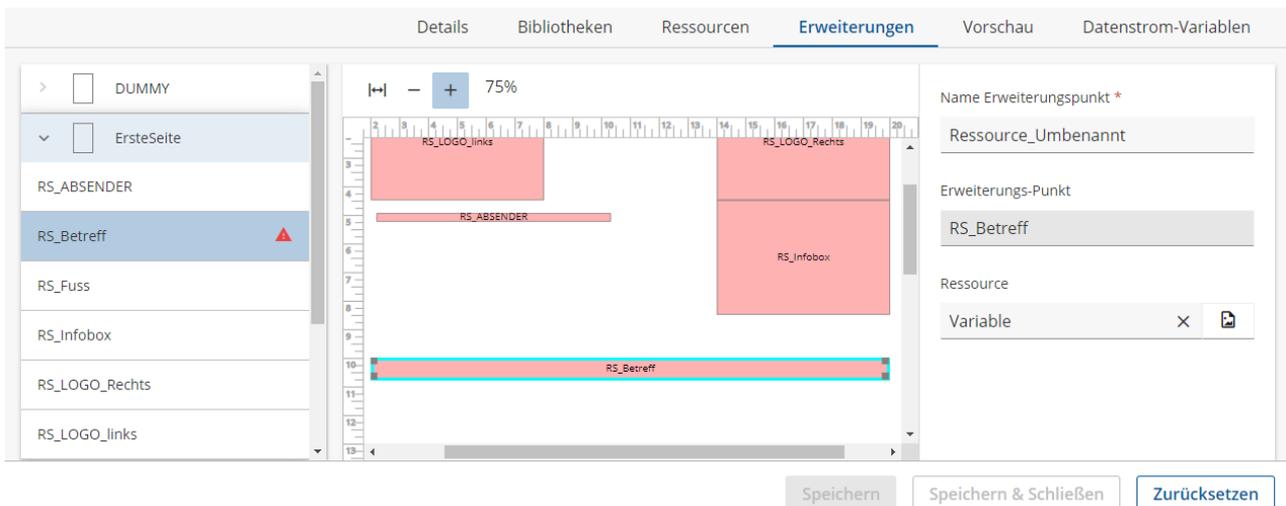


Abbildung B

Sollten Sie eine vorhandene Ressource umbenennen, sehen Sie die Warnung sofort im Reiter Erweiterungen:



Solange dieser Konflikt besteht ist ein Speichern nicht möglich.

Um das Problem zu beheben, müssen Sie eine andere Ressource für die Erweiterung angeben. Mehr Informationen dazu finden Sie [hier](#).

## Registerkarte - Vorschau in Schema-Versionen

In dieser Registerkarte können Sie sich mit Hilfe von Testdaten eine Vorschau generieren lassen.

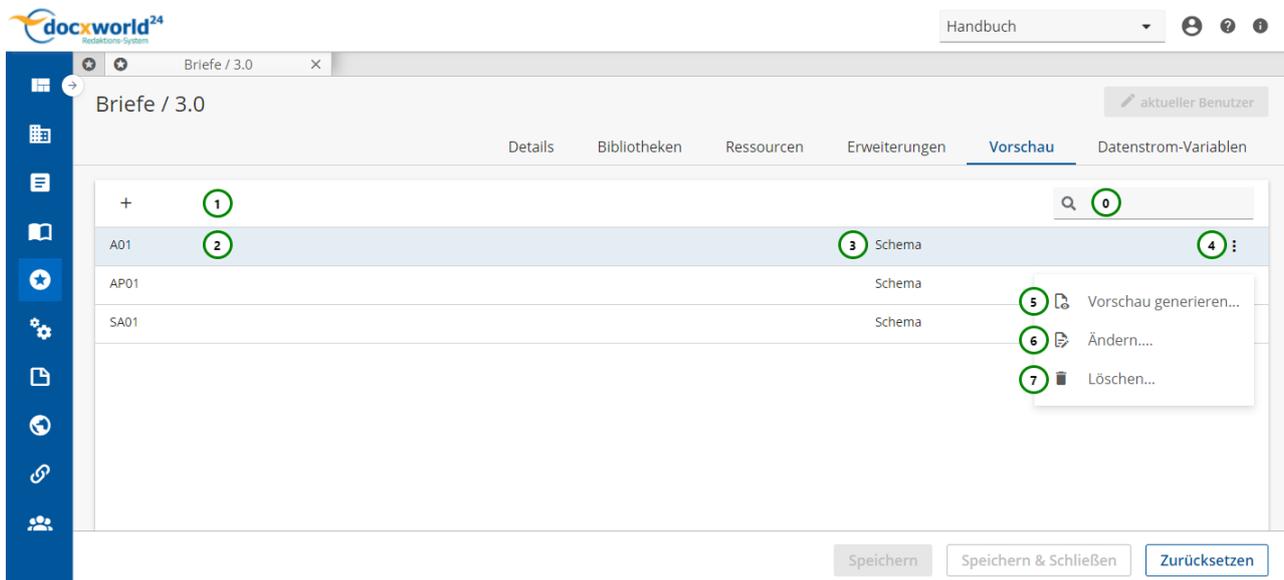


Abbildung A

### ⚠ Bei den Testdaten wird unterschieden zwischen:

- Testdaten, die vom Transaktions-Formular zur Verfügung gestellt worden sind Abb. A (3).

Sie können nicht bearbeitet oder gelöscht werden und sind im Import-Paket des Transaktions-Formulars enthalten (*Herkunft: Trans.-Formular*)

- Testdaten, die selbstständig hochgeladen worden sind (3).

Sie können bearbeitet oder gelöscht werden und werden durch den R-S Nutzer erstellt und mit Testdaten angereichert (*Herkunft: Schema*).

ⓘ Test-Vorschauen werden, wie bei der finalen Dokumenterstellung, durch die Programme im Binär-Paket erstellt. Deshalb muss, um eine Vorschau generieren zu können, das benötigte Binär-Paket im R-S vorhanden sein. Dies garantiert, dass die Testvorschau exakt der finalen Dokumenterstellung entspricht.

Um mit Testdaten zu arbeiten, können Sie das Kontextmenü  Abb. A (4) öffnen:

Sie können :

- Testdaten **erstellen**  *Abb. A (1)*. und **ändern**  (6).
- Testdaten **löschen**  (7).
- sich für Testdaten eine **Vorschau generieren** lassen  (4).

**Testdaten hochladen und aktualisieren**

Abbildung B

**Testdaten erstellen und ändern**

Um eine Testvorschau zu Erstellen , klicken Sie bitte auf *Abb. A (1)*. Nachdem Sie eine Testdatei erstellt haben wird diese Zeile grün markiert *Abb. B (3)*.

Um eine Testvorschau zu Ändern , wählen Sie bitte eine Testvorschau aus der Übersichtstabelle aus und klicken Sie anschließend auf *Abb. A (6)*. Nachdem Sie eine Testdatei geändert haben wird diese Zeile orange markiert *Abb. B (2)*.

 Sie können nur Testdaten ändern, deren Herkunft "Schema-Version" ist, die Sie also zuvor manuell im Schema erstellt haben. Testdaten, die aus der Formular-Version kommen, können nicht bearbeitet werden.

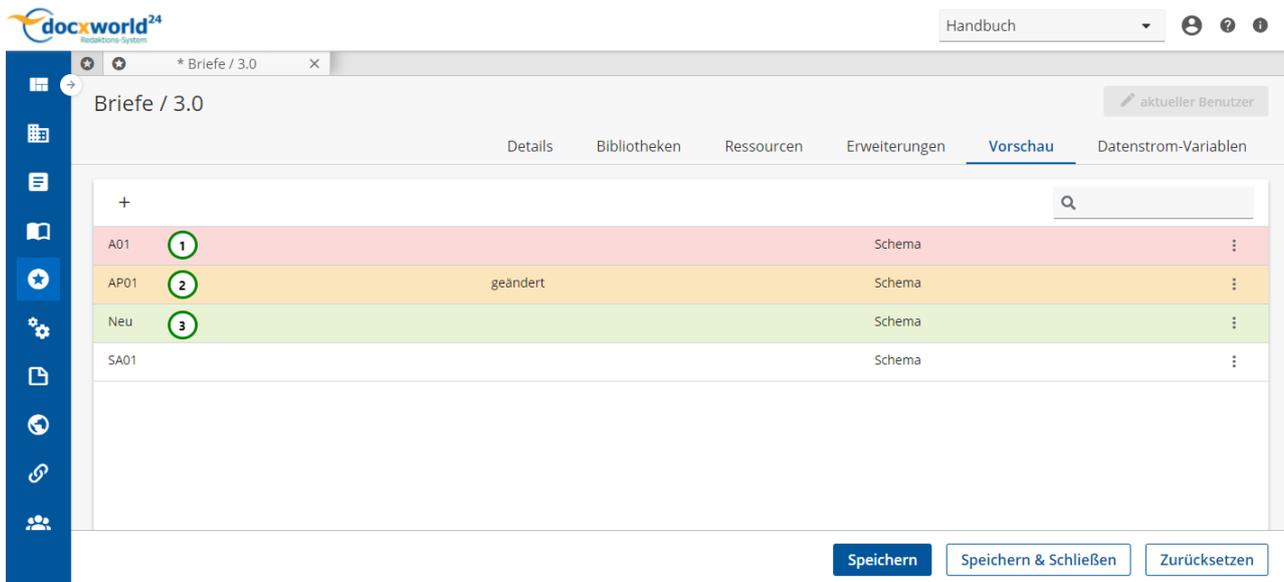


Abbildung C

### Sie können der Testdatei:

- ★ einen **Namen** geben *Abb. C (1)*.  
Beim Erstellen müssen Sie in das Feld etwas eingeben, da die Erweiterung noch keinen Namen hat. Beim Ändern muss das Feld nicht zwingend geändert werden.
- ★ eine **Beschreibung** geben *(2)*.  
Die Beschreibung ist optional. Sie wird z.B. in Auswahl-Dialogen angezeigt.
- ★ eine **Testdatei** zuordnen *(3)*.  
Die Testdatei enthält die nötigen Informationen, um eine Testvorschau aufbauen und generieren zu können.

⚠ Testdaten müssen vom Aufbau her realen Daten entsprechen. Fragen Sie hierzu Ihren Transaktions-Formular Ersteller.

### Testdaten löschen

Wählen Sie die Testvorschau aus, die Sie löschen möchten und klicken Sie anschließend auf löschen  *Abb. A (7)*.

Nachdem Sie die Daten gelöscht haben wird die Zeile rot markiert *Abb. B (1)*.

Speichern Sie die Schema-Version, um die Testdaten endgültig zu löschen oder brechen Sie die Bearbeitung der Schema-Version ab, um die Löschung zu verwerfen.

### Vorschau generieren

Um eine Vorschau anzeigen zu lassen, klicken Sie bitte auf Vorschau generieren  [Abb. A \(5\)](#) . Nun wird die Vorschau generiert. Die Ladedauer ist abhängig von der Testdatengröße. Nach erfolgreicher Erstellung sehen Sie nun folgendes Fenster:

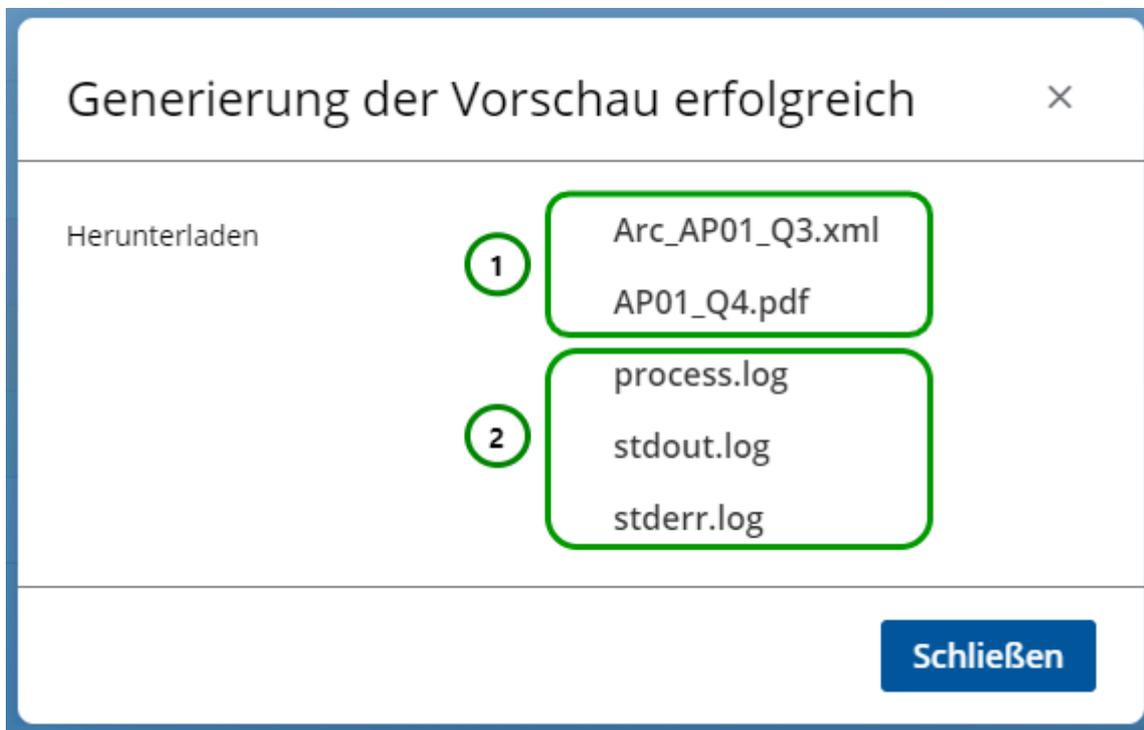


Abbildung D

Im oberen Teil [Abb. D \(1\)](#) sehen Sie die entstandenen Dokumente. Je nach Konfiguration durch den Transaktions-Formular Entwickler entstehen hier ein oder mehrere Dateien. Dateien mit dem Präfix "Arc\_" sind meist Archivkopien des Druckdokumentes. Das eigentliche Druckdokument erhält meist die Dateierdung "\*.pdf". Diese Datei können Sie nun anklicken, um sich die Vorschau anzeigen zu lassen.

Im unteren Teil [Abb. D \(2\)](#) sehen Informationsdateien, die zur Laufzeit des zur Vorschau angeforderten Dokumentes entstanden sind. Hier werden Fehler und Log-Ausgaben während der Laufzeit hineingeschrieben.

Die Testvorschau im Schema wird durch die Programme des Binär-Pakets generiert, also im gleichen Verfahren wie später beim Druck (Wenn Sie sich noch nicht mit Binär-Paketen vertraut gemacht haben, lesen Sie bitte den Abschnitt über die [Umgebung des R-S](#) und den Abschnitt über [Binär-Pakete](#)). In der Testvorschau wird deshalb exakt das Ergebnis produziert, das Sie später im Druck erwarten können. Sie können sich also darauf verlassen, dass die Dokumente, so wie Sie in der Testvorschau generiert wurden, auch gedruckt werden.

 Neu erstellte und geänderte Testdaten können erst nach dem Speichern der Schema-Version generiert werden.

 Zum Verändern der Inhalte müssen Sie vorher den Bearbeitungsmodus (  Bearbeiten) aktivieren (nur im Status Edit möglich).

## Registerkarte - Datenstrom-Variablen in Schema-Versionen

In der Registerkarte sehen Sie die vom Transaktions-Formular zur Verfügung gestellten Datenstrom-Variablen. Diese Variablen bilden eine Schnittstelle zwischen dem Datensystem (ERP-/HOST-/CRM-Systeme, etc.) und dem Transaktions-Formular des R-S. Sie enthalten formularspezifische Daten, die durch das Datensystem zur Verfügung gestellt werden. Lesen Sie dazu den [Abschnitt über Datenstrom-Variablen](#).

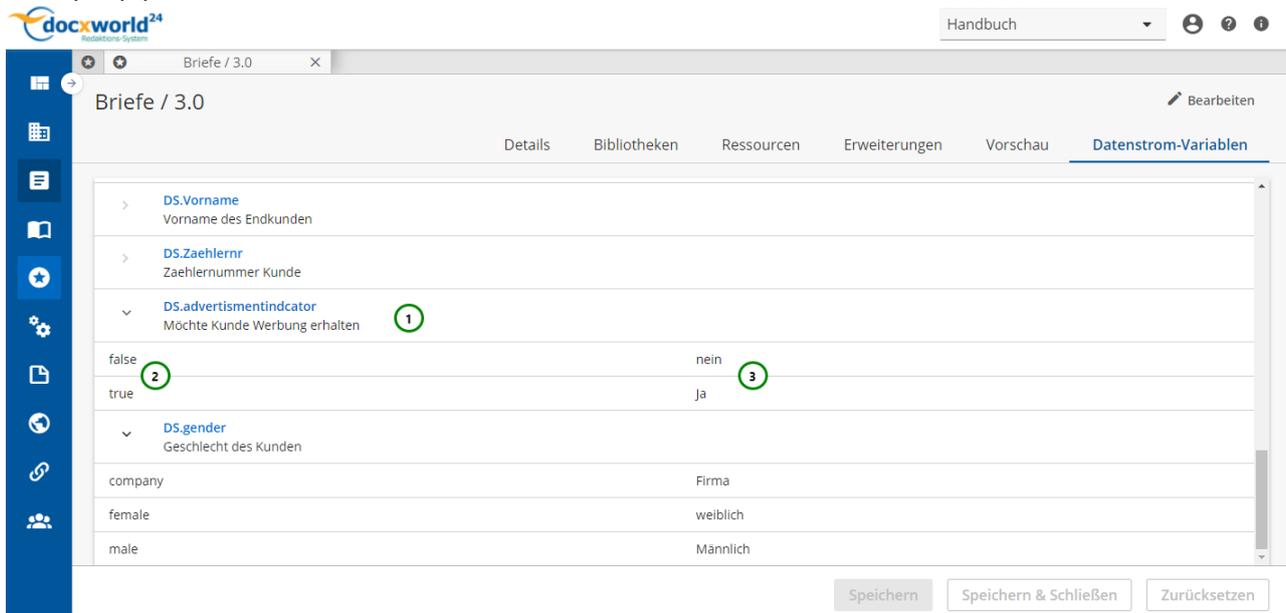


Abbildung A

### Sie sehen:

- den **Namen** der Variablen (Datenstrom-Variablen beginnen immer mit 'DS.')
- die dazugehörige **Beschreibung**
- den **Wert** der Variablen.

Wählen Sie dazu eine Datenstrom-Variable aus der Übersichtstabelle aus. Die dazugehörigen Werte werden Ihnen dann angezeigt.

⚠ Diese Werte sind nur exemplarische Werte. Sie wurden durch den Formular-Designer festgelegt und sollen anzeigen, welche Werte die Variable annehmen kann (oder wahrscheinlich annimmt).

- Eine **Beschreibungen** des möglichen Wertes.



Zum Verändern der Inhalte müssen Sie vorher den Bearbeitungsmodus ( Bearbeiten) aktivieren (nur im Status Edit möglich).

## Arbeiten mit Bibliotheken

In diesem Kapitel lernen Sie das Arbeiten mit Bibliotheken und Bibliotheks-Versionen kennen. Um zu der Übersicht des Schemas zu gelangen, wechseln Sie in den Navigationspunkt  Bibliotheken.

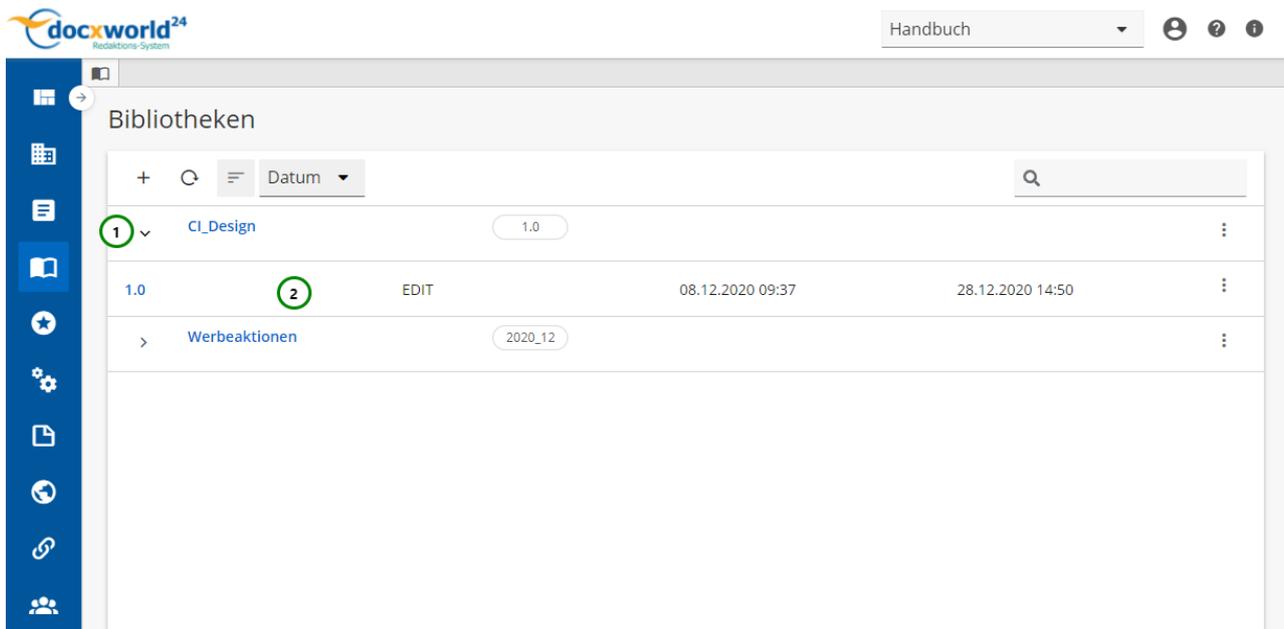


Abbildung A

Bibliotheken sind Sammlungen von Ressourcen, die Sie in ein Schema einbinden können. Sobald Sie eine Bibliothek erstellt haben *Abb. A (1)*, können Sie eine Version der Bibliothek (2) anlegen. Mit diesen Bibliotheks-Versionen können Sie dann arbeiten (siehe: *R-S Workflow und Versionierung*). Falls Sie den Abschnitt über Bibliotheken noch nicht gelesen haben, tun Sie dies bitte *hier*, bevor Sie beginnen, mit Bibliotheken zu arbeiten.

 Dieser Abschnitt behandelt nur das Erstellen und Verwalten von Bibliotheken. Um zu erfahren, wie man Bibliotheken in ein Schema einbindet, lesen Sie bitte den Abschnitt über die **Bibliotheken in Schema-Versionen**.

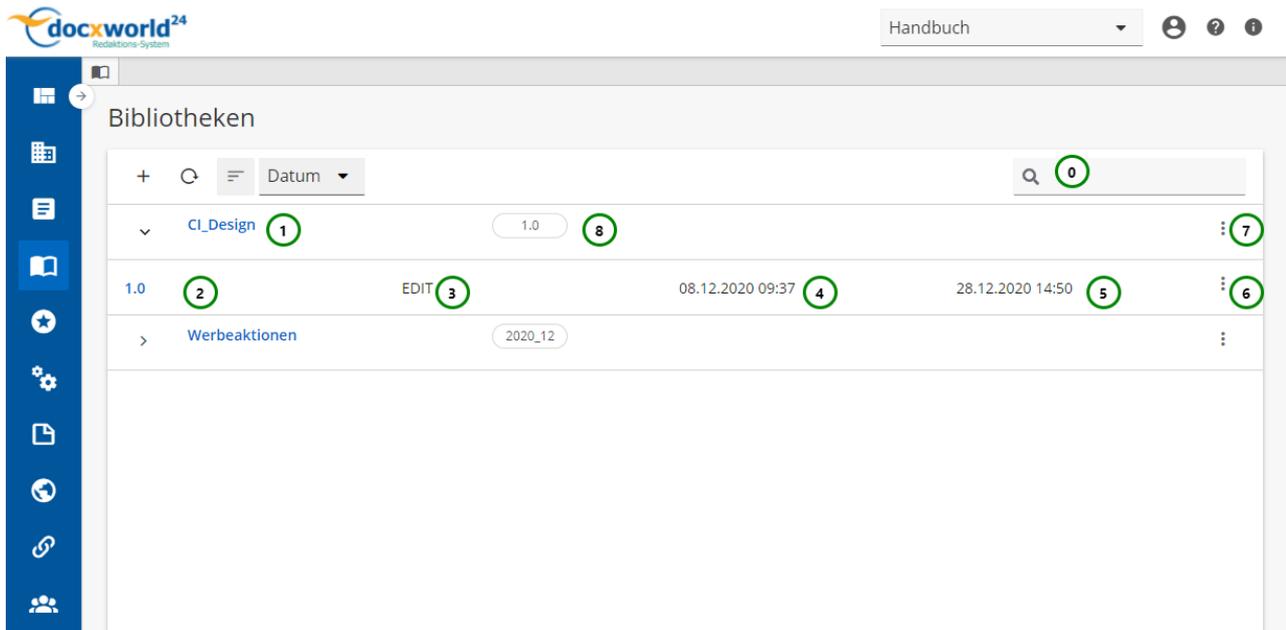


Abbildung B

Sie sehen in **Abb. B**:

- (0) Die Suche, in der Sie nach Bibliotheks-Versionen suchen können.
- (1) Name der Bibliothek inklusive Beschreibung.
- (2) Versions-Nummer der Bibliothek inklusive Beschreibung.
- (3) Den aktuellen Status der Bibliotheks-Version.
- (4) Das Erstelldatum der Bibliotheks-Version.
- (5) Das letzte Änderungsdatum der Bibliotheks-Version.
- (6) Das Kontextmenü der ausgewählten Bibliotheks-Version.
- (7) Das Kontextmenü der ausgewählten Bibliothek.
- (8) Quicklinks zu den letzten drei verwendeten Bibliotheks-Versionen der Bibliothek .

## Bibliotheken erstellen, ändern und löschen

In der Bibliotheken-Übersichtstabelle stehen Ihnen folgende Aktionen zur Verfügung, um eine Bibliothek neu zu erstellen oder eine bereits vorhandene Bibliothek zu bearbeiten:

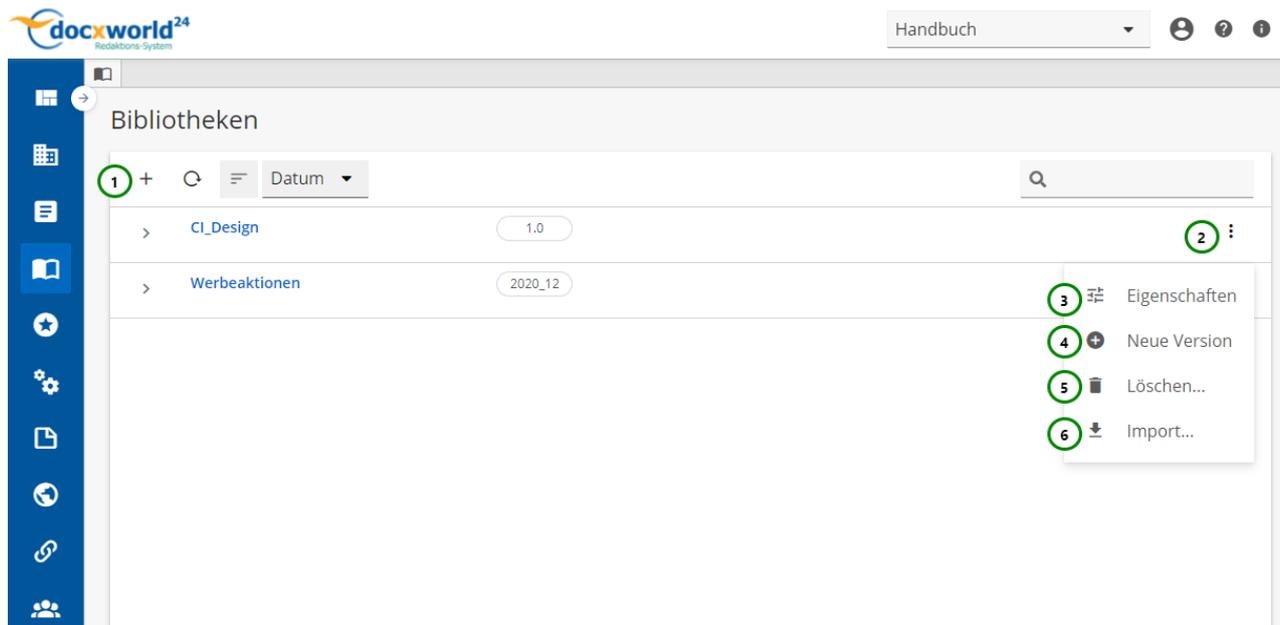


Abbildung A

Um eine Bibliothek zu erstellen, klicken Sie bitte auf **+** [Abb. A](#) (1). Im Weiteren öffnet sich ein Pop-up-Fenster, in welchem Sie den Namen und die Beschreibung erstellen können (*siehe [Abb. B](#)*).

Mit [Abb. A](#) (2) öffnen Sie das **⋮** Kontextmenü. Hier stehen Ihnen **≡** Eigenschaften (3), **+** Neue Version (4), **🗑️** Löschen (5) und **↓** Importieren (6) zur Auswahl.

Wollen Sie Änderungen vornehmen, wählen Sie das **⋮** Kontextmenü (2) der gewünschten Bibliothek in der Übersicht aus und klicken Sie anschließend auf **≡** (3). Hier öffnet sich ein Pop-up-Fenster, in welchem Sie die Bibliothek aktualisieren können (*siehe [Abb. B](#)*).



Abbildung B

Im Punkt  *Abb. A* (4) können Sie zu dieser ausgewählten Bibliothek eine neue leere Version erstellen.

Um die gewählte Bibliothek zu löschen, wählen Sie  (5) aus und um eine Bibliotheks-Version zu importieren steht Ihnen  (6) zur Auswahl .

Um mit Bibliotheken zu arbeiten benötigen Sie, wie bei Formularen und Schemas eine Bibliothek als Vorlage, von der Sie dann Versionen erstellen können, die sich in ein Schema einbinden lassen (*siehe [R-S Versionierung](#)*).



Beim Löschen des Schema müssen dessen Versionen bereits gelöscht sein. Beachten Sie dabei, dass das Objekt durch das Löschen nur inaktiv gesetzt wird (Es ist weder für den Nutzer noch für andere Objekte sichtbar oder verwendbar.) und durch einen Administrator auch wiederhergestellt werden kann. Mehr Informationen zum Löschen von Objekten finden Sie [hier](#).

## Bibliotheks-Version

Hier sehen Sie die Oberfläche, mit der Sie Bibliotheks-Versionen erstellen, anschauen, ändern, löschen, kopieren, für sie einen Status setzen, exportieren, importieren und filtern können.

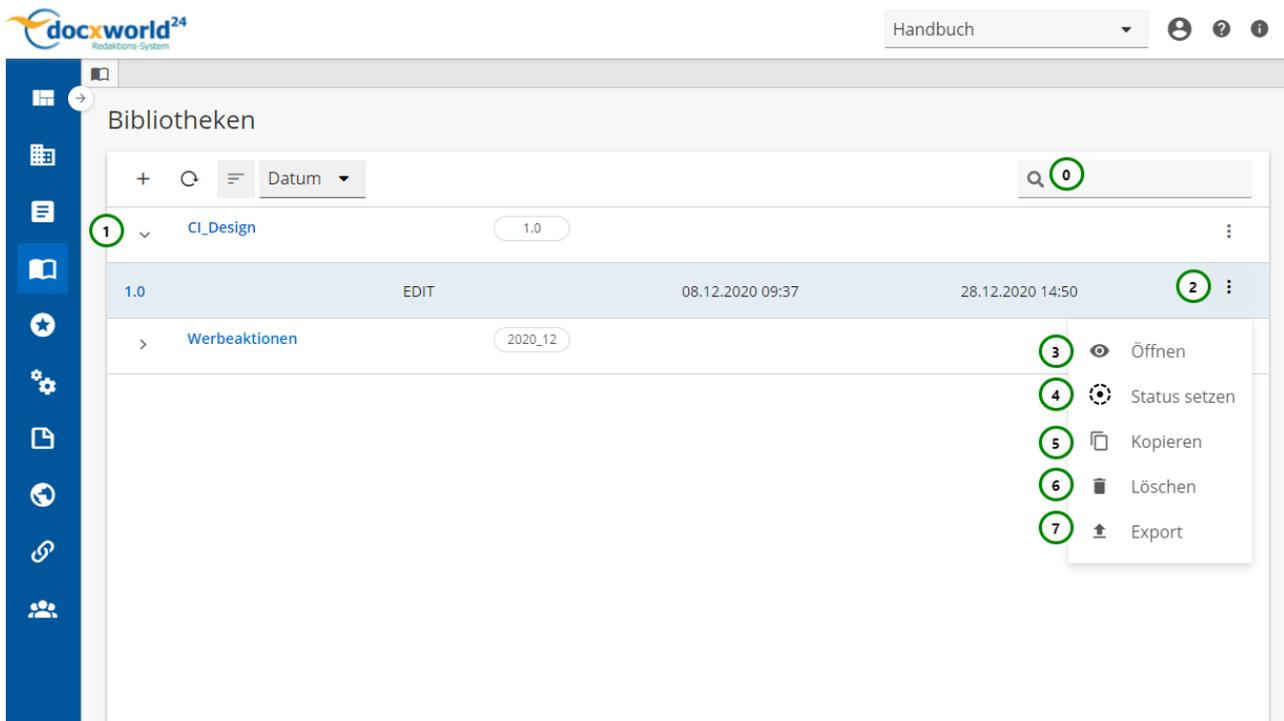


Abbildung A

### Sie sehen:

- (0) Die Suchfunktion , um die Auswahl einzuschränken.
- (1) Mit Hilfe von  können Sie Bibliotheken aufklappen bzw.  zuklappen.
- (2) Das Kontextmenü .
- (3) Öffnen , um die Bibliotheks-Version zu öffnen.
- (4) Status setzen , um den Workflow der Bibliotheks-Version zu ändern.
- (5) Kopieren , um eine Kopie der Bibliotheks-Version mit einer neuen Versionsnummer zu erstellen.
- (6) Löschen , um die Bibliotheks-Version zu löschen.
- (7) Exportieren , um die Bibliotheks-Version als Zip zu exportieren.

Durch einen Klick auf  (1) öffnen Sie die Versionsübersicht zu der ausgewählten Bibliothek. Nachdem Sie eine Bibliothek neu erstellt haben, müssen Sie eine Version für diese Bibliothek anlegen (siehe [hier](#)  *Neue Version (4)*), mit welcher Sie dann arbeiten können. Diese Bibliotheks-Version können Sie dann in ein Schema einbinden. (*Mehr Informationen über Versionierung und den R-S Workflow finden Sie [hier](#).*)

Außerdem können Sie die Schema-Versionen-Liste mit Hilfe von  (0) durchsuchen. Ausführliche Informationen zur Suchfunktion finden Sie [hier](#).

## Bibliotheks-Version erstellen, ändern und löschen

In der Bibliotheks-Versionen-Übersichtstabelle stehen Ihnen folgende Aktionen zur Verfügung, um eine Bibliotheks-Version neu zu erstellen oder eine bereits vorhandene Bibliotheks-Version zu bearbeiten:

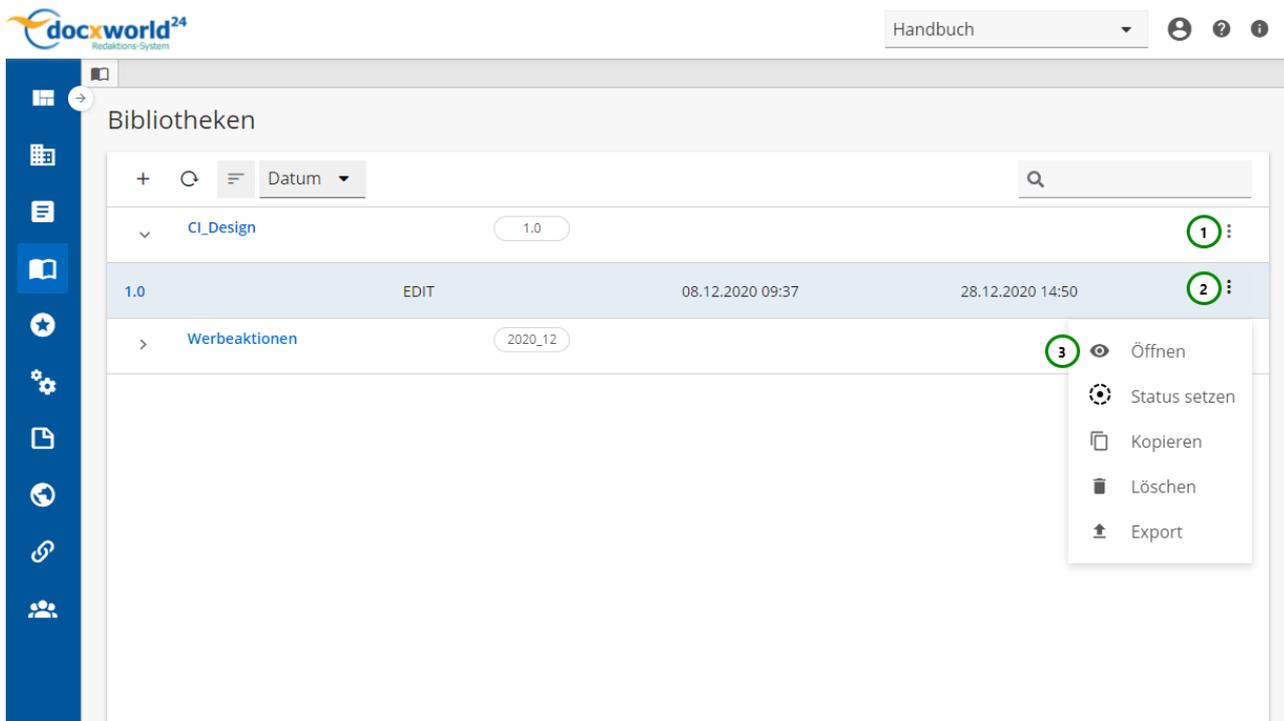


Abbildung A

Um eine Bibliotheks-Version zu erstellen, klicken Sie bitte auf  Abb. A (1) der Bibliothek und anschließend auf  Neue Version (Abbildung dazu [hier](#)).

Um eine bestehende Bibliotheks-Version zu öffnen, wählen Sie bitte  (2) der Bibliotheks-Version, die Sie öffnen möchten, aus und klicken Sie anschließend auf  Öffnen (3).

Wenn Sie eine Bibliotheks-Version neu erstellen oder eine bereits vorhandene Bibliotheks-Version öffnen und bearbeiten möchten, öffnet sich folgende Übersicht:

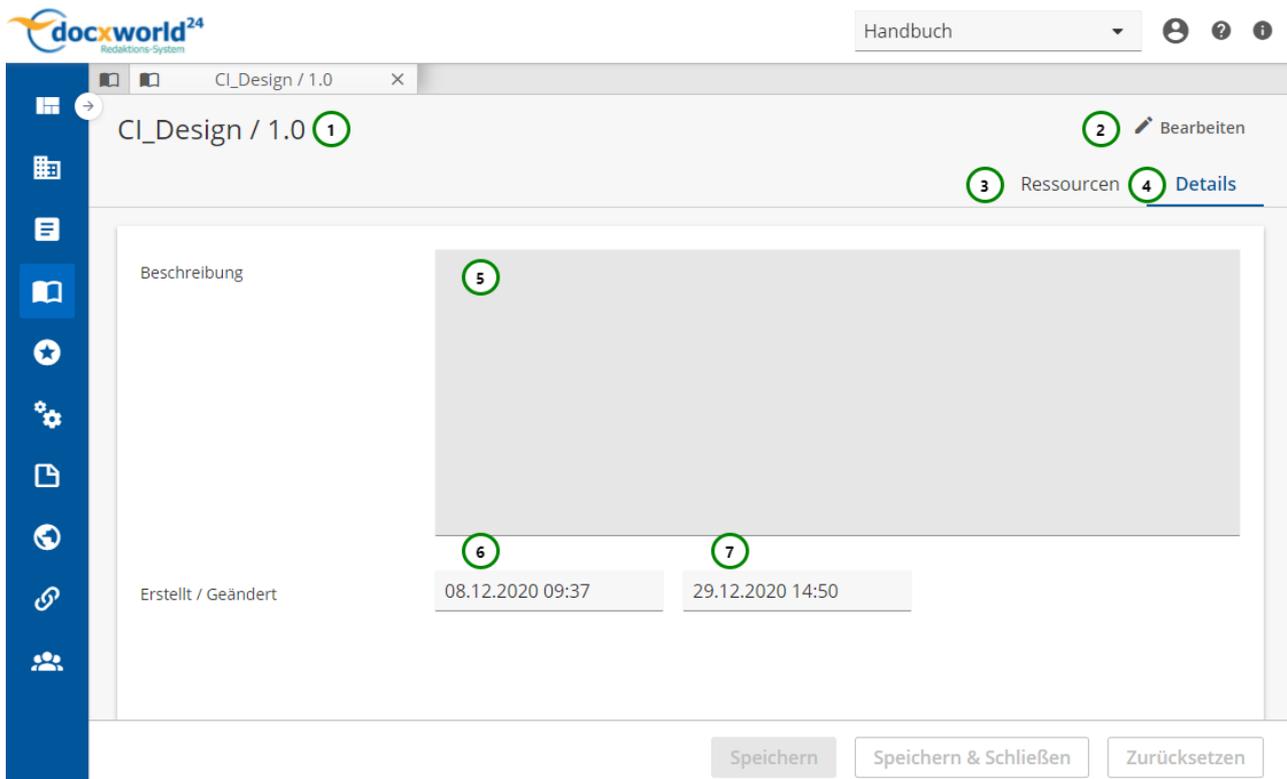


Abbildung B

⚠ Sie befinden sich immer zuerst im **View-Modus**. Die Bearbeitung kann mit  [Abb. B \(2\)](#) begonnen werden. Mehr dazu finden Sie [hier](#).

**Sie sehen in Abb. B:**

- (1) Name / Version der Bibliothek
- (2) Bearbeitungsmodus aktivieren
- (3) Registerkarte- Ressourcen
- (4) Registerkarte- Details
- (5) die Beschreibung
- (6) das Erstellungsdatum
- (7) das Änderungsdatum

## Bibliotheks-Version kopieren

In der Schema-Versionen-Übersichtstabelle steht Ihnen folgende Aktion zur Verfügung, um eine Bibliotheks-Version zu kopieren:

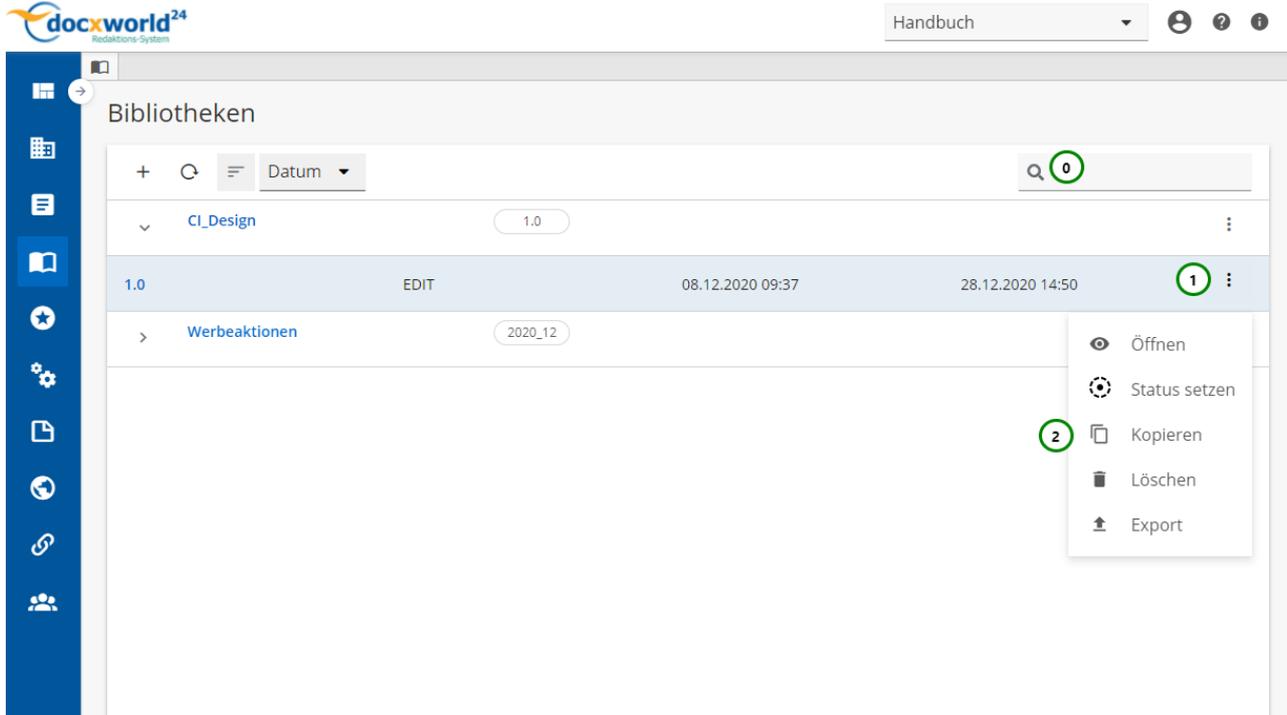


Abbildung A:

Um eine Bibliotheks-Version zu kopieren, wählen Sie bitte die Bibliotheks-Version aus, die Sie kopieren möchten und klicken Sie auf das  Kontextmenü Abb. A (1) und anschließend auf  Kopieren (2). Folgendes Fenster öffnet sich:



Abbildung B

Geben Sie für die neue Bibliotheks-Version eine Versionsnummer an Abb. B (1). Diese muss sich von der alten Version unterscheiden und darf für diese Bibliothek noch nicht existieren. Klicken Sie anschließend auf Ok (2), um den Kopiervorgang abzuschließen oder auf Abbrechen (3). Sie finden die kopierte Bibliotheks-Version nun in der Bibliotheks-Versionen-Liste:

<div style="display: flex; align-items: center;"> <span style="margin-right: 10px;">▼</span> <span style="color: blue; font-weight: bold;">CI_Design</span> <div style="margin-left: 20px;"> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 2px 10px; margin-right: 5px;">3.0_Copy</div> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 2px 10px;">1.0</div> </div> <div style="margin-left: 20px;">⋮</div> </div>										
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%;"><a href="#">3.0_Copy</a></td> <td style="width: 25%;">EDIT</td> <td style="width: 25%;">31.12.2020 07:20</td> <td style="width: 25%;">31.12.2020 07:20</td> <td style="text-align: right;">⋮</td> </tr> <tr> <td><a href="#">1.0</a></td> <td>EDIT</td> <td>08.12.2020 09:37</td> <td>29.12.2020 14:55</td> <td style="text-align: right;">⋮</td> </tr> </table>	<a href="#">3.0_Copy</a>	EDIT	31.12.2020 07:20	31.12.2020 07:20	⋮	<a href="#">1.0</a>	EDIT	08.12.2020 09:37	29.12.2020 14:55	⋮
<a href="#">3.0_Copy</a>	EDIT	31.12.2020 07:20	31.12.2020 07:20	⋮						
<a href="#">1.0</a>	EDIT	08.12.2020 09:37	29.12.2020 14:55	⋮						

Abbildung C

Wenn Sie eine Bibliotheks-Version kopieren, hat die Kopie immer den Status EDIT.

Eine kopierte Bibliotheks-Version besitzt alle Eigenschaften und Ressourcen, die das Original zum Kopierzeitpunkt besaß. Ändert sich das Original oder die Kopie, hat dies aber keine Auswirkungen auf die andere Bibliotheks-Version, bei der Kopie handelt es sich um eine neues, eigenständiges Objekt. Da Kopien den Status EDIT haben, ist es durch Kopieren möglich, mit geschlossene Bibliotheks-Versionen weiterzuarbeiten, ohne sie komplett nachbilden zu müssen.

Sie können die Schema-Versionen-Liste mit Hilfe von Abb. A (0) durchsuchen. Ausführliche Informationen zur Suchfunktion finden Sie [hier](#).

## Status einer Bibliotheks-Version setzen

In der Bibliotheks-Versionen-Übersichtstabelle steht Ihnen folgende Aktion zur Verfügung, um den Status einer Bibliotheks-Version zu ändern:

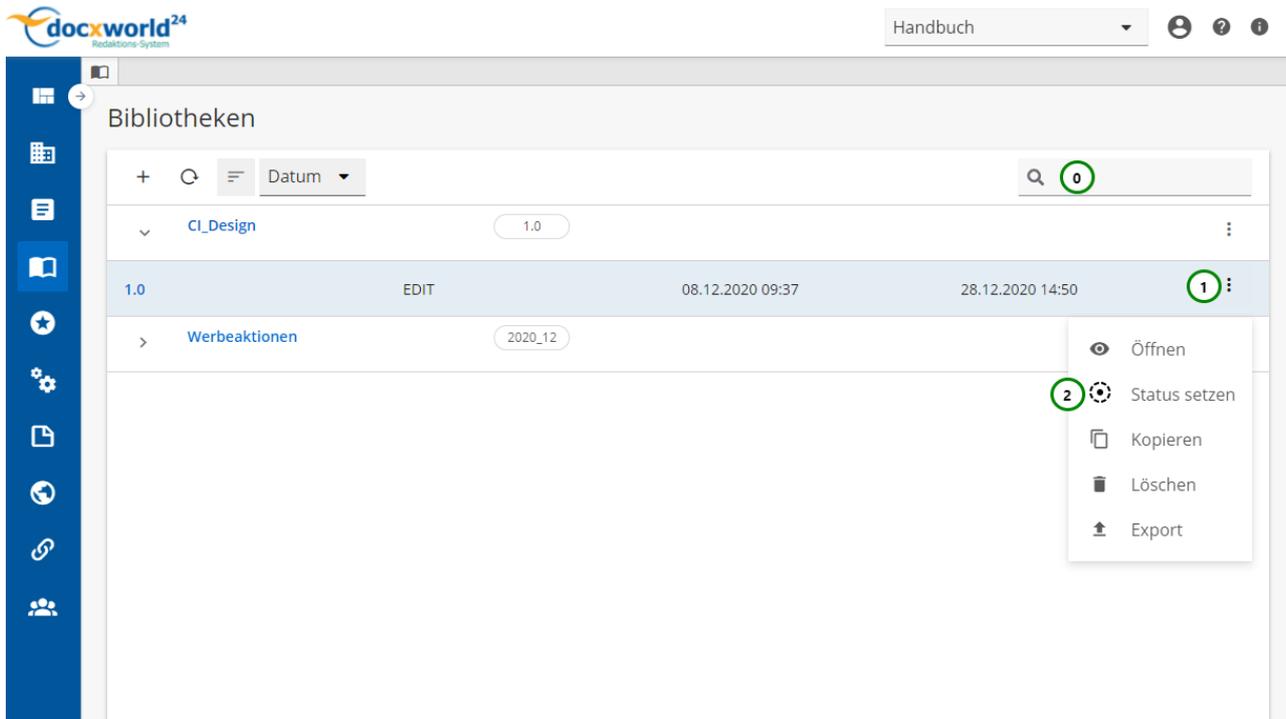


Abbildung A

Bibliotheken sind wie **Formulare**, **Schemas** oder **Produktions-Pakete** Teil des R-S Status-Workflows. Sie sind deshalb der Versionierung unterworfen und besitzen einen **Status**. Lesen Sie dazu bitte den Abschnitt über den **R-S Workflow**.

Mit ⋮ (1) öffnen Sie das Kontextmenü der ausgewählte Schema-Version. Wählen Sie hier ⚙️ Status setzen (2). Folgendes Fenster öffnet sich:

Abbildung B

Sie sehen bei Abb. B:

- den aktuellen **Status** der Bibliotheks-Version
- den neuen **Status** der Bibliotheks-Version. Wählen Sie den Status aus.



- Der Status kann nur verändert werden, wenn kein Benutzer im Bearbeitungsmodus der ausgewählten Schema-Version ist.
- Die Auswahl in Abb. B (2) wird durch den aktuellen Status bestimmt, siehe hierzu Abb. C.

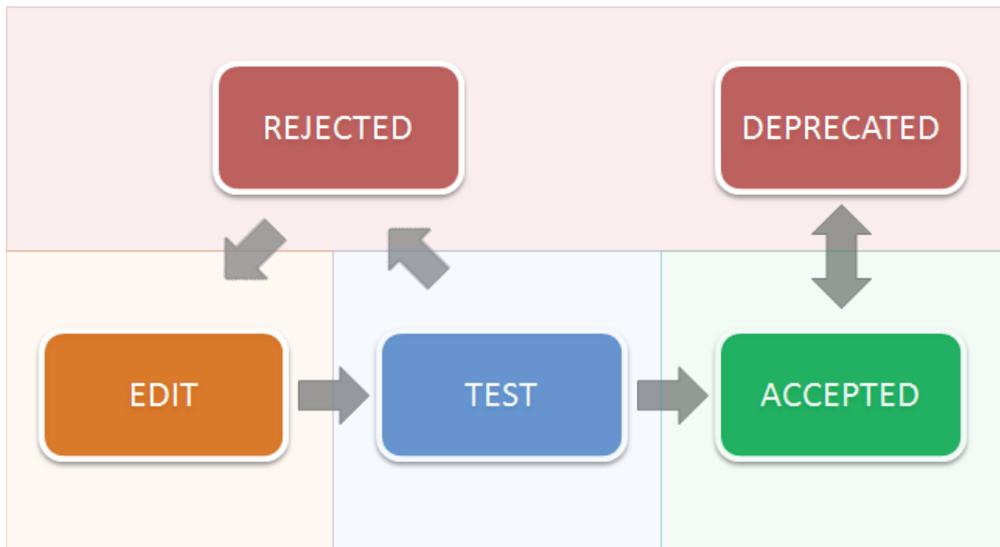


Abbildung C

Klicken Sie auf Ok [Abb. B](#), um den neuen Status zu setzen oder auf Abbrechen [Abb. B](#), um den aktuellen Status beizubehalten. Der Status der Bibliotheks-Version ist jetzt geändert. Beachten Sie bitte, dass Sie vom ACCEPTED -Status nicht wieder zurück in den EDIT -Status wechseln können. Wechseln Sie daher den Status von TEST zu ACCEPTED erst dann, wenn die Bibliotheks-Version wirklich zum Release freigegeben werden soll.

**!** Für eine genaue Beschreibung des Status-Workflows bei Bibliotheks-Versionen lesen Sie bitte unbedingt den [Abschnitt über den R-S Workflow](#).

## Bibliotheks-Version exportieren

In der Bibliotheks-Versionen-Übersichtstabelle steht Ihnen folgende Aktion zur Verfügung, mit der Sie Bibliotheks-Version exportieren können:

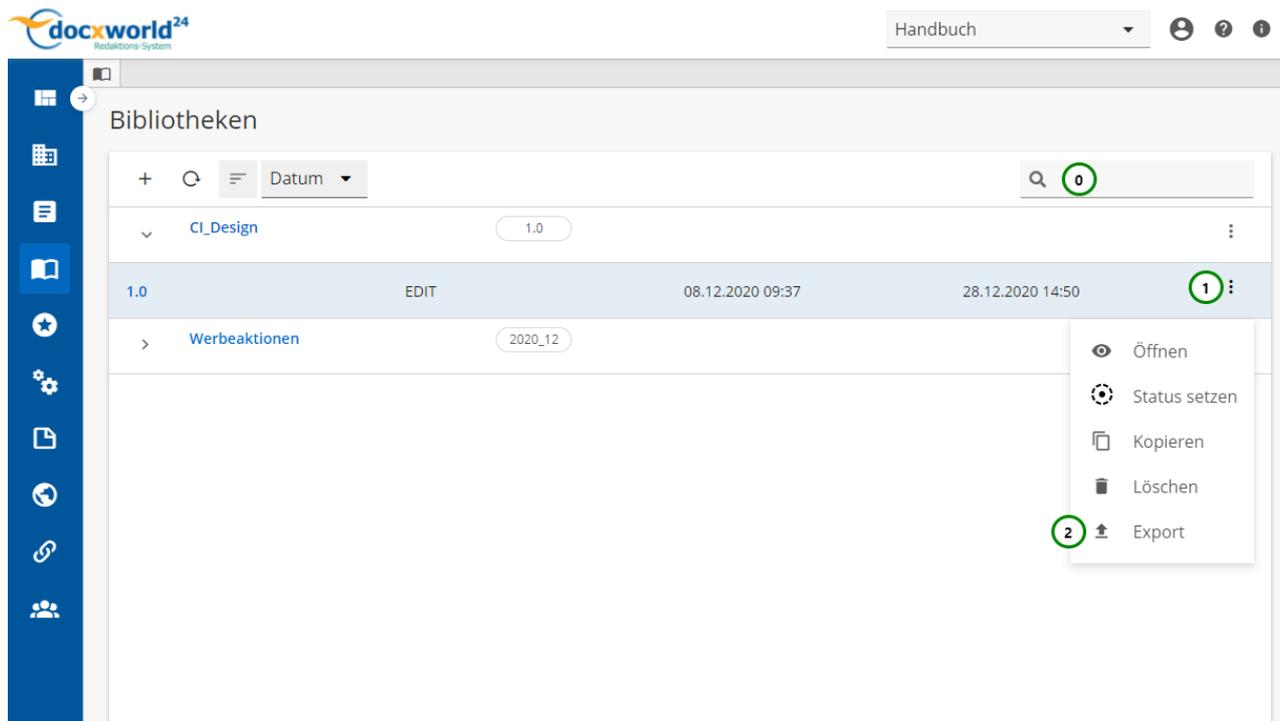


Abbildung A

Sie können eine Bibliotheks-Version exportieren, um Sie außerhalb des Arbeitsbereichs verfügbar zu machen und sie dann z.B. in einem anderen Arbeitsbereich wieder zu importieren und dort zu verwenden. Wählen Sie dazu die Bibliotheks-Version aus, welche Sie exportieren möchten. Öffnen Sie nun das Kontextmenü **Abb. A (1)** und klicken Sie anschließend zum Exportieren auf **Export (2)**. Der Export startet dann voll automatisch und die Bibliotheks-Version wird in Ihr Standard-Download-Verzeichnis exportiert (*unter Windows: User-Verzeichnis/Downloads*). Sie finden die Datei, indem Sie zum Download-Verzeichnis wechseln oder sich den Download in Ihrem Browser anzeigen lassen. Die Datei wird standardmäßig unter dem Namen "library-revision-export" abgespeichert und besitzt keine Dateiendung, liegt aber im ZIP-Format vor.

**!** Wenn Sie eine exportierte Schema-Version wieder importieren, besitzt diese immer den Status EDIT.

## Bibliotheks-Version importieren

In der Bibliotheks-Versionen-Übersichtstabelle steht Ihnen folgende Aktion zur Verfügung, mit der Sie Bibliotheks-Version importieren können:

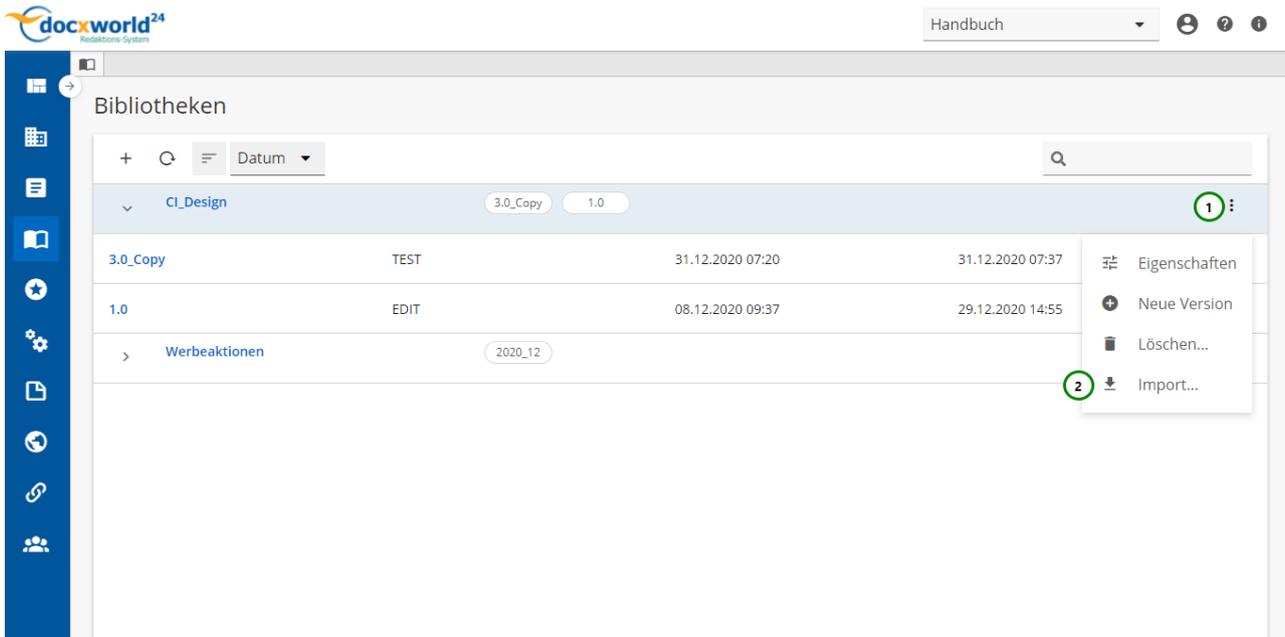


Abbildung A

Sie können eine zuvor **exportierte** Bibliotheks-Version wieder importieren. Klicken Sie dazu auf das Kontextmenü **Abb. A (1)** und wählen Sie Importieren **↓ (2)** aus.

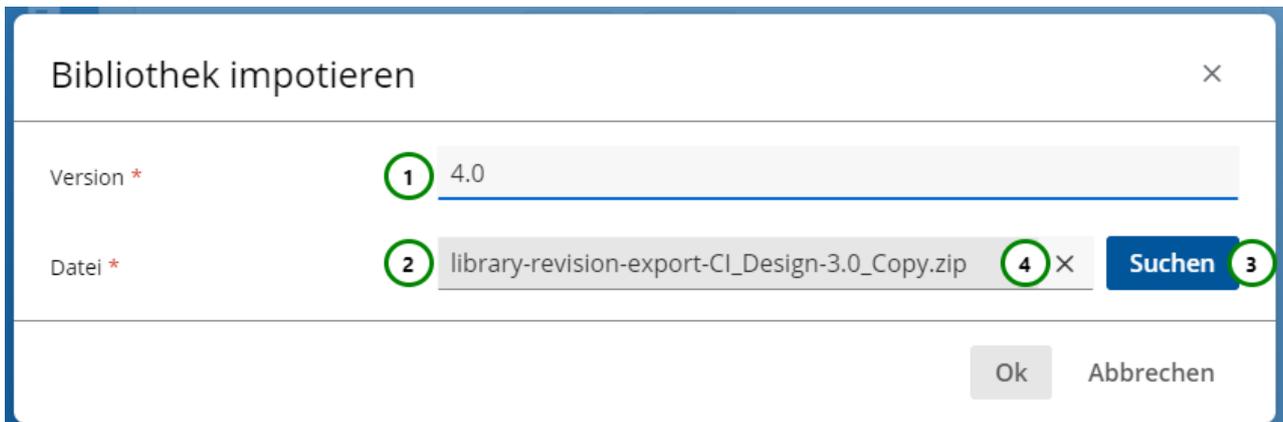


Abbildung B

Geben Sie in Abb. B (1) eine Versionsnummer für die importierte Bibliotheks-Version an. Diese darf in der Bibliothek noch nicht vorhanden sein. Klicken Sie anschließend auf (3), um die zu importierende Bibliotheks-Version auszuwählen. Nachdem Sie eine Datei ausgewählt haben, erscheint der Button  (4), mit dem Sie Ihre Auswahl wieder entfernen können. Ist die Datei bereit zum Import, so sehen Sie in (2) den Dateinamen. Ob es sich um eine gültige Bibliotheks-Versions-Datei handelt wird erst beim Importieren überprüft.

Klicken Sie zum Importieren der Datei auf Ok oder zum Abbruch des Importvorgangs auf Abbrechen. Falls die Datei ungültig ist, bricht das R-S mit einer Fehlermeldung ab. Andernfalls schließt sich der Dialog automatisch und die Schema-Version ist nun in das gewählte Schema Abb. A importiert.



Wenn Sie eine Bibliotheks-Version importieren, hat diese immer den Status EDIT.

Eine importierte Bibliotheks-Version ist, wie eine **Kopie**, unabhängig von der Original-Bibliotheks-Version. Auch wenn das Original in der selben Bibliothek vorhanden ist, können Sie mit der importierten Bibliotheks-Version arbeiten. Eine Änderung an der einen Bibliotheks-Version wirkt sich nicht auf die Andere aus.

## Bibliotheks-Ressourcen

Die vorherigen Abschnitte haben sich auf das Erstellen und Verwalten der Bibliotheks-Versionen bezogen. Hier sehen Sie jetzt, wie Sie den Inhalt einer Bibliotheks-Version, nämlich die Ressourcen, verwalten können. Dazu haben Bibliotheks-Versionen neben der Registerkarte Details noch die Registerkarte Ressourcen, welche die Ressourcenverwaltung beinhaltet:

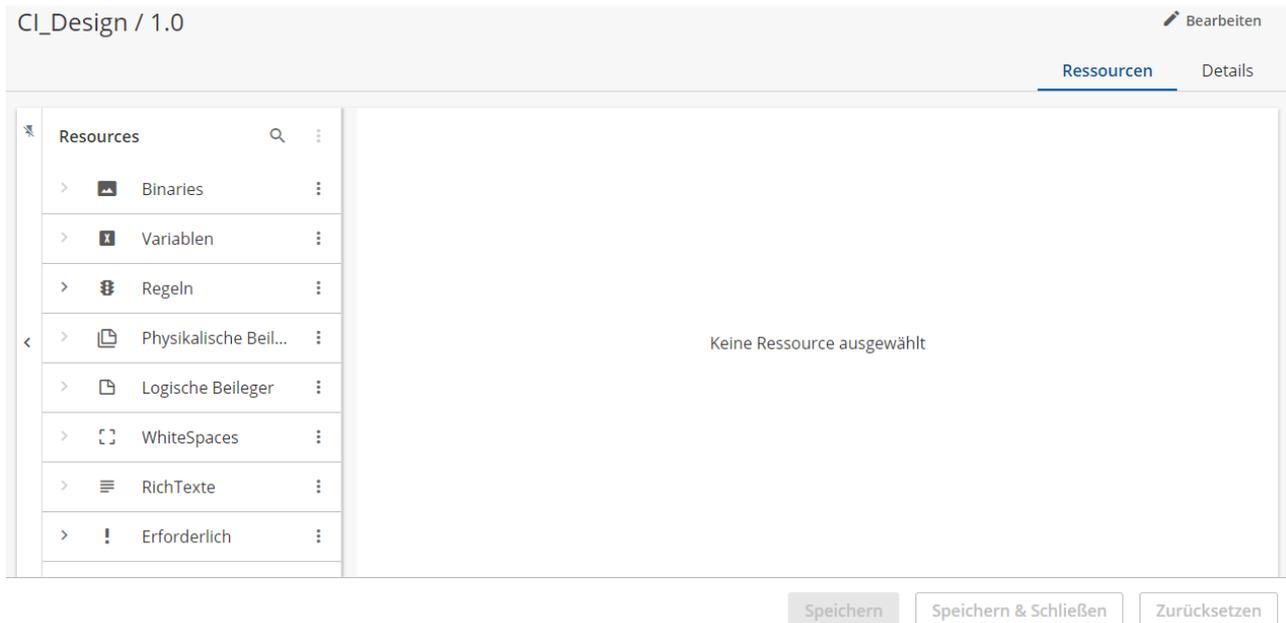


Abbildung A

In der Ressourcenverwaltung haben Sie folgende Möglichkeiten:

- Sie können Ressourcen folgenden Typs  erstellen:
  -  **Binäre Ressourcen**
  -  **Variablen**
  -  **Regeln**
  -  **Physikalische Beileger**
  -  **Logische Beileger**
  -  **WhiteSpaces**
  -  **RichTexte**
- Sie können Ressourcen vom folgenden Typ **einsehen**:
  -  **Datenstrom-Variablen**
  -  **Erforderliche Ressourcen (der eingebundenen Bibliotheken)**
- Sie können bestehende Ressourcen  **bearbeiten**,  **kopieren** und  **löschen**.
- Sie können Ressourcen  **exportieren** und  **importieren**.

Die Ressourcenverwaltung wird in einem gesonderten Kapitel behandelt, da sie in Bibliotheken und Schemas nahezu identisch ist:

[Zur Ressourcenverwaltung](#)

## Ressourcenverwaltung

In diesem Abschnitt wird die Ressourcenverwaltung in Schema- und Bibliotheks-Versionen behandelt, mit der Sie Ressourcen erstellen und bearbeiten können. Die Ressourcenverwaltung befindet sich bei Schema- und Bibliotheks-Versionen in der Registerkarte Ressourcen.

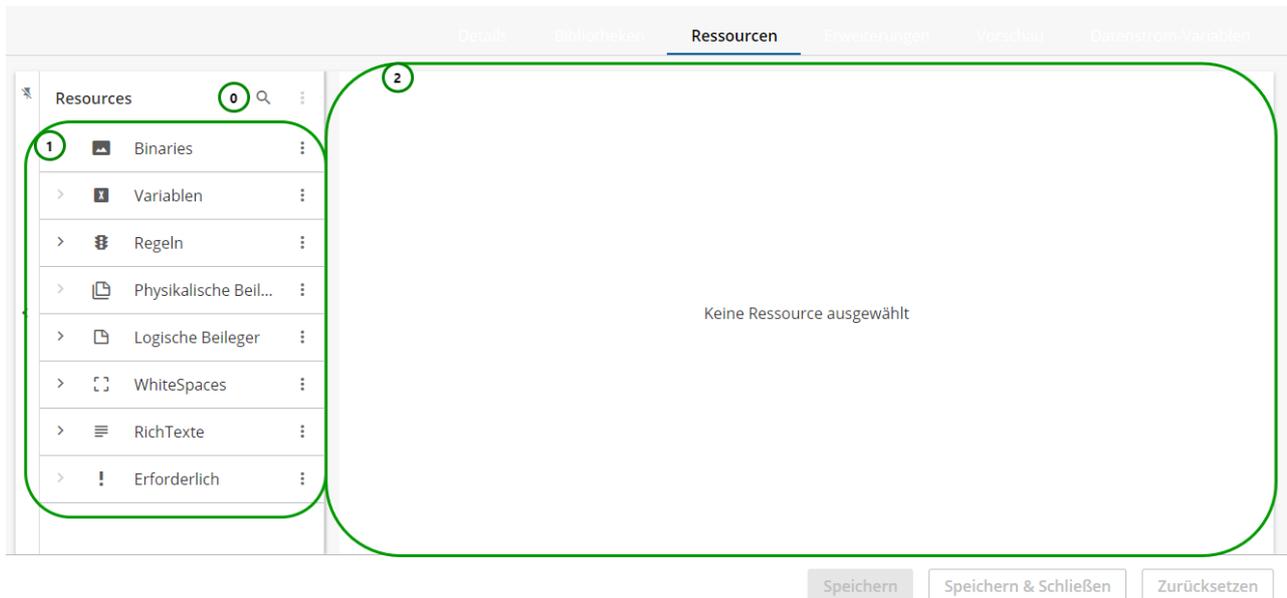


Abbildung A

### **Sie sehen:**

- Die **Ressourcen-Suche** (0)  
In der Ressourcen-Suche, können Sie nach Namen der Ressourcen suchen. Dabei werden die angezeigten Ressourcen gemäß des Suchbegriffes eingeschränkt.
- Die **Ressourcen-Typen-Übersicht** (1)  
In der Ressourcen-Typen-Übersicht sehen Sie die verfügbaren Ressourcen-Typen. Klappen Sie einen Ressourcen-Typ mit > auf, um die vorhandenen Ressourcen dieses Typs zu sehen.
- Den **Ressourcen-Inhalt** (2)  
Hier wird der Inhalt der ausgewählten Ressource dargestellt.

**i Die Ressourcenverwaltung unterscheidet sich bei Schema- und Bibliotheks-Versionen in folgenden Punkten :**

- In Schema-Versionen gibt es vom Formular bereitgestellte Datenstrom-Variablen, die es in Bibliotheks-Versionen nicht gibt.
- Erforderliche Ressourcen können nur in Bibliotheks-Versionen erstellt werden (In Schema-Versionen werden aber die erforderlichen Ressourcen der eingebundenen Bibliotheken angezeigt.).
- In Schema-Versionen sehen Sie unterhalb einer Ressource, ob diese aus einer Bibliothek stammt. Wird keine Herkunft angegeben, so ist die Ressource aus der Schema-Version. In Bibliotheks-Versionen sehen Sie diese Information nicht, da alle Ressourcen in einer Bibliotheks-Version auch aus dieser Bibliotheks-Version kommen.

Auf den folgenden Seiten lernen Sie, wie Sie mit der Ressourcenverwaltung arbeiten.

**Sie können:**

- eine Ressource  **erstellen** und **ändern** .
- eine Ressource  **löschen** .
- eine Ressource  **duplizieren** .
- eine Ressource  **exportieren** .
- eine Ressource  **importieren** .
- eine Ressource  **zurücksetzen** .

## Ressourcen erstellen und ändern

In der Ressourcenverwaltung stehen Ihnen folgende Aktionen zur Verfügung, mit der Sie Ressourcen erstellen und ändern können.

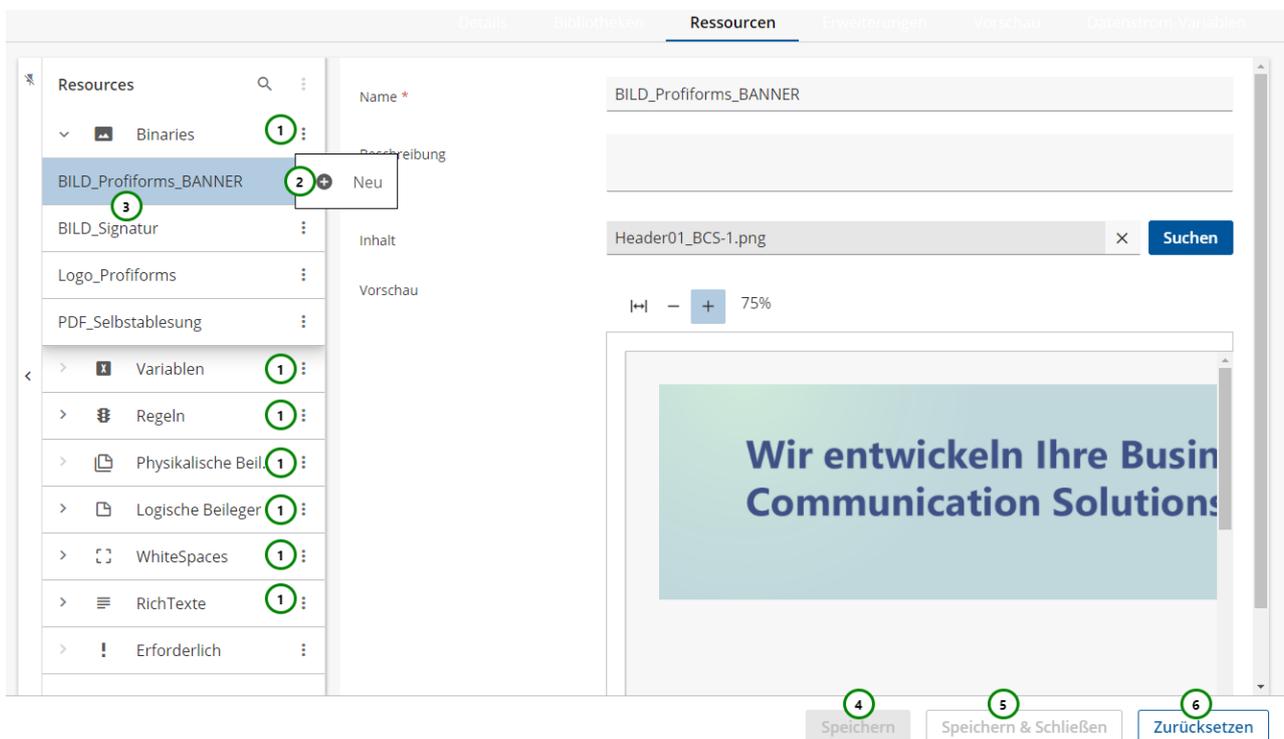


Abbildung A

Um eine Ressource zu erstellen, klicken Sie bitte auf Abb. A (1) des jeweiligen Ressourcen-Typs, den Sie anlegen möchten. Es öffnet sich das Kontextmenü mit dem Eintrag Neu (2). Nach einem Klick legen Sie eine neue Ressource des ausgewählten Typs an. Unter (3) finden Sie alle bereits angelegten Ressourcen dieses Typs.

Bei Schema- und Bibliotheks-Versionen können Sie folgende Ressourcen  erstellen und ändern:

-  *Binäre Ressourcen*
-  *Variablen*
-  *Regeln*
-  *Physikalische Beileger*
-  *Logische Beileger*
-  *WhiteSpaces*
-  *RichTexte*

Bei Bibliotheks-Versionen können Sie zusätzlich folgende Ressource  erstellen und ändern:

-  *Erforderliche Ressourcen*

Um alle Änderungen zu speichern, verwenden Sie Speichern *Abb. A* (4) oder Speichern & Schließen (5).  
Um alle Änderungen zu verwerfen, klicken Sie bitte auf Zurücksetzen (6).

 Beim Zurücksetzen werden alle Änderungen in der Schema-Version / Bibliotheksversion auf die letzte gespeicherte Version zurückgesetzt.

Sie können auch einzelne Ressourcen zurücksetzen, um so gezielt Änderungen zu verwerfen. Lesen Sie dazu [hier](#) mehr.

## Binäre Ressource erstellen und ändern

Binäre Ressourcen ermöglichen das Einbinden externer Daten, z.B. von Bildern oder PDFs. Falls Sie sich noch nicht mit binären Ressourcen vertraut gemacht haben, eine Beschreibung finden Sie [hier](#).

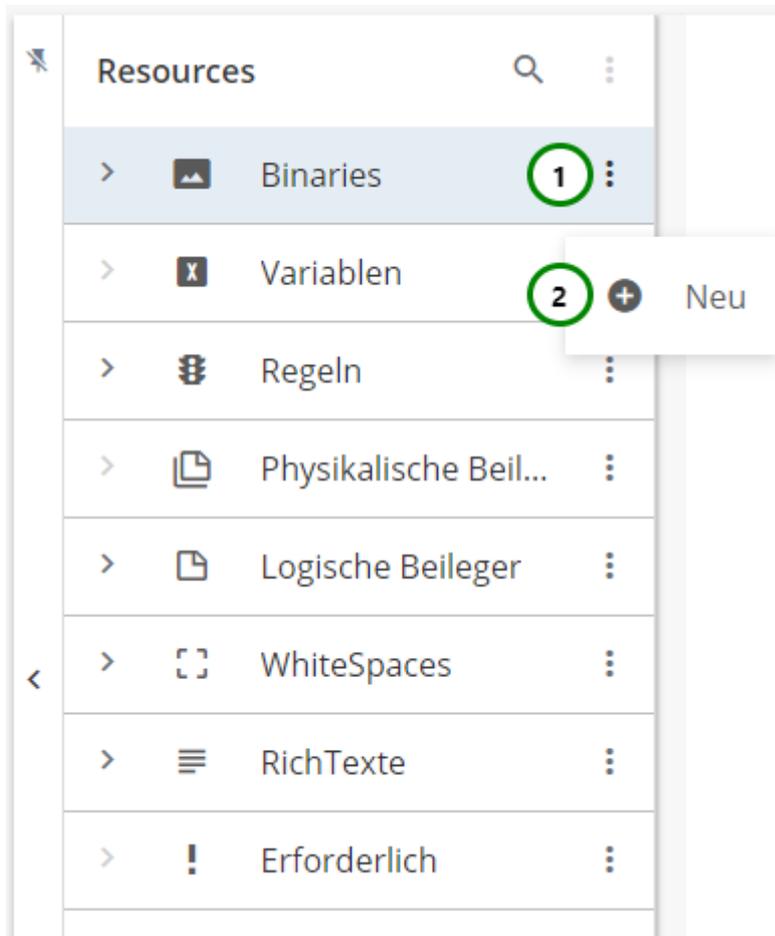


Abbildung A

In dem Ressourcen-Typ  Binaries finden Sie alle bereits angelegt binären Ressourcen. Klappen Sie dazu den Ressourcen-Typ auf .

Um eine neue binäre Ressource zu erstellen, klicken Sie bitte auf das  Kontextmenü [Abb. A\(1\)](#) und wählen Sie nun  Neu (2) aus.

Um eine Änderung vorzunehmen, wählen Sie die Ressource, die Sie ändern möchten, aus der Übersichtsliste aus [Abb. B\(1\)](#). Nun sehen Sie in dem Inhaltsbereich die Informationen zu dieser Ressource:

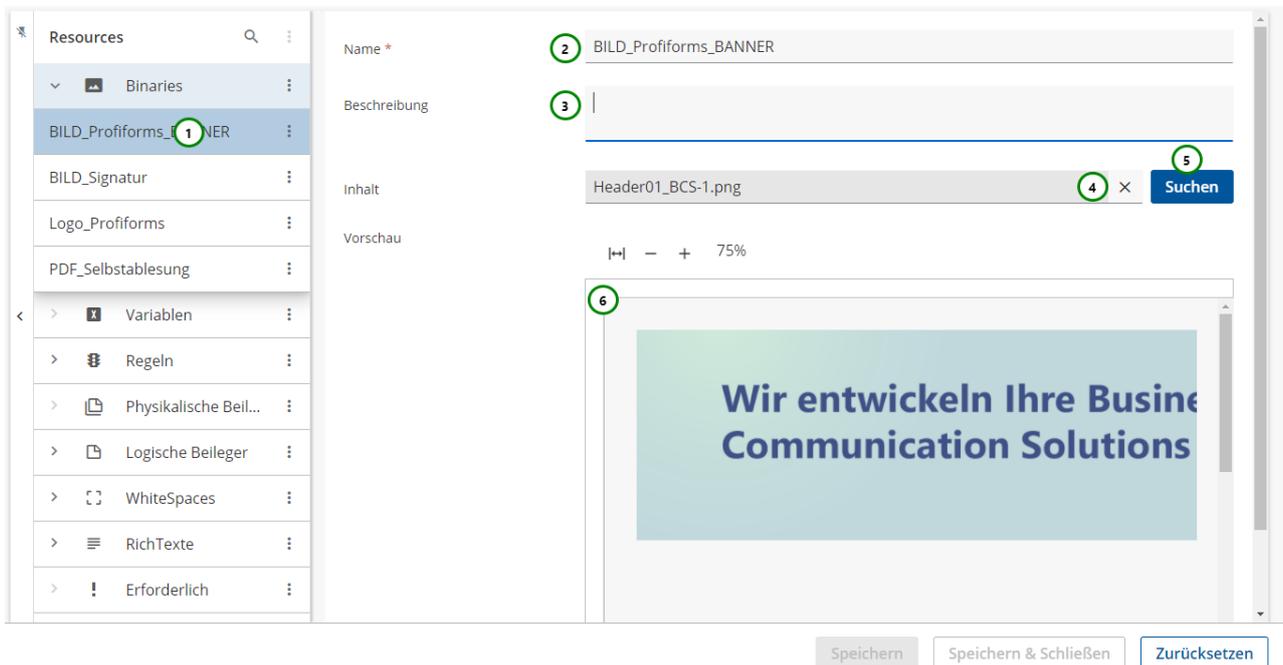


Abbildung B

- ⚠ Wenn Sie eine binäre Ressource neu erstellen, sehen Sie folgende Objekte nicht:
- **Entfernen-Button** *Abb. B* (4)
  - **Vorschau** (6)

Der **X** Entfernen-Button ist nur verfügbar, wenn bereits eine binäre Datei hochgeladen wurde. Die Vorschau wird erst verfügbar, nachdem Sie die Ressource und die Schema-Version gespeichert haben.

**Sie können der binären Ressource:**

- ★ einen **Namen** geben *Abb. B* (2).  
Sie müssen der binären Ressource einen Namen geben, unter dem Sie die Ressource dann ansprechen und nutzen.  
**Beachten Sie:** In einer Schema-Version/Bibliotheks-Version dürfen keine 2 Ressourcen denselben Namen haben. Hat eine Ressource aus einer Schema-Version denselben Namen wie eine Ressource aus einer eingebundenen Bibliotheks-Version, überschreibt die Schema-Ressource die Bibliotheks-Ressource.
- ★ eine **Beschreibung** geben (3).

**Um der binären Ressource einen Inhalt zuzuordnen, müssen Sie:**

★ Eine **Datei** in die binäre Ressource laden .

Klicken Sie dazu auf Abb. B (5). Ein Dialog öffnet sich, in dem Sie eine Datei auswählen können. Nachdem Sie eine Datei ausgewählt haben, wird diese in die Ressource geladen. Hat die Datei ein unerlaubtes Dateiformat, wird der Prozess abgebrochen und es erscheint eine Fehlermeldung. Um die aktuelle Ressource zu entfernen, klicken Sie auf (4) .

❗ Welche Dateiformate in einer binären Ressource verwendet werden dürfen, ist durch den Administrator in der rs.server.properties Datei festgelegt.



Beachten Sie, dass die Umbenennung einer Ressource alle Verweise auf die Ressource ungültig macht, insbesondere in Erweiterungen. Wird die Ressource in einer Erweiterung oder in einer anderen Ressource verwendet, müssen Sie nach dem Umbenennen alle Verweise anpassen.

## Variable erstellen und ändern

Variablen dienen zur Speicherung von Werten, z.B. von Texten oder numerischen Werten. Falls Sie sich noch nicht mit Variablen vertraut gemacht haben, eine Beschreibung finden Sie [hier](#).

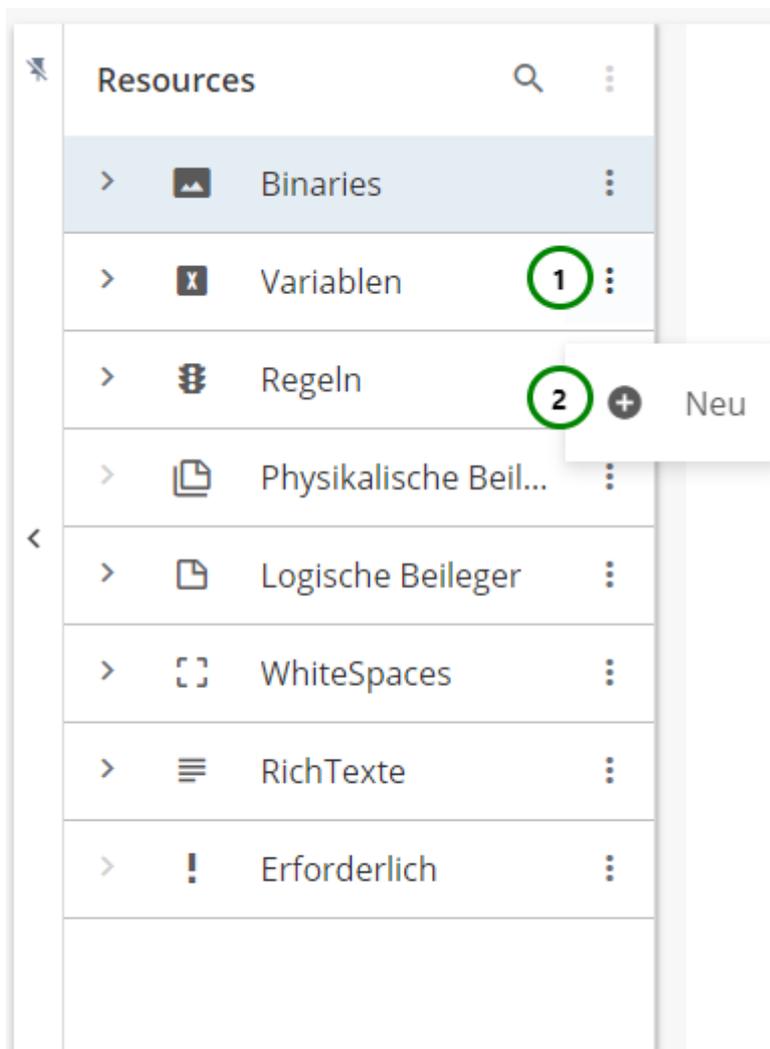


Abbildung A

In dem Ressourcen-Typ **X** Variablen finden Sie alle bereits angelegt Variablen. Klappen Sie dazu den Ressourcen-Typ auf **>**.

Um eine neue Variable zu erstellen, klicken Sie bitte auf das **⋮** Kontextmenü [Abb. A \(1\)](#)

des Ressourcen-Typs und wählen Sie nun **+** Neu (2) aus.

Um Änderungen vorzunehmen, wählen Sie die Ressource, die Sie ändern möchten, aus der Übersichtsliste aus [Abb. B \(1\)](#). Nun sehen Sie in dem Inhaltsbereich die Informationen zu dieser Ressource:

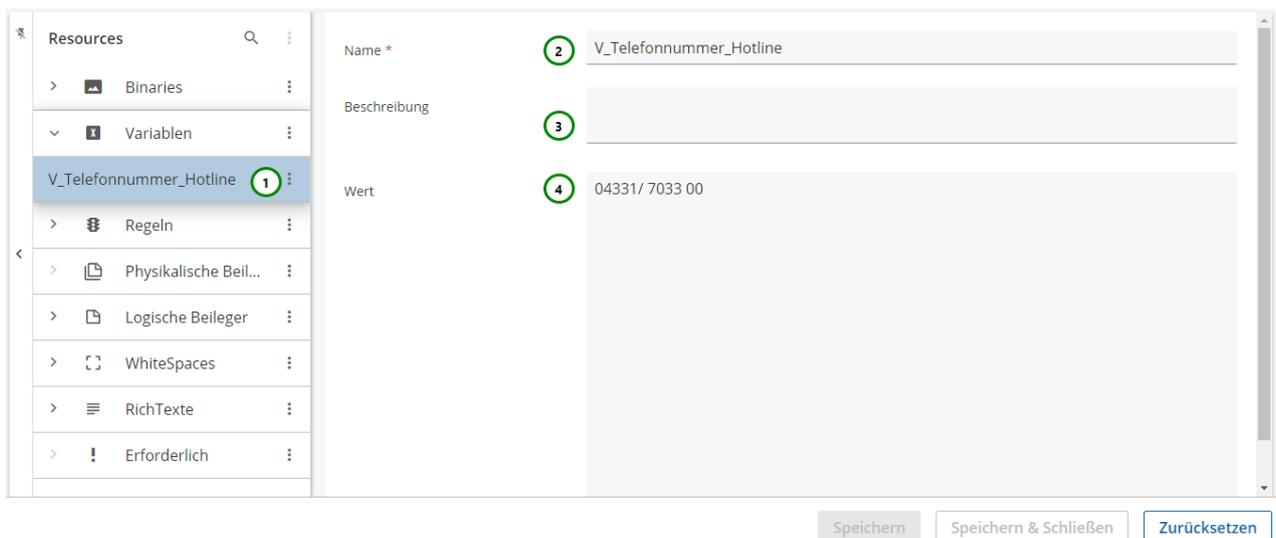


Abbildung B

### Sie können der Variable:

- ★ einen **Namen** geben *Abb. B (2)*.  
Sie müssen der Variable einen Namen geben, unter dem Sie die Variable dann ansprechen und nutzen.  
**Beachten Sie:** In einer Schema-Version/Bibliotheks-Version dürfen keine 2 Ressourcen denselben Namen haben. Hat eine Ressource aus einer Schema-Version denselben Namen wie eine Ressource aus einer eingebundenen Bibliotheks-Version, überschreibt die Schema-Ressource die Bibliotheks-Ressource.
- ★ eine **Beschreibung** geben (3).  
Die Beschreibung ist optional.

### Um der Variable einen Inhalt zuzuordnen, können Sie:

- ★ den **Wert** der Variable angeben *Abb. B (4)*.  
Der Wert einer Variable kann ein beliebiger **STRING / Datentyp** sein. Auch erlaubt sind **EVAL-Anweisungen** (inkl. Funktionen), **Inline-** und **Escape** -Sequenzen. Beachten Sie dazu bitte die **Definition von Variablen**.

Sie können hier die **Eingabe-Hilfe** verwenden

- ⚠ Beachten Sie, dass die Umbenennung einer Ressource alle Verweise auf die Ressource ungültig macht, insbesondere in Erweiterungen. Wird die Ressource in einer Erweiterung oder in einer anderen Ressource verwendet, müssen Sie nach dem Umbenennen alle Verweise anpassen.

## Regel erstellen und ändern

⚠️ Bevor Sie mit Regeln arbeiten, lesen Sie bitte den [Abschnitt über Regeln](#).

Regeln bieten eine Möglichkeit, abhängig von Bedingungen unterschiedliche Werte zurückzugeben.

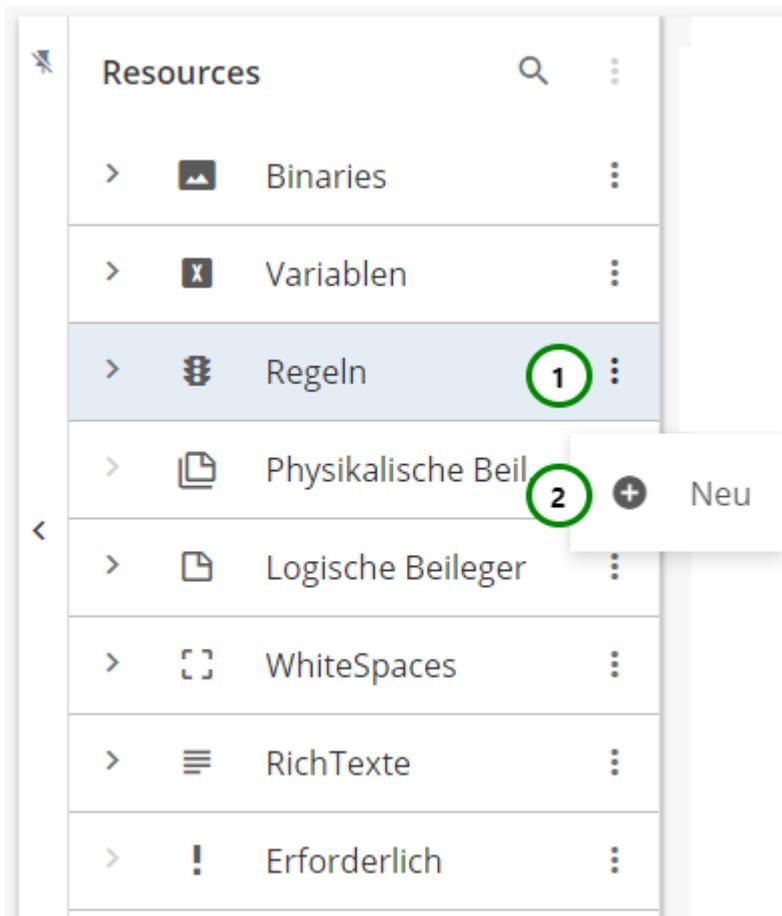


Abbildung A

In dem Ressourcen-Typ  Regeln finden Sie alle bereits angelegten Regeln. Klappen Sie dazu den Ressourcen-Typ auf  .

Um eine neue Regel zu erstellen, klicken Sie bitte auf das  Kontextmenü [Abb. A \(1\)](#) des Ressourcen-Typs und wählen Sie nun  Neu (2) aus. Sie landen zuerst in den Eigenschaften der Regel. Weitere Informationen finden Sie [hier](#).  
Um Änderungen vorzunehmen, wählen Sie die Ressource, die Sie ändern möchten, aus der Übersichtsliste aus [Abb. B \(1\)](#) . Nun sehen Sie in dem Inhaltsbereich die Informationen zu dieser Ressource:

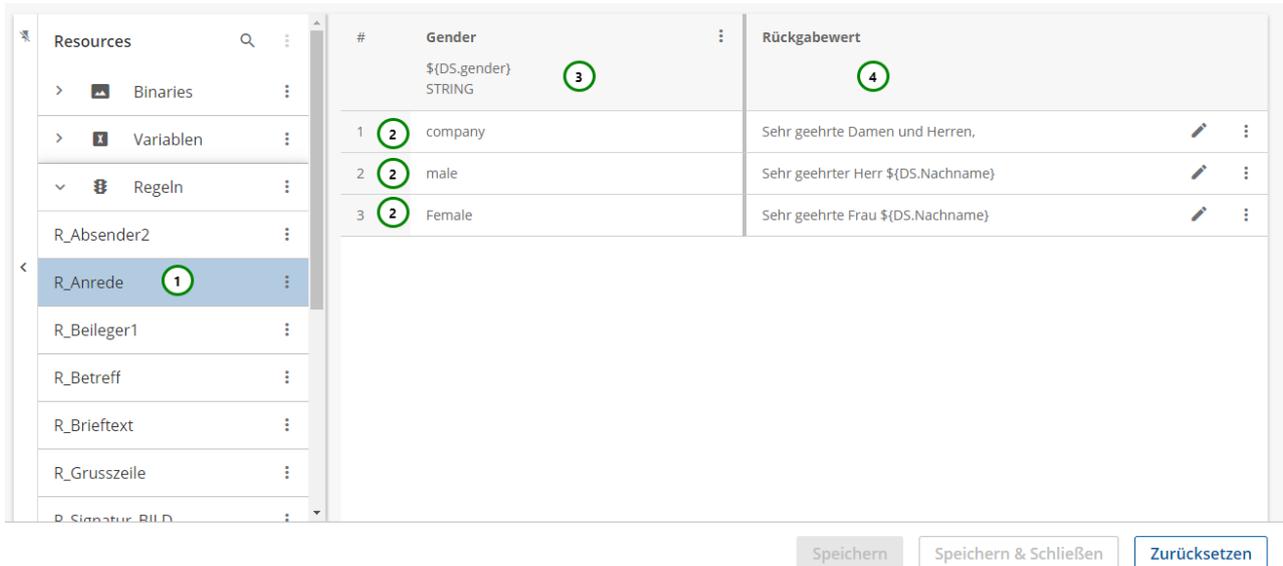


Abbildung B

**Sie sehen:**

- die vorhandenen **Zeilen** *Abb. B (2)*.
- die vorhandenen **Spalten** *(3)*.
- eine feste Spalte für die **Rückgabewerte** *(4)*.

**Um der Regeln einen Inhalt zuzuordnen, können Sie:**

- ★ eine **Spalte hinzufügen / bearbeiten / löschen** .  
Spalten stellen die Bedingungen einer Regel. Wie man mit Spalten arbeitet, erfahren Sie [hier](#).
- ★ eine **Zeile hinzufügen / bearbeiten / löschen**.  
Zeilen stellen die alternativen Auswahlmöglichkeiten einer Regel. Wie man mit Zeilen arbeitet, erfahren Sie [hier](#).
- ★ eine **Zeile auf- oder abwärts verschieben**.  
Die Hierarchie der Zeilen in einer Regel wird durch deren Position bestimmt. Um die Hierarchie zu ändern, können Sie die Zeilen in der Liste verschieben. Wie Sie Zeilen verschieben, erfahren Sie [hier](#).

**Eigenschaften der Regel**

Regeln werden als komplexe Elemente betrachtet und haben daher zusätzlich eine Eigenschaften-Seite. Diese erreichen Sie über das Kontextmenü  *Abb. C.(1)* der ausgewählten Ressource nach einem Klick auf  Eigenschaften *(2)* .

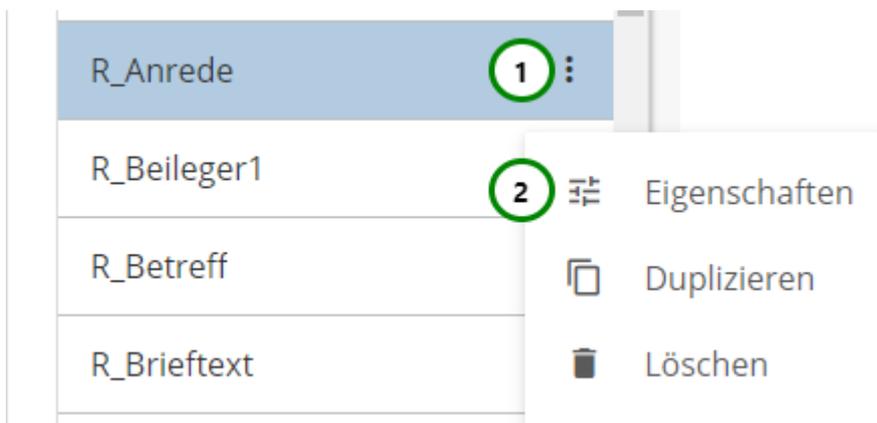


Abbildung C

Es öffnet sich folgender Inhaltsteil:

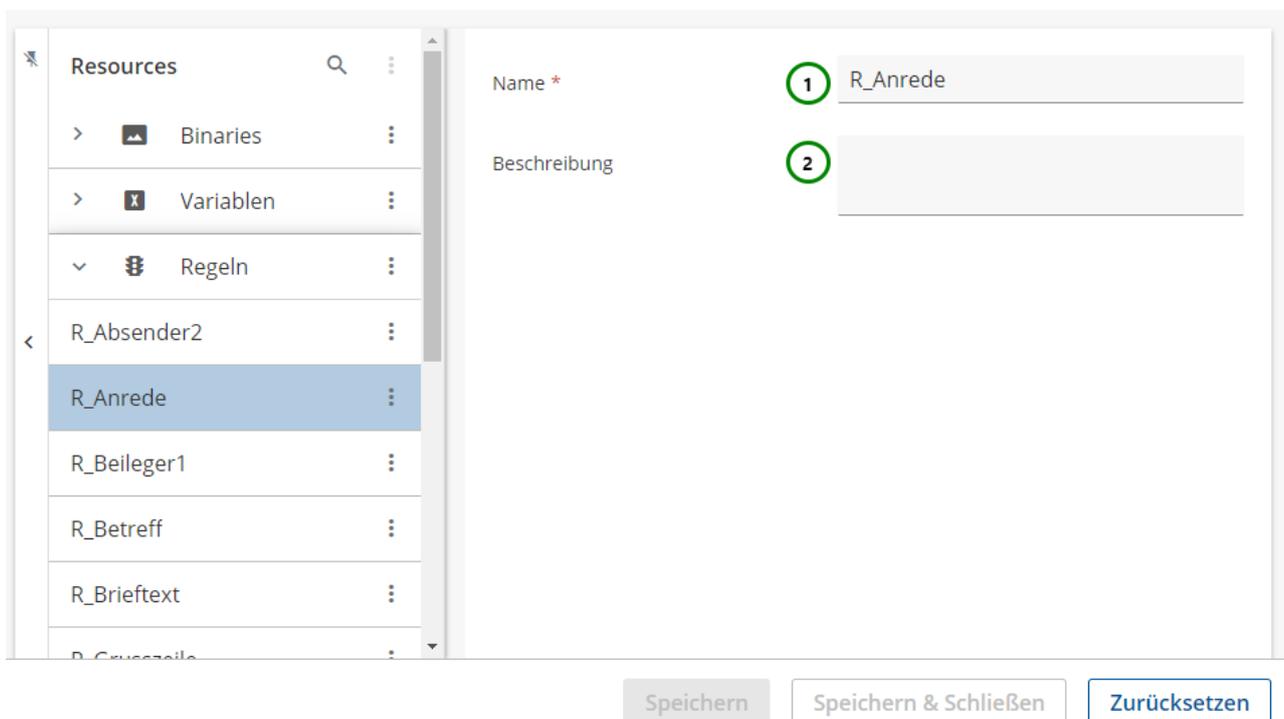


Abbildung D

**Sie können in den Eigenschaften der Regel:**

- ★ einen **Namen** vergeben *Abb. D (1)*.

Sie müssen der Regel einen Namen geben, unter dem Sie die Regel dann ansprechen und nutzen.

**Beachten Sie:** In einer Schema-Version/Bibliotheks-Version dürfen keine 2 Ressourcen denselben Namen haben. Hat eine Ressource aus einer Schema-Version denselben Namen wie eine Ressource aus einer eingebundenen Bibliotheks-Version, überschreibt die Schema-Ressource die Bibliotheks-Ressource.

- ★ eine **Beschreibung** hinzufügen (2).

Die Beschreibung ist optional. Sie wird z.B. in Auswahl-Fenstern angezeigt.

⚠ Die Bearbeitung des Inhaltes erreichen Sie über einen Klick der gewünschten Ressource in der Übersichtstabelle.

⚠ Beachten Sie, dass die Umbenennung einer Ressource alle Verweise auf die Ressource ungültig macht, insbesondere in Erweiterungen. Wird die Ressource in einer Erweiterung oder in einer anderen Ressource verwendet, müssen Sie nach dem Umbenennen alle Verweise anpassen.

## Spalte einfügen, bearbeiten und löschen

Spalten bilden die Bedingungen einer Regel.

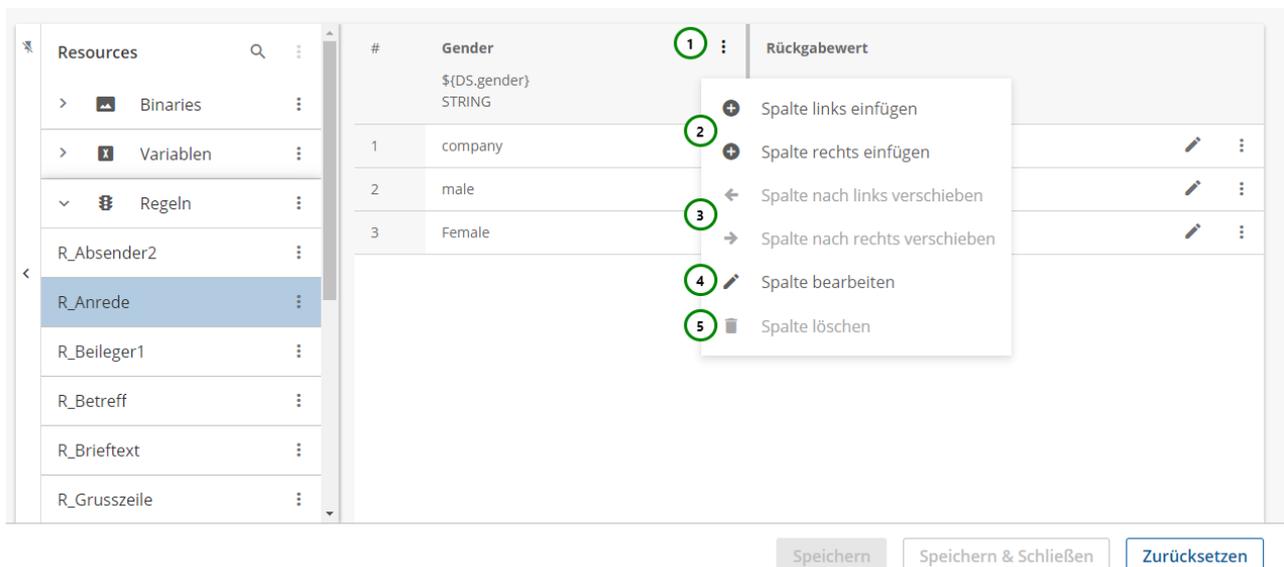


Abbildung A

Um eine Spalte hinzuzufügen, klicken Sie bitte auf das  Kontextmenü [Abb. A \(1\)](#) einer Spalte und wählen Sie anschließend  Spalte links/rechts einfügen (2). Es öffnet sich ein neues Fenster:

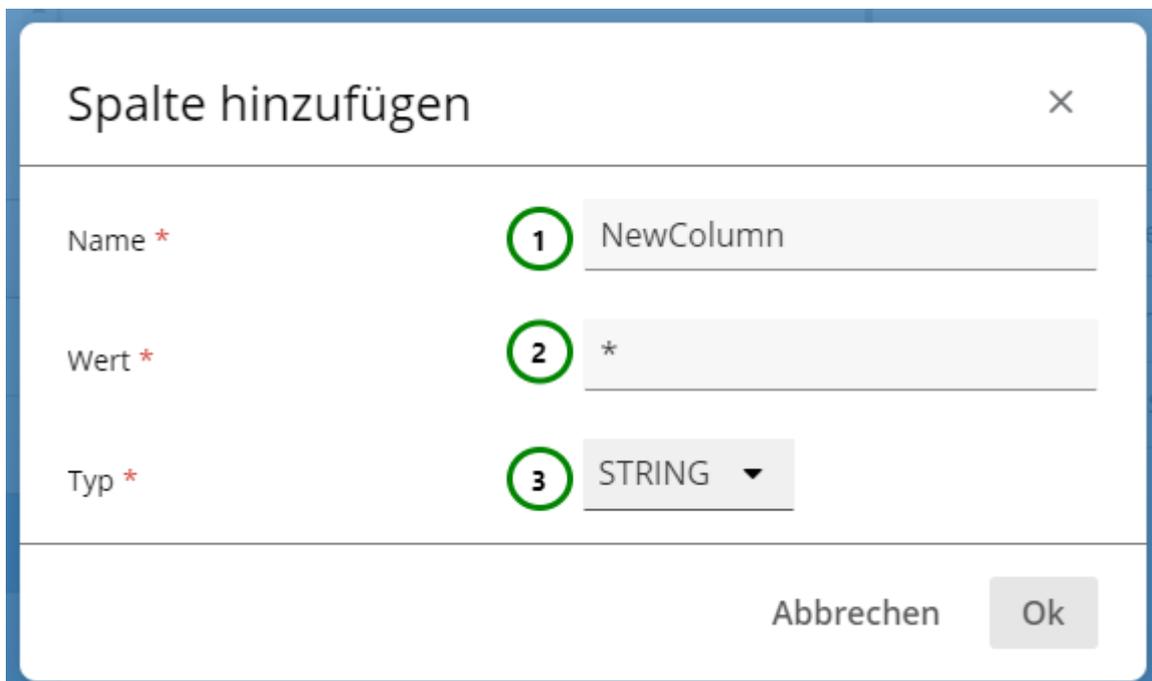


Abbildung B

**Geben Sie der Spalte:**

★ einen **Namen** *Abb. B (1)*.

★ einen **Wert** (2).

Der Wert muss vom gewählten Datentyp sein. Der Wert einer Spalte ist die Bedingung dieser Spalte. Erlaubt sind statische Werte oder EVAL-Anweisungen.

★ einen **Typ** (3).

Klicken Sie auf , um die verfügbaren **Datentypen** einzusehen.

Der Wert der Spalte muss vom gewählten Datentyp sein. Folgende Datentypen sind verfügbar:

- **STRING** (Zeichenkette)
- **NUMBER** (Ganzzahliger numerischer Wert)
- **REAL** (numerischer Wert mit Nachkommastellen)
- **DATE** (Datum)
- **BOOLEAN** (Wahrheitswert)

Wählen Sie den gewünschten Datentyp aus.

 Der zugewiesene Wert kann beispielsweise aus einem einfachen statischen Wert, einer **EVAL-Anweisung** (mit **Funktion**) oder der Abfrage einer **Datenstromvariablen** bestehen.

Bestätigen Sie die eingetragenen Inhalte mit Ok oder verwerfen Sie die neue Spalte mit Abbrechen.

Die Spalte wurde nun eingefügt. Sie können die Spalte jetzt bearbeiten, verschieben oder sie auch wieder löschen.

- Klicken Sie auf das  Kontextmenü *Abb. A (1)* einer Spalte und wählen Sie anschließend  Spalte bearbeiten (4), um die Spalte zu bearbeiten.
- Klicken Sie auf das  Kontextmenü (1) einer Spalte und wählen Sie anschließend   Spalte nach links/rechts verschieben (3), um eine vorhandene Spalte zu bearbeiten.
- Klicken Sie auf das  Kontextmenü (1) einer Spalte und wählen Sie anschließend  Spalte löschen (5), um die Spalte zu löschen.

**Spalten bearbeiten**

Nach dem Klick auf  öffnet sich das gleiche Fenster wie bei der Erstellung der Spalte:

Abbildung D:

Sie können die Spalte jetzt bearbeiten. Klicken Sie nach der Bearbeitung auf Ok, um die Änderungen zu übernehmen oder auf Abbrechen, um sie zu verwerfen.

### **Spalten löschen**

Nach dem Klick auf  löschen Sie ohne weitere Abfrage die Spalte.

 Sollten Sie die Spalte unabsichtlich gelöscht haben, können Sie die Regel wieder auf den letzten gespeicherten Zustand zurücksetzen. Mehr dazu finden Sie [hier](#).

## Zeile hinzufügen, bearbeiten, löschen und verschieben

Zeilen bilden die Auswahlmöglichkeiten einer Regel.

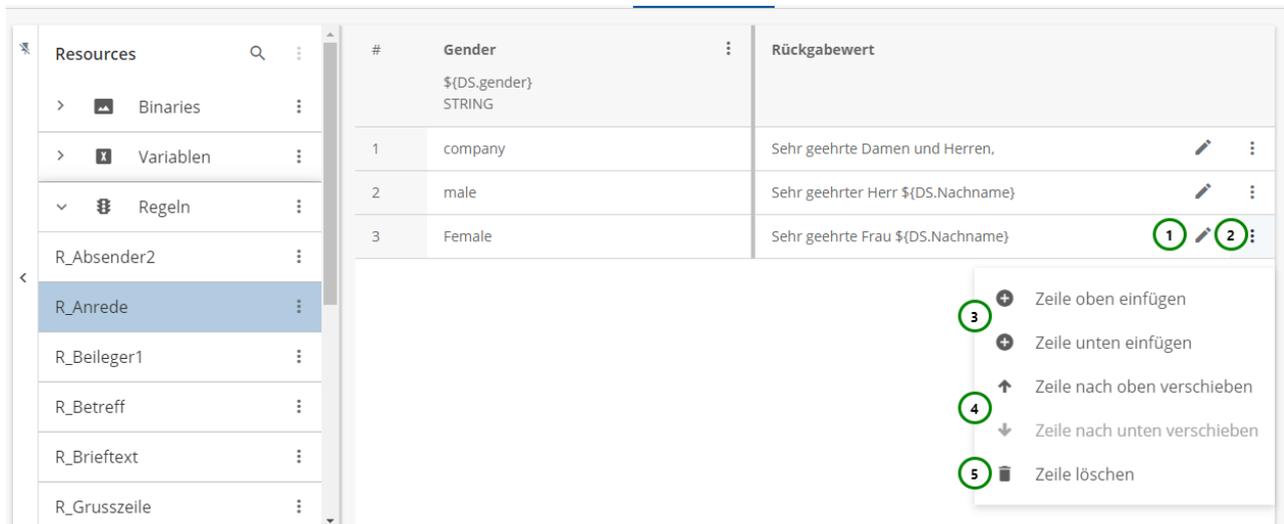


Abbildung A

Die Bearbeitung einer vorhandenen Zeile, beginnen Sie mit einem Klick auf .

Um eine Zeile einzufügen, öffnen Sie bitte das Kontextmenü [Abb. A \(2\)](#) und wählen anschließend Zeile oben/unten einfügen (3).

Es entsteht eine neue Zeile, in der Sie direkt im Bearbeitungsmodus sind:

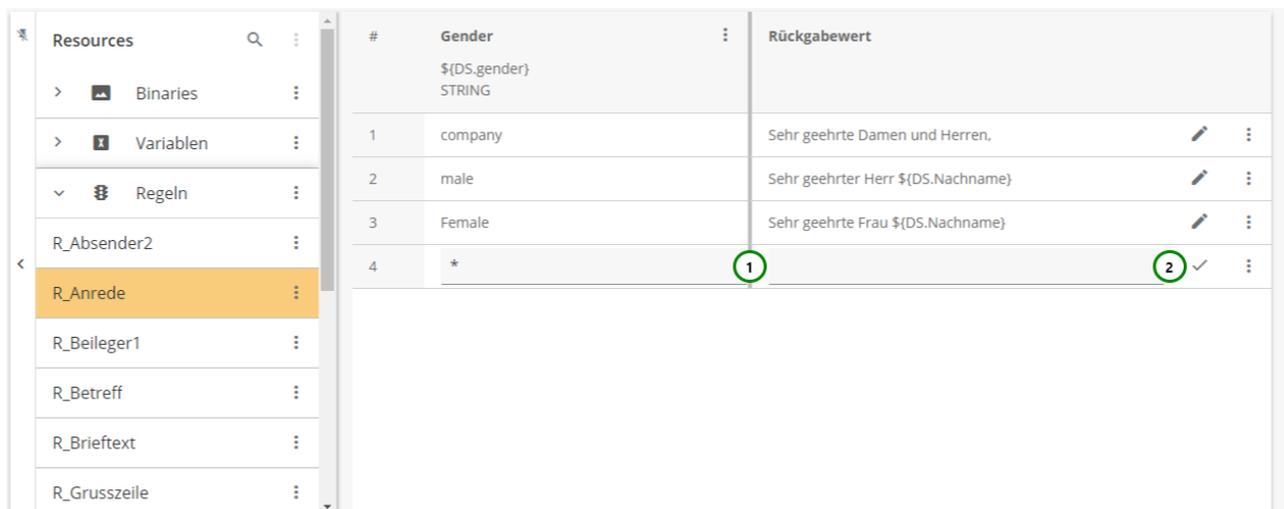


Abbildung B

In der [Abb. B \(1\)](#) wurde eine leere Zeile eingefügt. Die neue Zeile hat in allen vorhandenen Spalten den ALWAYSTRUE -Operator \*, die Zeile trifft also immer auf alle Bedingungen zu ([Definition Regeln](#)).

Der Rückgabewert der Zeile ist leer. Sie können die Zeile jetzt bearbeiten und mit (2) abschließen.

Eine Zeile enthält für jede Spalte einen Wert, für den die Spaltenbedingung geprüft wird, und einen Rückgabewert. Die Regel liefert dann den Rückgabewert der ersten zutreffenden Zeile, bei der alle Spaltenwerten auch den Zeilenwerten entsprechen (Abb. D).

### Beispiel:

#	Gender	Rückgabewert
	\${DS.gender} STRING	
1	female	Sehr geehrte Frau \${DS.Nachname}
2	male	Sehr geehrter Herr \${DS.Nachname}
3	*	Sehr geehrte Damen und Herren,

Abbildung D

Hier wird eine Regel mit einer Spalte definiert: **Gender** vom Datentyp String mit dem Wert " \${DS.Gender} ", wobei DS.Gender eine **Datenstromvariable** des Transaktions-Formulars ist, die das Geschlecht des Kunden enthält.

Die Regel gibt nun im Fall, dass DS.Gender "female" enthält, den STRING "Sehr geehrte Frau \${DS.Nachname} ", im Fall von "male" den String "Sehr geehrter Herr \${DS.Nachname} " und in allen anderen Fällen "Sehr geehrte Damen und Herren" zurück.

`\${DS.Nachname}` ist ebenfalls eine Datenstromvariable, die den Nachnamen des Kunden beinhaltet. So lässt sich eine gezielte Anrede erstellen. Weitere Beispiele finden Sie [hier](#).

### Sie können

- ★ den **Standardwert** auswählen.  
Der Standardwert ist der ALWAYSTRUE -Operator \*. Wenn sie \* angeben, ist die Bedingung immer erfüllt.
- ★ einen **eigenen Wert** definieren.  
Geben Sie den Wert an, den Sie der Zeile in dieser Spalte geben möchten. Dieser Wert wird dann auf die Spaltenbedingung (den Wert der jeweiligen Spalte) geprüft, wenn die Regel angewandt wird. (In der Spalte Rückgabewert ist der Wert der Wert, den die Regel annimmt, falls die Zeile zutrifft (**Definition Regel**)).

### Zeilen löschen

Wählen Sie die Zeile aus, die Sie löschen möchten. Öffnen Sie dann das  Kontextmenü Abb. A (2) und wählen anschließend  löschen (5). Die Zeile wird nun ohne weitere Abfrage gelöscht.

**i** Sollten Sie die Zeile unabsichtlich gelöscht haben, können Sie die Regel wieder auf den letzten gespeicherten Zustand zurücksetzen. Mehr dazu finden Sie [hier](#).

**Zeilen verschieben**

Bei Regeln gilt, dass Zeilen von oben nach unten in ihrer Priorität abnehmen. Das bedeutet, dass wenn 2 oder mehr Zeilen den Spaltenbedingungen entsprechen, die Regel den Rückgabewert der obersten Zeile annimmt, alle gültigen Zeilen darunter werden ignoriert (**Definition Regel**).

Um die Priorität von Zeilen festzulegen, können Sie Zeilen nach oben und unten verschieben. Öffnen Sie dazu das **⋮** Kontextmenü **Abb. A (2)** der Zeile, die Sie verschieben wollen, und klicken Sie auf **↑ ↓** Zeile aufwärts/abwärts verschieben (2) .

**Beispiel:**

#	Gender \${DS.gender} STRING	Rückgabewert <b>1</b>	#	Gender \${DS.gender} STRING	Rückgabewert <b>2</b>
1	female	Sehr geehrte Frau \${DS.Nachname}	1	female	Sehr geehrte Frau \${DS.Nachname}
2	male	Sehr geehrter Herr \${DS.Nachname}	2	*	Sehr geehrte Damen und Herren,
3	*	Sehr geehrte Damen und Herren,	3	male	Sehr geehrter Herr \${DS.Nachname}

Abbildung E

In **Abb. E** sind jeweils Zeile #2 und Zeile #3 vertauscht. Gehen wir nun davon aus, dass im Datenstrom die Variable DS.Gender = "male" ist, so wird, aufgrund der Interpretationsreihenfolge, der Rückgabewert in (1) "Sehr geehrter Herr \${DS.Nachname}" und in(2) "Sehr geehrte Damen und Herren," sein.

Sie können hier die **Eingabe-Hilfe** verwenden

**i** Die Änderungen werden erst angewandt, nachdem die Regel gespeichert wurde.

## Physikalischen Beileger erstellen und ändern

Physikalische Beileger legen fest, welches Material dem Dokument während des Herstellungsprozesses beigelegt werden soll.

Falls Sie sich noch nicht mit physikalischen Beileger vertraut gemacht haben, eine Beschreibung finden Sie [hier](#).

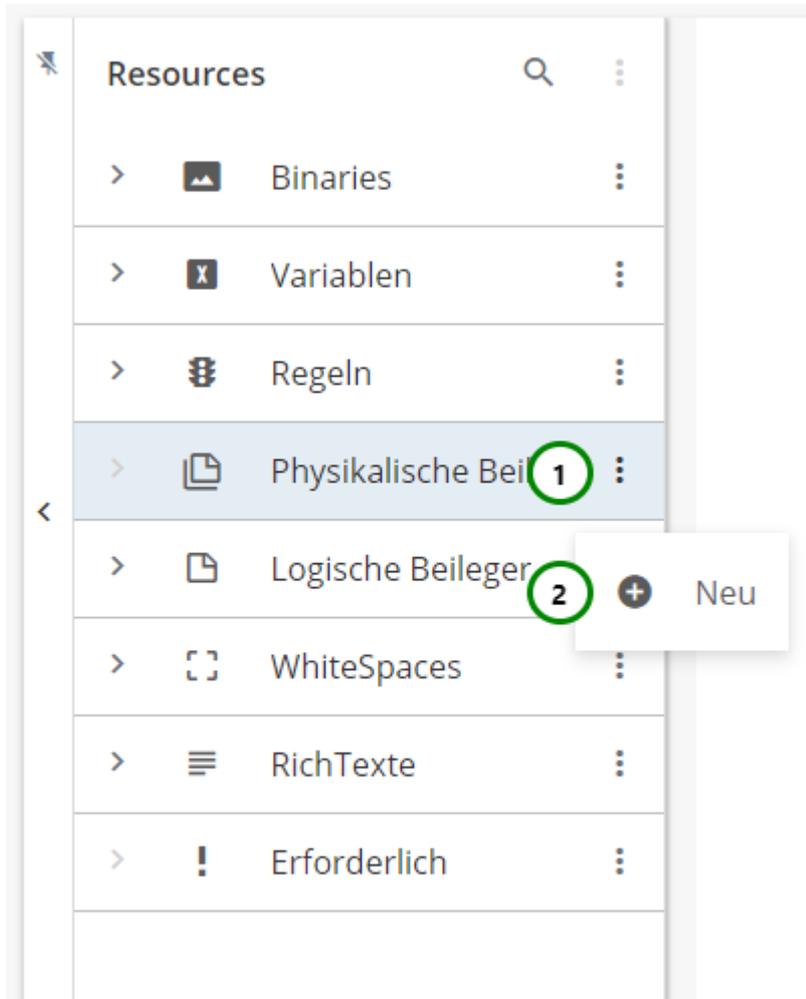


Abbildung A

In dem Ressourcen-Typ  Physikalische Beileger finden Sie alle bereits angelegten physikalischen Beileger. Klappen Sie dazu den Ressourcen-Typ auf  .

Um einen neuen physikalische Beileger zu erstellen, klicken Sie bitte auf das  Kontextmenü [Abb. A \(1\)](#) des Ressourcen-Typs und wählen Sie nun  Neu (2) aus.

Um Änderungen vorzunehmen, wählen Sie die Ressource, die Sie ändern möchten, aus der Übersichtsliste aus [Abb. B \(1\)](#). Nun sehen Sie in dem Inhaltsbereich die Informationen zu dieser Ressource:

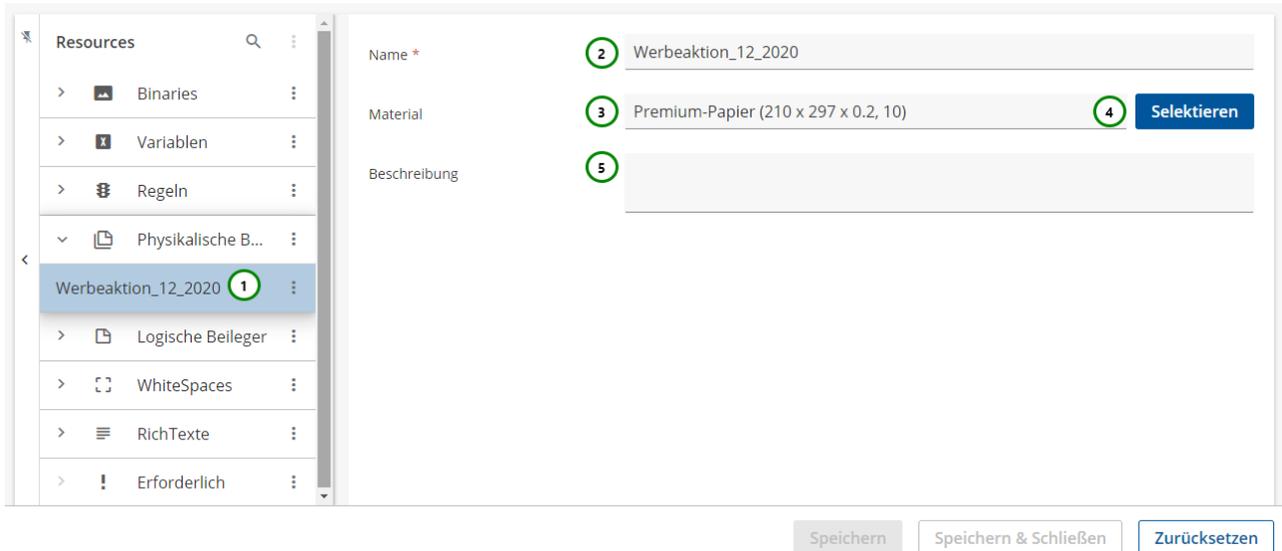


Abbildung B

**Sie können dem physikalischen Beileger:**

- ★ einen **Namen** geben *Abb. B (2)*.  
 Sie müssen dem physikalischen Beileger einen Namen geben, unter dem Sie den Beileger dann ansprechen und nutzen.  
**Beachten Sie:** In einer Schema-Version/Bibliotheks-Version dürfen keine 2 Ressourcen denselben Namen haben. Hat eine Ressource aus einer Schema-Version denselben Namen wie eine Ressource aus einer eingebundenen Bibliotheks-Version, überschreibt die Schema-Ressource die Bibliotheks-Ressource.
- ★ ein **Material** zuweisen.  
 Klicken Sie dazu auf *(4) Selektieren*. Eine Übersicht über die im Arbeitsbereich vorhandenen Materialien öffnet sich *(Abb. C)*. Wählen Sie dort das Material aus, das Sie dem Beileger zuweisen möchten. Um das Material zu entfernen, klicken Sie auf **X**.  
 Das zugeordnete Material wird unter *(3)* angezeigt.
- ★ eine **Beschreibung** geben *(5)*.  
 Die Beschreibung ist optional.

⚠ Beachten Sie, dass die Umbenennung einer Ressource alle Verweise auf die Ressource ungültig macht, insbesondere in Erweiterungen. Wird die Ressource in einer Erweiterung oder in einer anderen Ressource verwendet, müssen Sie nach dem Umbenennen alle Verweise anpassen.

Um mit dem physikalischen Beileger arbeiten zu können, müssen Sie ihm ein Material zuweisen, das Sie zuvor erstellt haben. Lesen Sie dazu bitte den Abschnitt über das **Arbeiten mit Materialien**.  
 Um ein Material für den Beileger auszuwählen, klicken Sie bitte auf *Abb. B (4)*. Folgender Dialog öffnet sich:

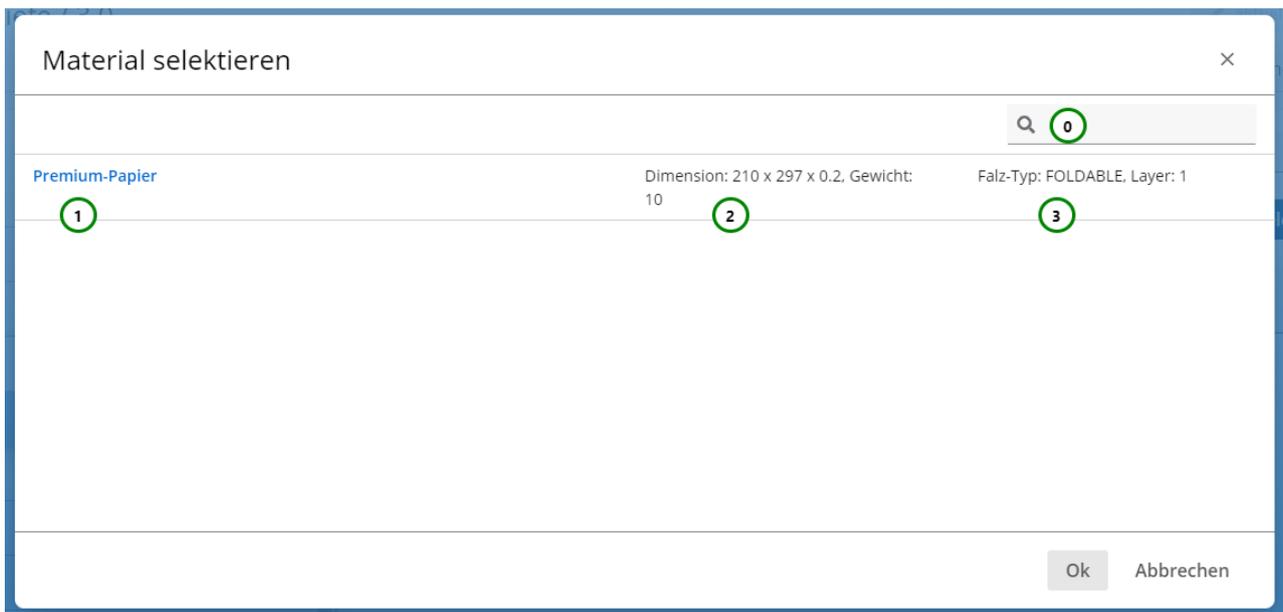


Abbildung C

**Sie sehen:**

- Den **Namen** des Materials *Abb. C (1)*.
- Die **Dimension** und das **Gewicht** des Materials *(2)*.
- Den **Falz-Typ** des Materials *(3)*.

Sie haben die Möglichkeit, die Übersichtstabelle zu durchsuchen *Abb. C (0)*. Detaillierte Informationen über die Suchfunktion im R-S finden Sie [hier](#).

Wählen Sie das passende Material aus der Übersichtstabelle aus. Klicken Sie anschließend auf Ok, um die Zuweisung abzuschließen oder auf Abbrechen, um die Auswahl zu verwerfen.

Mit *Abb. B (4)* können Sie die Materialzuweisung auch ändern. Wenn Sie die Zuweisung löschen möchten, klicken Sie bitte auf **X**.

**!** Wenn Sie die Materialzuweisung eines Beilegers löschen verlieren auch alle **Erweiterungen**, die den Beileger als Ressource nutzen, das Material. Der Beileger kann dann nicht genutzt werden und bei der Vorschaugenerierung tritt ein Fehler auf.

## Logischen Beileger erstellen und ändern

Logische Beileger ermöglichen es, dem Dokument individuelle Beileger hinzuzufügen, die beim Druckprozess mit erstellt werden.

Falls Sie sich noch nicht mit logischen Beileger vertraut gemacht haben, eine Beschreibung finden Sie [hier](#).

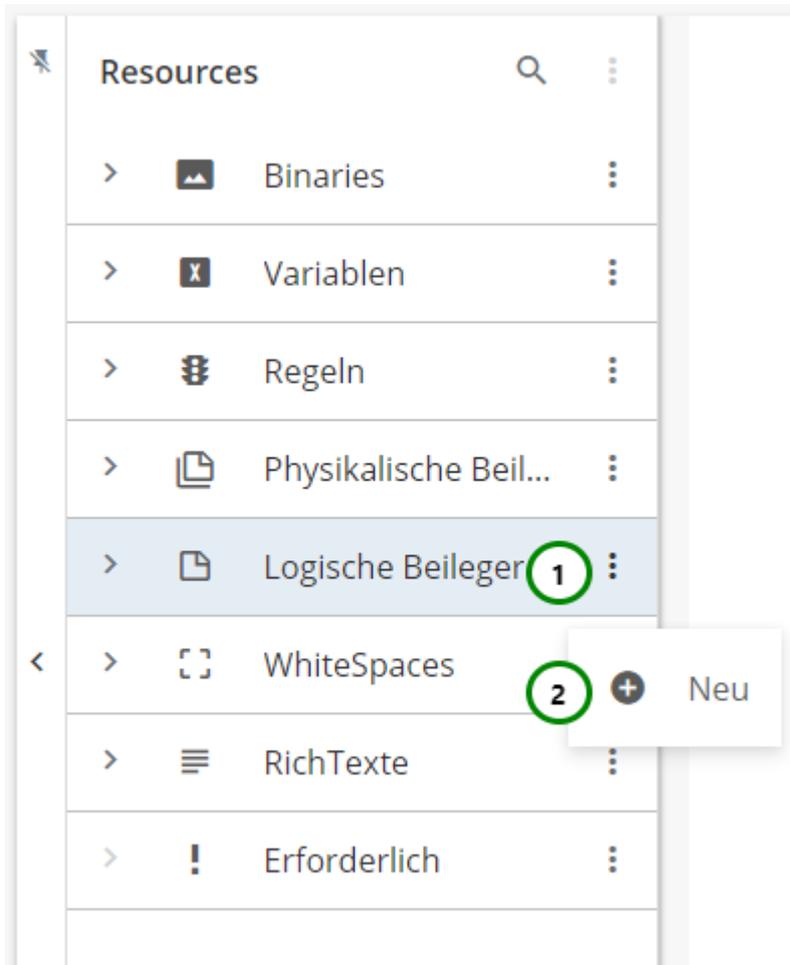


Abbildung A

In den Ressourcen-Typ  Logische Beileger finden Sie alle bereits angelegten logischen Beileger. Klappen Sie dazu den Ressourcen-Typ auf .

Logische Beileger werden als komplexe Elemente betrachtet und bestehen daher aus einer Eigenschaften- und Designseite.

Um einen neuen logische Beileger zu erstellen, klicken Sie bitte auf das  Kontextmenü [Abb. A \(1\)](#) des Ressourcen-Typs und wählen Sie nun  Neu (2) aus.

Sie landen zuerst in den Eigenschaften des logische Beilegers.

Um die Eigenschaften zu ändern, wählen Sie die Ressource, die Sie ändern möchten, aus der Übersichtsliste aus [Abb. B \(1\)](#) und öffnen Sie das  Kontextmenü (1).

Nach einem weiteren Klick auf  Eigenschaften öffnet sich folgender Inhalt:

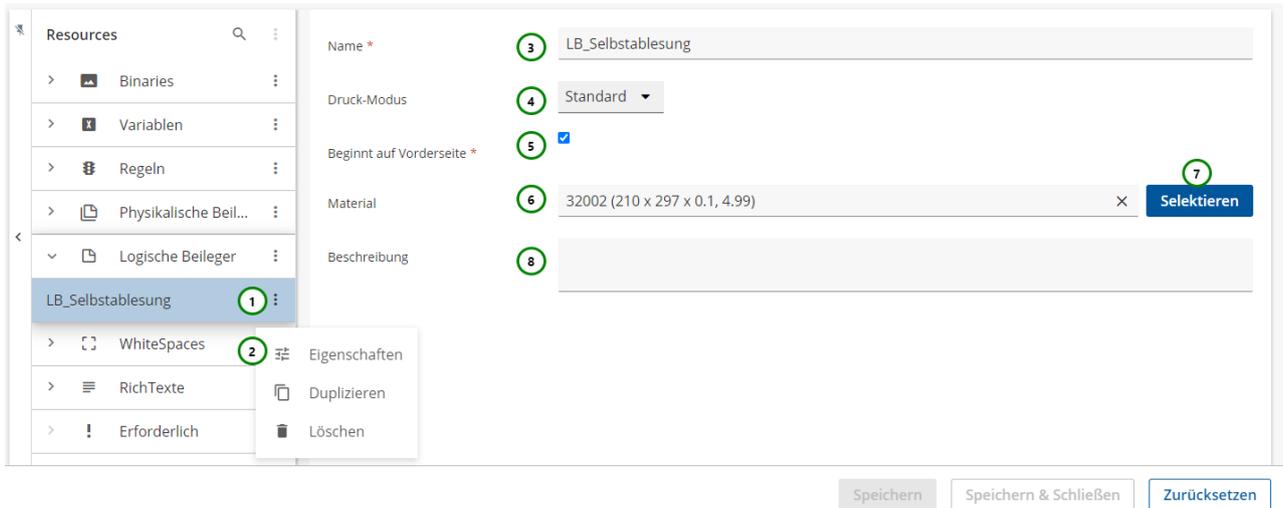


Abbildung B

**In den Eigenschaften können Sie dem logischen Beileger:**

- ★ einen **Namen** geben *Abb. B (3)*.  
Sie müssen dem logischen Beileger einen Namen geben, unter dem Sie den Beileger dann ansprechen und nutzen.  
**Beachten Sie:** In einer Schema-Version/Bibliotheks-Version dürfen keine 2 Ressourcen denselben Namen haben. Hat eine Ressource aus einer Schema-Version denselben Namen wie eine Ressource aus einer eingebundenen Bibliotheks-Version, überschreibt die Schema-Ressource die Bibliotheks-Ressource.
- ★ eine **Beschreibung** geben *(8)*.  
Die Beschreibung ist optional.

**Um den logischen Beileger zu konfigurieren, können Sie:**

- ★ einen **Druck-Modus** auswählen. Klicken Sie dazu auf [Abb. B \(4\)](#) und wählen Sie aus der Liste den passenden Modus aus:
  - **DEFAULT** für den Standard-Modus,
  - **SIMPLEX** für einseitigen Druck,
  - **DUPLEX** für beidseitigen Druck.
- ★ Bestimmen, ob ein logischer Beileger immer auf einer **Vorderseite** beginnen soll (5).
- ★ Dem Beileger ein **Material** zuweisen (6).  
 Klicken Sie dazu auf (7). Eine Übersicht über die im Arbeitsbereich vorhandenen Materialien öffnet sich [Abb. C](#). Wählen Sie dort das Material aus, welches Sie dem Beileger zuweisen möchten. Um das Material zu entfernen, klicken Sie auf .  
 Im Transaktions-Formular ist vorgegeben, ob formularfremdes Material (Papier) benutzt werden darf. Ist das nicht der Fall, müssen logische Beileger auf demselben Papier gedruckt werden wie die übrigen Dokumente und es erscheint folgende Meldung:  
**WARNUNG: In der Produktion wird das vom Transaktions-Formular vorgegebene Material genutzt**
- ★ Den logischen Beileger **gestalten**. Lesen Sie dazu den Abschnitt über das [Design](#).

 Beachten Sie, dass die Umbenennung einer Ressource alle Verweise auf die Ressource ungültig macht, insbesondere in Erweiterungen. Wird die Ressource in einer Erweiterung oder in einer anderen Ressource verwendet, müssen Sie nach dem Umbenennen alle Verweise anpassen.

Um mit dem logischen Beileger arbeiten zu können, müssen Sie ihm ein Material zuweisen, das Sie zuvor erstellt haben. Lesen Sie dazu bitte den Abschnitt über das [Arbeiten mit Materialien](#).  
 Um ein Material für den Beileger auszuwählen, klicken Sie bitte auf [Abb. B \(7\)](#). Es öffnet sich ein neues Fenster:

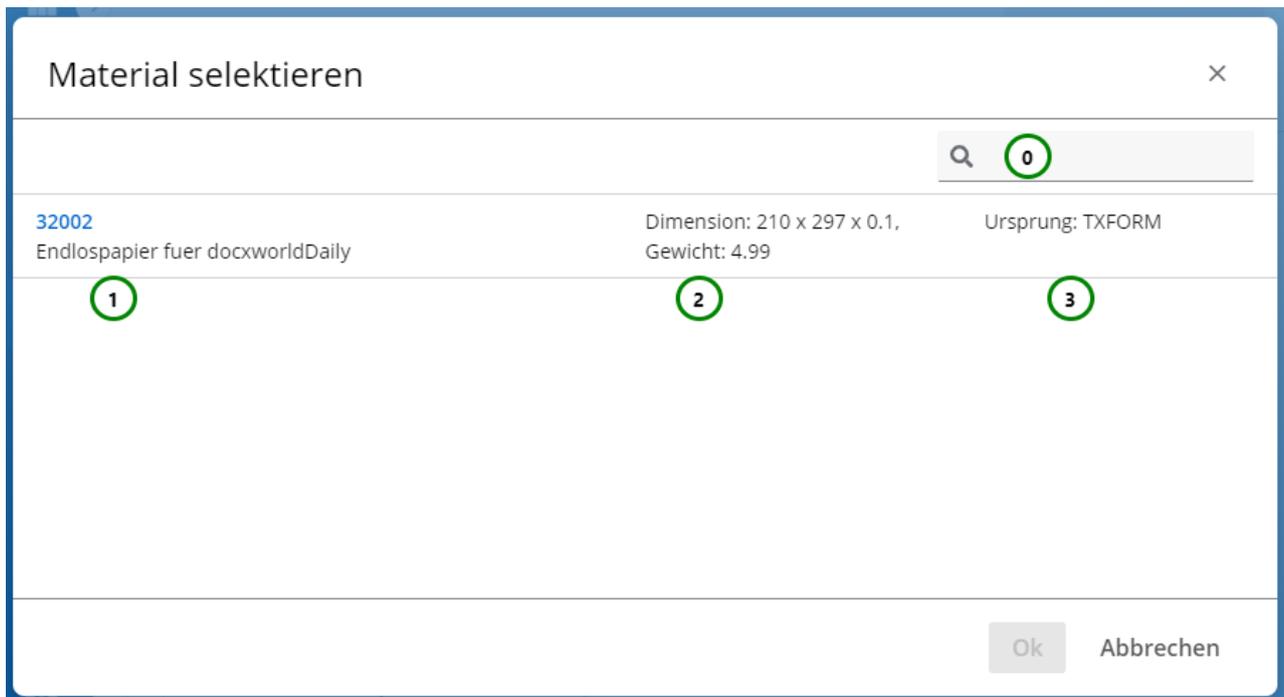


Abbildung C

**Sie sehen :**

- Name und Beschreibung des Materials Abb. C (1).
- Dimensionen und Gewicht des Materials (2).
- Herkunft des Materials (3).

Wenn durch das Transaktions-Formular festgelegt wurde, dass nur das Material des Transaktions-Formulars benutzt werden darf, ist die Material-Übersichtstabelle Abb. C nur mit dem vorgegebenen Materialien gefüllt.

Sie haben die Möglichkeit, die Übersichtstabelle zu durchsuchen  (0). Detaillierte Informationen über die Suchfunktion im R-S finden Sie [hier](#) . Wählen Sie das passende Material aus der Übersichtstabelle (1) aus. Klicken Sie anschließend auf Ok, um die Zuweisung abzuschließen oder auf Abbrechen, um die Auswahl zu verwerfen.

Nachdem Sie dem Beileger ein Material zugewiesen haben, werden Ihnen die Abmessungen des Materials angezeigt Abb. B (6).

 Wenn Sie die Materialzuweisung eines Beilegers löschen, verlieren auch alle **Erweiterungen**, die den Beileger als Ressource nutzen, das Material. Der Beileger kann dann nicht mehr genutzt werden.

## Design des logischen Beilegers

In der Design-Ansicht können Sie den logischen Beileger gestalten. Klicken Sie dazu den Logischen Beileger aus der Übersichtsliste an [Abb. A \(3\)](#). Sie sehen folgenden Inhalt:

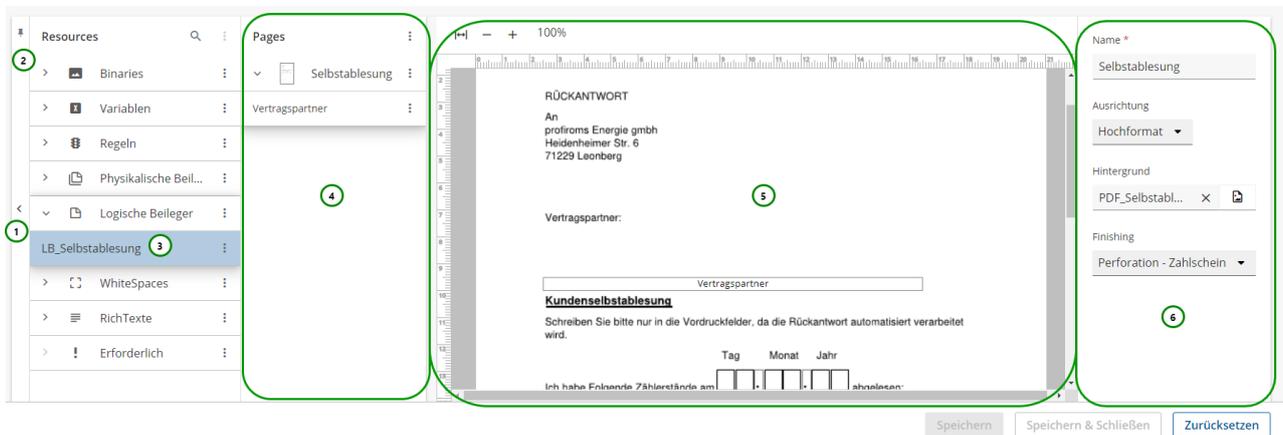


Abbildung A

In der Design-Ansicht wird die Ressourcen-Übersichtstabelle automatisch ausgeblendet, sobald Ihr Mauszeiger die Tabelle verlässt. Sie können die Übersichtstabelle erneut mit [Abb. A \(1\)](#) einblenden. Um das automatische Ausblenden zu verhindern, können Sie mit  (2) die Übersicht anpinnen.

### Die Übersicht ist in 3 Bereiche aufgeteilt:

- Die **Seiten-Übersicht** [Abb. A \(4\)](#) : Hier sehen und verwalten Sie die Seiten des Beilegers.
- Die **Layout-Vorschau** des Beilegers (5) : Hier wird Ihnen das Design der Seite und das Layout der Bereiche angezeigt.
- Die **Objekt- Eigenschaften** (6) : Hier sehen und verwalten Sie die Eigenschaften von Seiten- und Layout-Bereichen.

Neu erstellte logische Beileger haben noch keinen Inhalt. Nachdem Sie bereits im vorherigen Kapitel die Eigenschaften des logischen Beileger definiert haben, können nun mit dem Design beginnen.

Zuerst müssen Sie eine Seite für den Beileger anlegen. Klicken Sie dazu auf das  Kontextmenü [Abb. B \(1\)](#) der Seiten-Übersicht und anschließend auf  Neue Seite (2).

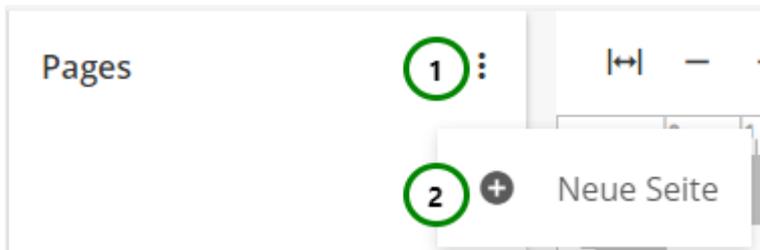


Abbildung B

Es wird eine neue Seite eingefügt [Abb. C \(1\)](#):

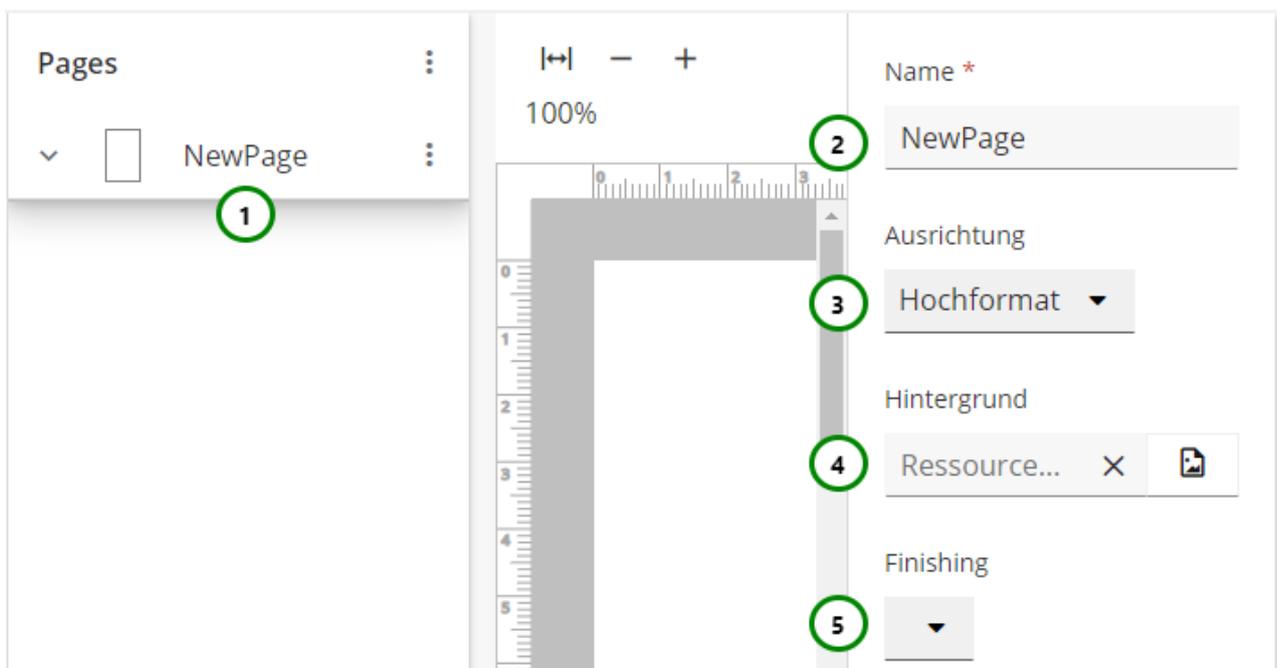


Abbildung C

**Sie können:**

- ★ Der Seite einen **Namen** geben *Abb. C (2)*.
  - ★ Der Seite eine **Ausrichtung** geben. Klicken Sie auf (3). Hier können Sie wählen zwischen:
    - *Hochformat*
    - *Querformat*
  - ★ Einen **Hintergrund** für die Seite wählen (4).
  - ★ Ein **Finishing** (z.B.: Lochung, Stanzung, Ösung, ...) wählen (5).  
 Finishing sind Papier-Optionen für Seiten/Blätter von logischen Beilegern, welche während der Verarbeitung in den Druckzentren angeboten werden. Diese Optionen müssen mit dem Druckbetrieb abgestimmt sein. Im Eingabefeld für das Finishing dürfen nur numerische Werte zwischen 1 und 99 eingegeben werden, welche mit dem Druckbetrieb abgestimmt sind. Der Administrator des Redaktions-Systems kann mögliche Werte zur einfachen Vorauswahl in einer Liste konfigurieren - bitte wenden Sie sich hierbei an Ihren Administrator.  
 Ist der manuell eingegebene Wert nicht konfiguriert, wird Ihnen das mit roter Schrift signalisiert. Die Finishing-Option wird an die nachgelagerte Software weitergegeben (z.B. ReportWriter).
- ⚠ Soll das Eingabefeld für das Finishing keinen Wert enthalten, so kann der entsprechende Inhalt über das Eingabefeld gelöscht werden.

**Hintergrund der Seite wählen**

Nachdem Sie nun die Seite angelegt haben, können Sie einen Hintergrund  *Abb. C (4)* auswählen. Es öffnet sich folgender Dialog:

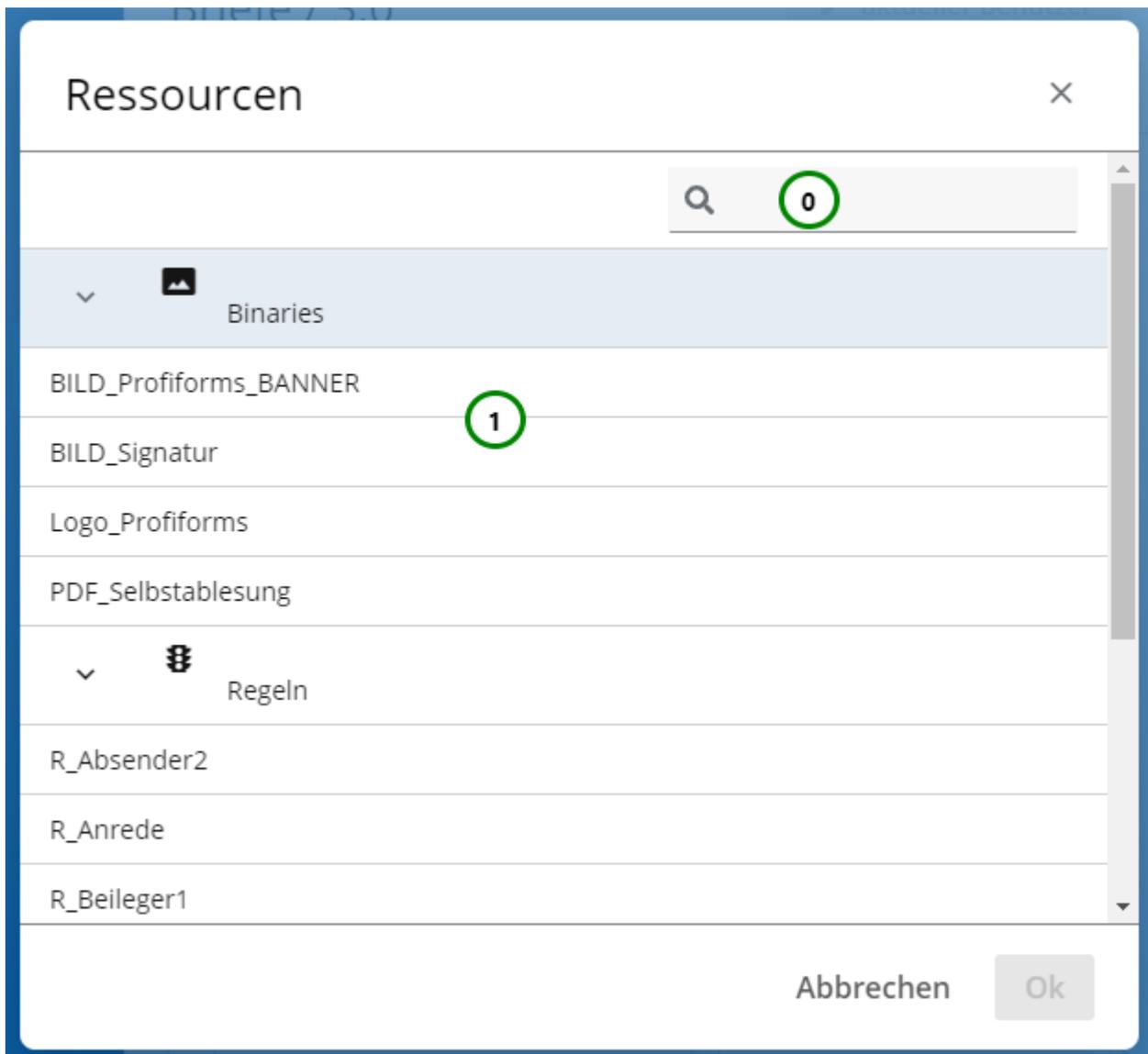
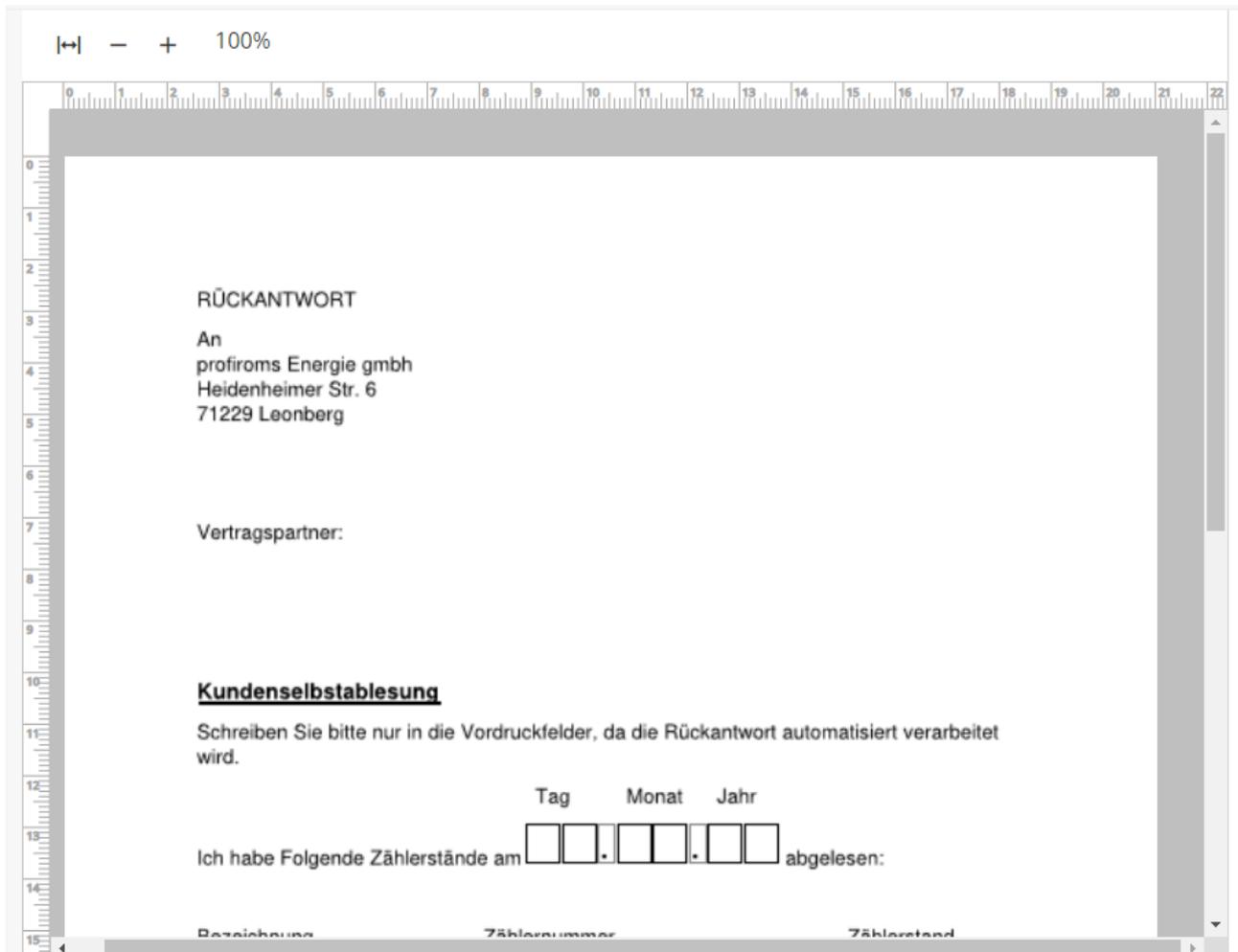


Abbildung D

Wählen Sie aus der Ressourcen-Übersicht die Ressource aus *Abb. D(1)*, welche Sie als Hintergrund für die Seite verwenden möchten. Mit Hilfe von  (0) können Sie nach einer Ressource suchen. Klicken Sie anschließend auf Ok, um die Auswahl abzuschließen oder auf Abbrechen, um die Zuweisung abubrechen.

Sie sehen nun den Hintergrund auf der **Layout-Vorschau Abb. A (5)**, sofern Sie eine Binäre-Ressource, die ein Bild oder PDF beinhaltet, ausgewählt haben. Für den Ressourcen-Typ  Regeln kann keine Vorschau erzeugt werden, da der Rückgabewert erst zur Laufzeit bekannt wird.



Die Größe der Seite wird dabei durch das Material festgesetzt, das Sie dem Beileger in den **Eigenschaften** zugewiesen haben.

### Umgang mit Seiten

Nachdem Sie eine Seite angelegt haben, können Sie diese auch löschen, duplizieren, verschieben und Objekte erstellen. Öffnen Sie dazu das  Kontextmenü Abb. E (1) der Seite.

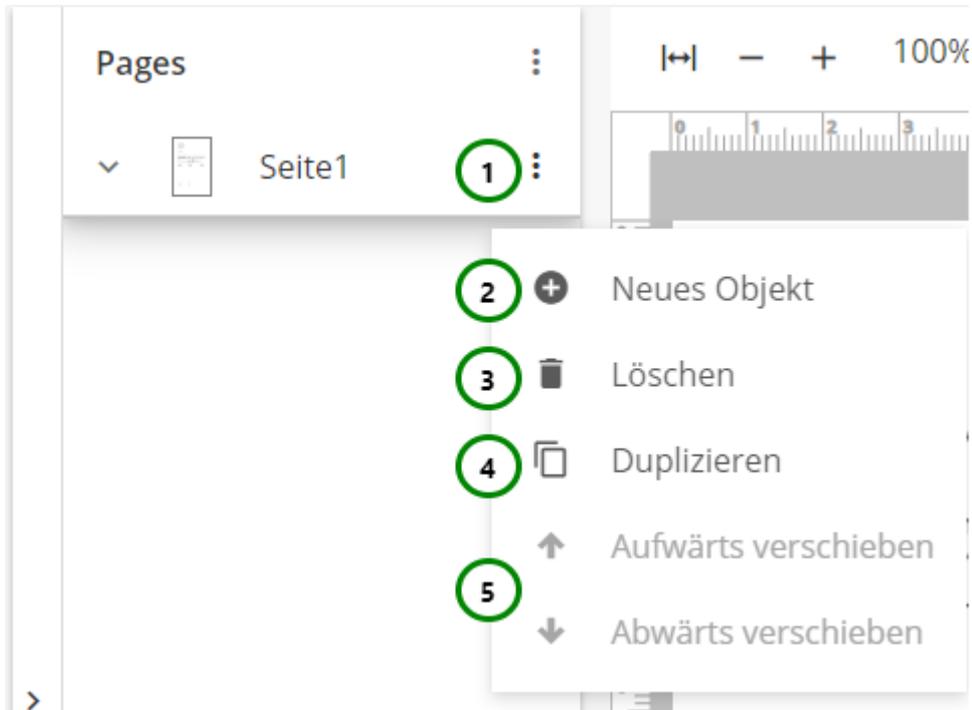


Abbildung E

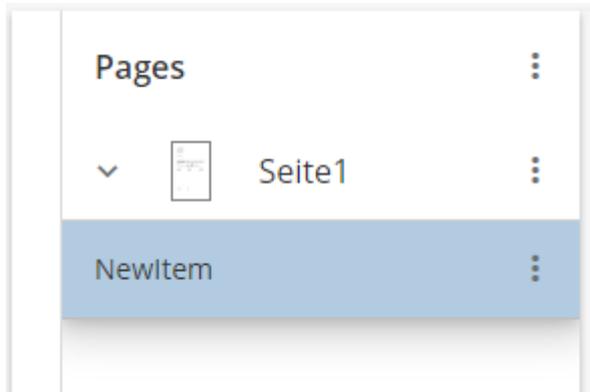
#### Sie können:

-  Neue Objekte auf dieser Seite erstellen Abb. E (2).
- Diese Seite  Löschen (3).
- Diese Seite  Duplizieren (4)
- Die Seite   Auf- und Abwärts verschieben (5)

### Objekte anlegen und bearbeiten

Sie können auf den Seiten des logischen Belegers Objekte anlegen und platzieren. Diese Objekte sind wie Erweiterungspunkte zu betrachten und können mit Ressourcen verknüpft werden. So wird zur Laufzeit die individuelle Ressource in das Objekt hineingeladen und kann so einen personalisierten Beleger abbilden.

Mit einem Klick auf  Neues Objekt Abb. E(2) erstellen Sie ein neues Objekt:



Nach dem Selektieren des neuen Objektes, sehen Sie in den Objekt-Eigenschaften Abb. A(6) die Angaben zu diesem Objekt:

Name \*

1 NewItem

X

2 5

Y

3 5

Breite

4 20

Höhe

5 20

Ressource

6 Ressource... ×  7

8  sichtbar

9  gesperrt

Abbildung F

**Sie können:**

- ★ Dem Layout-Bereich einen **Namen** geben *Abb. F (1)*.
- ★ Die **Position und Größe** des Bereichs angeben:
  - Die **X- und Y-Koordinate** (Angabe in **mm**) des Bereichs in der Seiten (2 & 3), also die Position der linken oberen Ecke des Bereichs.
  - Die maximale **Größe** (Angabe in **mm**) des Objektes in Breite und Höhe (4 & 5).
- ★ Bestimmen, ob das Objekt im Dokument **sichtbar** (8) sein soll und ob es **in der Design-Ansicht geändert** (9) werden darf.
- ★ Dem Objekt eine **Ressource** zuordnen, die in dem Bereich angezeigt werden soll (7). Es öffnet sich der Dialog zur Ressourcenauswahl ([Hintergrund der Seite](#)).  
Ist bereits eine Ressource vorhanden, sehen Sie diese in (6).

Objekte die Sie bereits angelegt haben können Sie mit einem Klick selektieren und in den Objekt-Eigenschaften *Abb. A (6)* verändern.

Auch das Objekt besitzt ein   
Kontextmenü *Abb. G (1)*, mit dem Sie weitere Anpassungen vornehmen können:

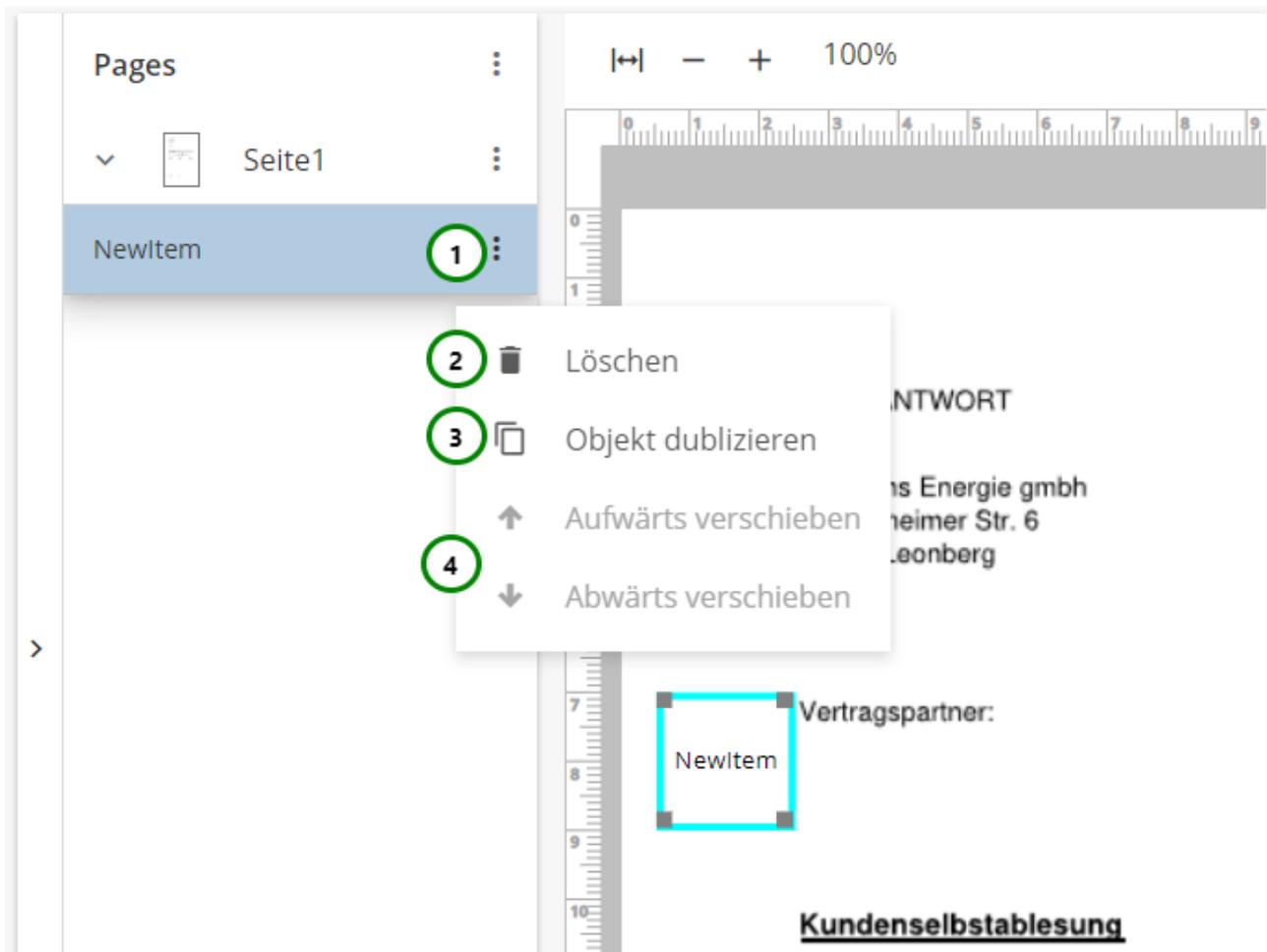


Abbildung G

**Sie können:**

- Das Objekt  **löschen** *Abb. G (2)*.
  - Das Objekt  **duplizieren** (3).
  - Das Objekt   **Auf-** und **Abwärts** verschieben (4).
- Die Reihenfolge ist entscheidend, falls sich Objekte überlappen.

**Beispiel:**

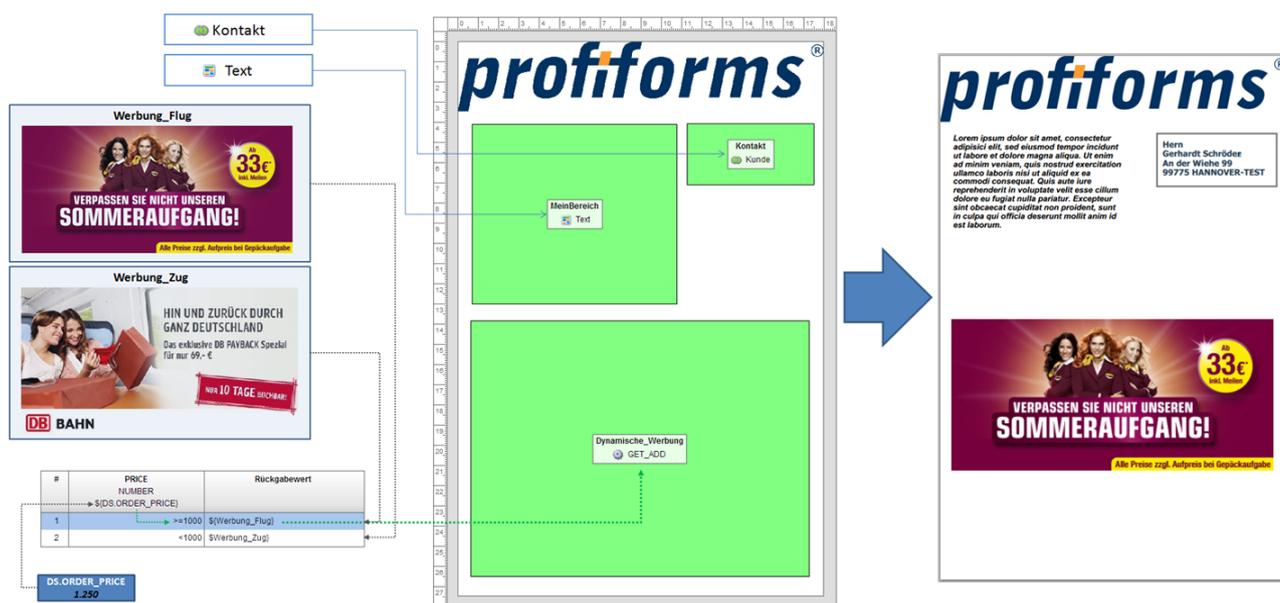


Abbildung H

Im Beispielbeleger *Abb. H* haben wir 3 Bereiche:

- **MeinBereich,**
- **Kontakt** und
- **Dynamische\_Werbung.**

**MeinBereich** verwendet den **RichText** Text, **Kontakt** die **Variable** Kunde. Diese werden direkt in den jeweiligen Bereich eingefügt. Der Bereich **Dynamische\_Werbung** benutzt eine Regel, um dynamisch eine Ressource zugeordnet zu bekommen. Je nachdem, welchen Wert die Datenstrom-Variable DS.ORDER\_PRICE hat, wird einer von zwei Werbe-Flyern ausgewählt. Ist der Wert von DS.ORDER\_PRICE kleiner als 1000, enthält der Bereich die Ressource Werbung\_Zug. Ist der Wert größer oder gleich 1000, enthält der Bereich die Ressource Werbung\_Flug (hier der Fall). Rechts sehen Sie das Endresultat.

## WhiteSpace erstellen und ändern

WhiteSpaces legen fest, wie mit ungenutzten Bereichen auf dem Dokument umgegangen werden soll. Falls Sie sich noch nicht mit WhiteSpaces vertraut gemacht haben, eine Beschreibung finden Sie [hier](#).

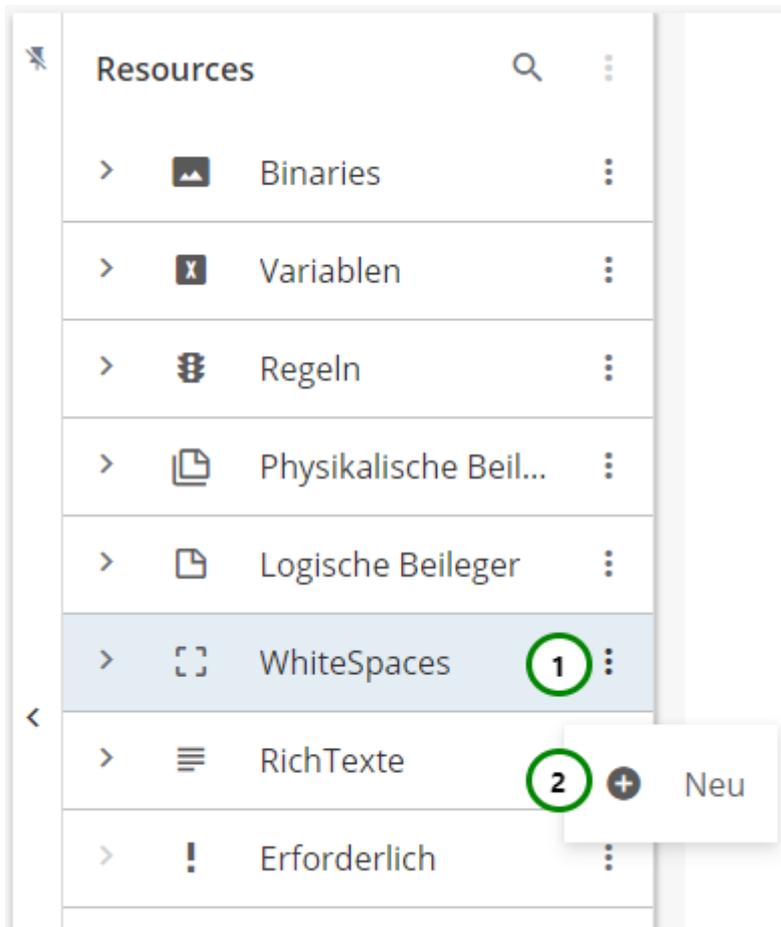


Abbildung A

In dem Ressourcen-Typ  WhiteSpaces finden Sie alle bereits angelegten logischen Beileger.

Klappen Sie dazu den Ressourcen-Typ auf  .

WhiteSpaces werden als komplexe Elemente betrachtet und bestehen daher aus Eigenschaften- und Designseite.

Um einen neuen WhiteSpace zu erstellen, klicken Sie bitte auf das  Kontextmenü [Abb. A \(1\)](#)

des Ressourcen-Typs und wählen Sie nun  Neu (2) aus. Sie landen zuerst in den Eigenschaften des WhiteSpaces.

Um die Eigenschaften zu ändern, wählen Sie die Ressource, die Sie ändern möchten, aus der Übersichtsliste aus [Abb. B \(1\)](#) und öffnen Sie das  Kontextmenü (1). Nach einem weiteren Klick auf  Eigenschaften öffnet sich der folgende Inhalt:

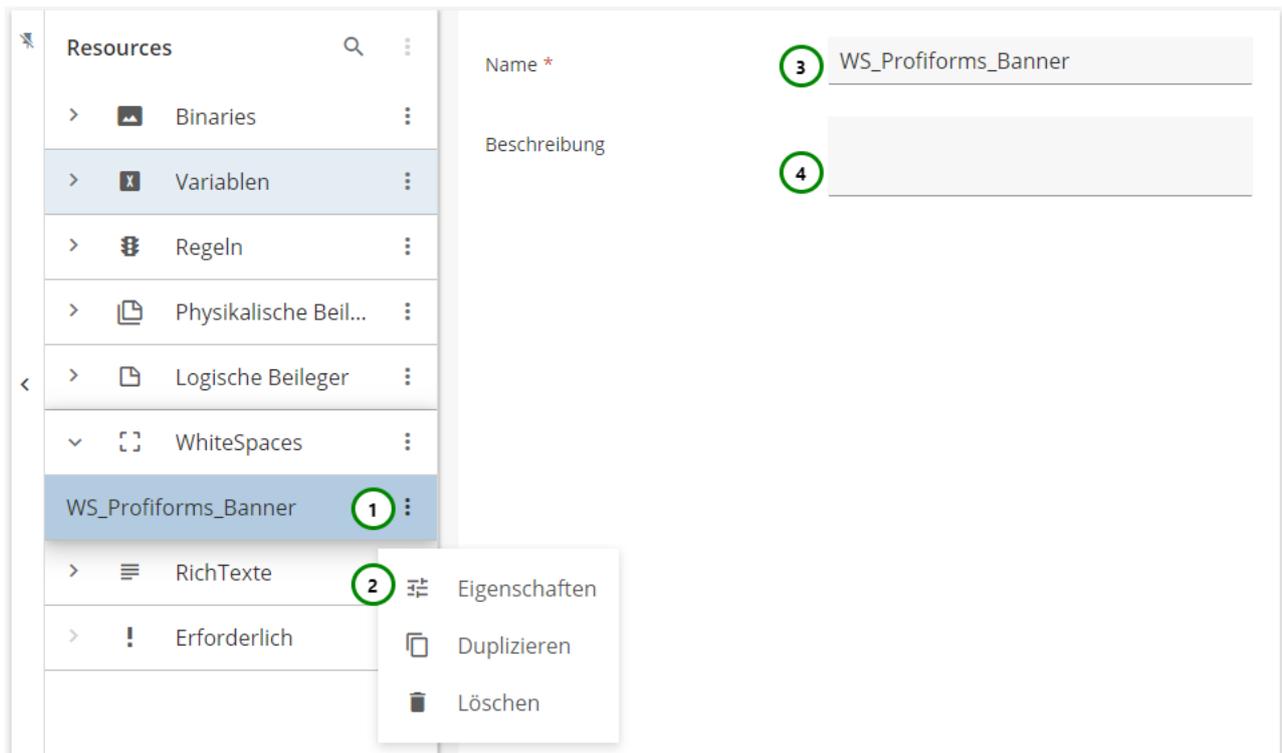


Abbildung B

**Sie können dem WhiteSpace:**

- ★ einen **Namen** geben *Abb. B (3)*.  
 Sie müssen dem WhiteSpace einen Namen geben, unter dem Sie den WhiteSpace dann ansprechen und nutzen.  
**Beachten Sie:** In einer Schema-Version/Bibliotheks-Version dürfen keine 2 Ressourcen denselben Namen haben. Hat eine Ressource aus einer Schema-Version denselben Namen wie eine Ressource aus einer eingebundenen Bibliotheks-Version, überschreibt die Schema-Ressource die Bibliotheks-Ressource.
- ★ eine **Beschreibung** geben *(4)*.  
 Die Beschreibung ist optional. Sie wird z.B. in Auswahl-Fenstern angezeigt.

⚠ Beachten Sie, dass die Umbenennung einer Ressource alle Verweise auf die Ressource ungültig macht, insbesondere in Erweiterungen. Wird die Ressource in einer Erweiterung oder in einer anderen Ressource verwendet, müssen Sie nach dem Umbenennen alle Verweise anpassen.

## WhiteSpaces designen

Um den WhiteSpace Inhalt zuzuordnen, müssen Sie auf die Designseite des WhiteSpace wechseln. Dazu genügt ein Klick auf die eben erstellte Ressource Abb. C (1) in der Ressourcen-Übersicht. Sie sehen nun eine Design-Vorschau (2).

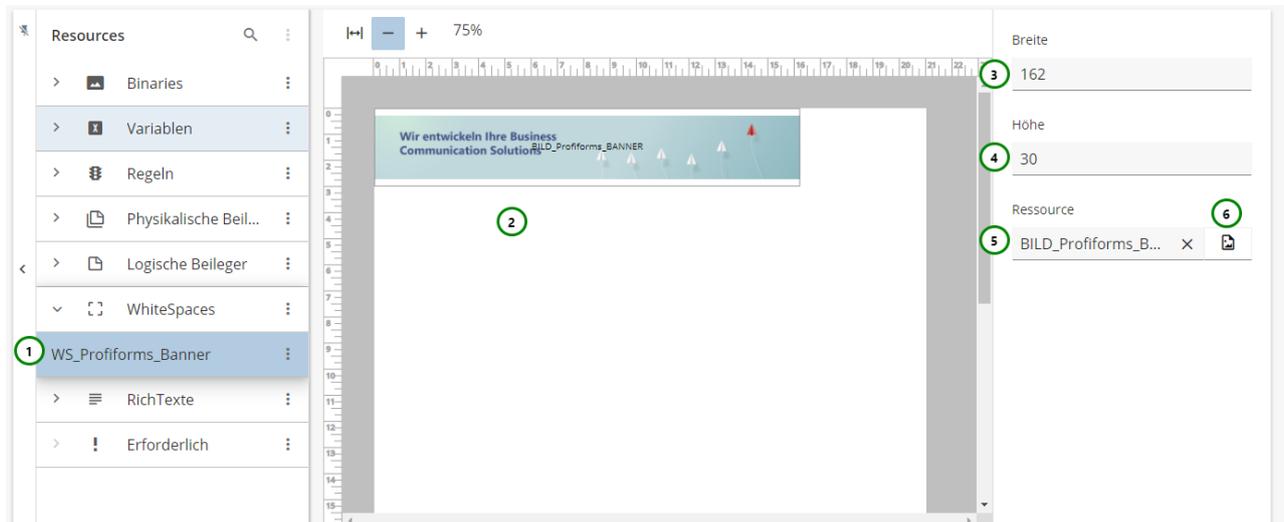


Abbildung C

### Sie können dem WhiteSpace:

- eine **Breite** und **Höhe** (Angabe in **mm**) vergeben Abb. C (3 & 4).  
Diese Maße sind ausschlaggebend, ob der WhiteSpace noch in einen leeren Bereich passt.
- eine **Ressource** zuweisen (6).

### Um dem WhiteSpace einen Inhalt zuzuordnen, müssen Sie:

Ihm eine Ressource zuweisen, die Sie zuvor erstellt haben. Lesen Sie dazu gegebenenfalls den Abschnitt über das [Erstellen von Ressourcen](#).

Dazu klicken Sie bitte auf Abb. C (6). Es öffnet sich die Ressourcenauswahl Abb. D:

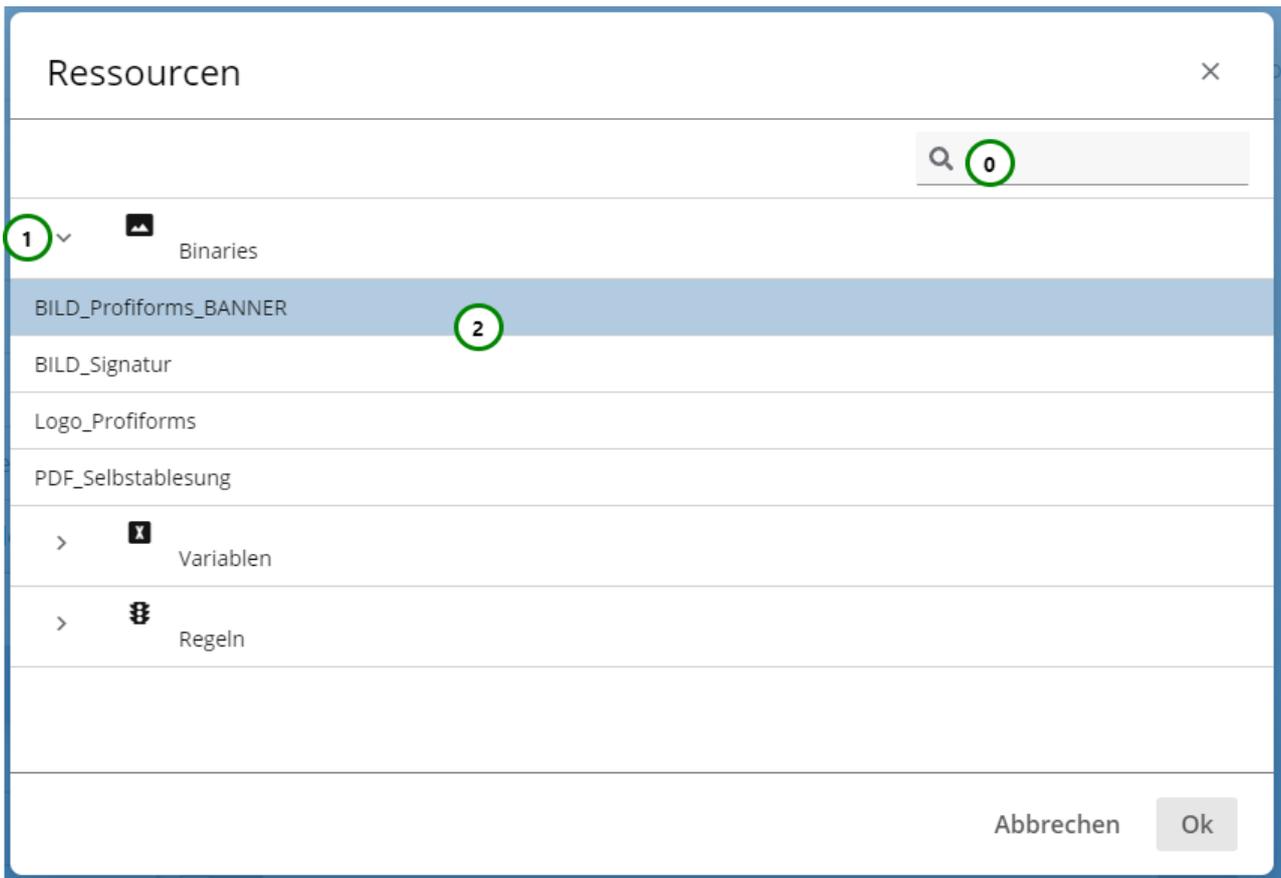


Abbildung D

Sie haben die Möglichkeit, die Ressourcenauswahl zu durchsuchen [Abb. D \(0\)](#). Detaillierte Informationen über die Suchfunktion im R-S finden Sie [hier](#).

Klappen > Sie einen Ressource-Typ aus [Abb. D \(1\)](#), um die darin enthaltenen Ressourcen (2) zu sehen. Wählen Sie nun Ihre gewünschte Ressource aus und bestätigen Sie die Zuweisung mit Ok oder brechen Sie sie mit Abbrechen ab.

Eine Vorschau des WhiteSpace kann nur mit Binaries angezeigt werden und auch erst nachdem Sie den WhiteSpace gespeichert haben.

Wenn Sie die Zuweisung löschen möchten, klicken Sie bitte auf [Abb. C \(5\)](#).

 Wenn Sie die Ressourcenzuweisung eines WhiteSpaces löschen verlieren auch alle **Erweiterungen**, die den WhiteSpace als Ressource nutzen, die Ressource.

## RichText erstellen und ändern

RichTexte bieten eine Möglichkeit, formatierten Text zu erstellen. Falls Sie sich noch nicht mit RichTexten vertraut gemacht haben, eine Beschreibung finden Sie [hier](#).

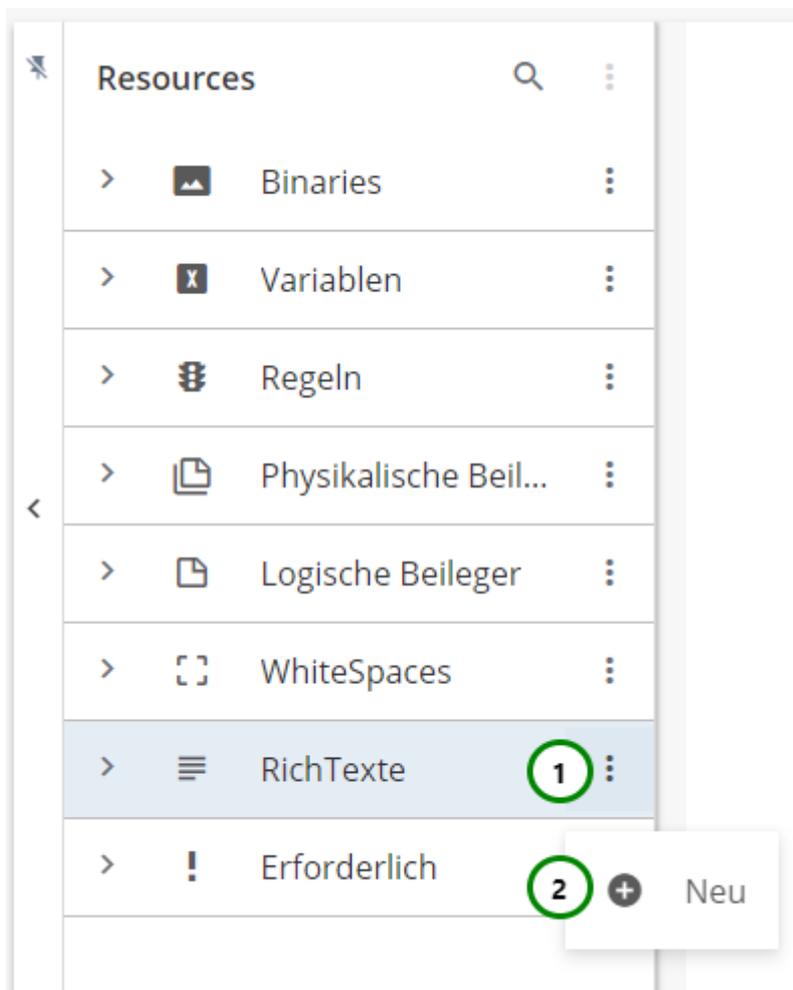


Abbildung A

In dem Ressourcen-Typ  RichText finden Sie alle bereits angelegten RichTexte.

Klappen Sie dazu den Ressourcen-Typ auf .

RichTexte werden als komplexe Elemente betrachtet und bestehen daher aus einer Eigenschaften- und Designseite.

Um einen neuen RichText zu erstellen, klicken Sie bitte auf das  Kontextmenü [Abb. A \(1\)](#)

des Ressourcen-Typs. Wählen Sie nun  Neu (2). Sie landen zuerst in den Eigenschaften des RichTextes.

Um die Eigenschaften zu ändern, wählen Sie die Ressource, die Sie ändern möchten, aus der Übersichtsliste aus [Abb. B \(1\)](#) und öffnen Sie das  Kontextmenü (1). Nach einem weiteren Klick auf  Eigenschaften (2) öffnet sich der folgende Inhalt:

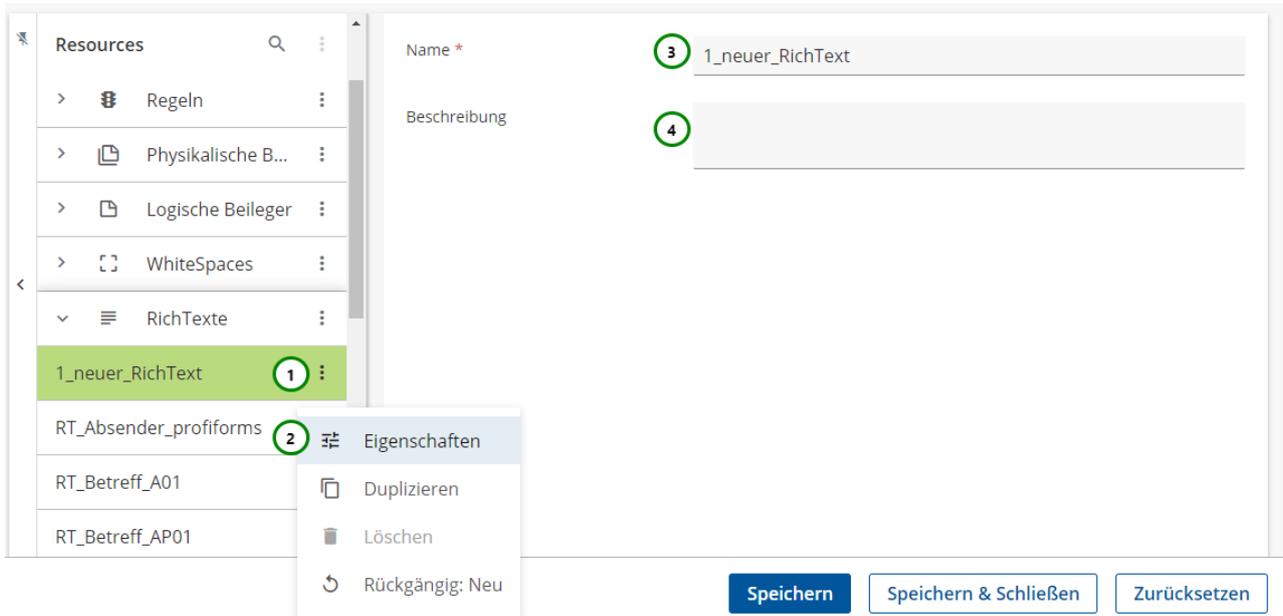


Abbildung B

**Sie können dem RichText:**

- ★ einen **Namen** geben *Abb. B (3)*.  
 Sie müssen dem RichText einen Namen geben, unter dem Sie den Beileger dann ansprechen und nutzen.  
**Beachten Sie:** In einer Schema-Version/Bibliotheks-Version dürfen keine 2 Ressourcen denselben Namen haben. Hat eine Ressource aus einer Schema-Version denselben Namen wie eine Ressource aus einer eingebundenen Bibliotheks-Version, überschreibt die Schema-Ressource die Bibliotheks-Ressource.
- ★ eine **Beschreibung** geben *(4)*. Beim Erstellen und Ändern müssen Sie das Feld nicht zwingend ausfüllen.  
 Die Beschreibung einer Ressource wird z.B. in Auswahl-Fenstern angezeigt.

**RichText Inhalt zuordnen**

Um einen Rich-Text Inhalt zuzuordnen, müssen Sie in die Designansicht wechseln. Dazu genügt ein Klick auf die eben erstellte Ressource *Abb. B (1)* in der Ressourcen-Übersicht. Es öffnet sich folgende Designansicht:

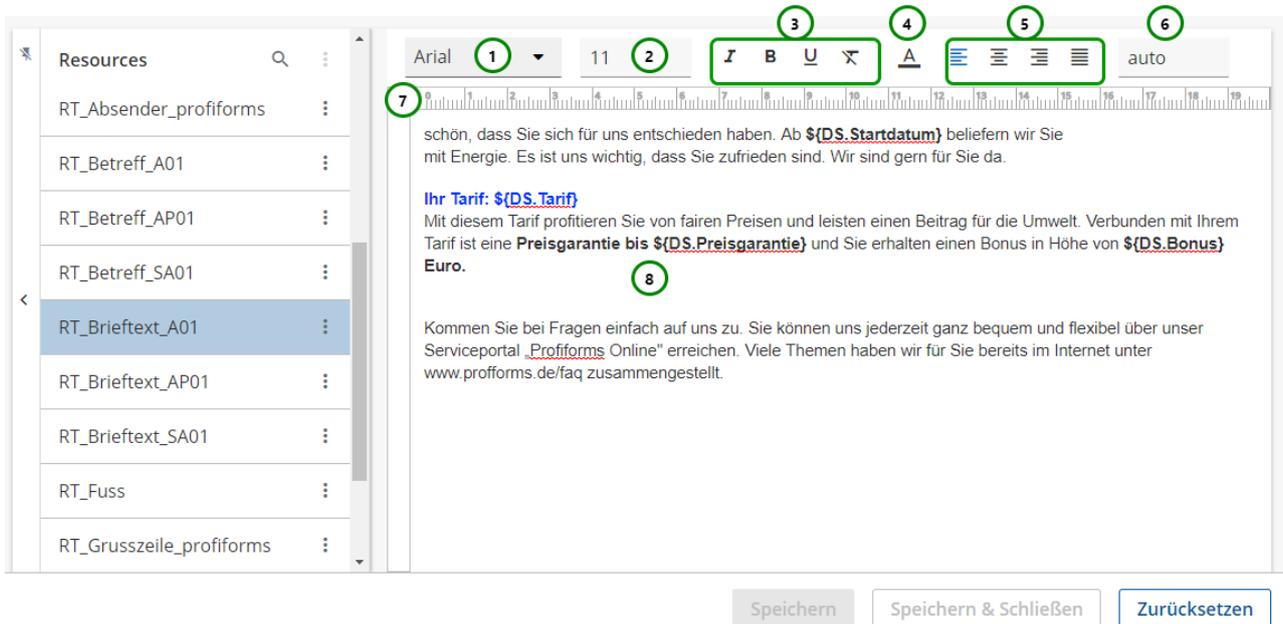


Abbildung C

Sie können nun den Rich-Text bearbeiten Abb. C (8). Dazu stehen Ihnen zusätzlich diverse Formatierungshilfen zur Verfügung Abb. C (1 bis 7).

**Sie können**

- eine **Schriftart** auswählen. Klicken Sie dafür bitte auf Abb. C (1). Es klappt ein Auswahlménü auf. Beachten Sie, dass Sie nur die Schriftarten auswählen können, die vom Transaktions-Formular erlaubt werden. Einsehen können Sie dies in der **Registerkarte Schriftarten**.
- eine **Schriftgröße** geben. Klicken Sie dafür bitte auf (2). Tragen Sie die gewünschte Schriftgröße ein oder wählen Sie aus dem vorgegebenen Auswahlménü aus. Es werden auch Angaben mit "." unterstützt z.B. 10.5.
- eine **Textformatierung** geben (3).  
 Sie können zwischen **B** **fettem**, *I* **kursivem** und U **unterstrichenem** Text wählen. **Beachten Sie**, dass Sie nur die Formatierungen auf eine Schriftart anwenden können, die im Transaktions-Formular als erlaubt definiert wurden.  
 Sie können die Formatierung mit  wieder entfernen.
- eine **A** **Schriftfarbe** geben. Klicken Sie dafür bitte auf (4). Es klappt ein Auswahlménü auf:

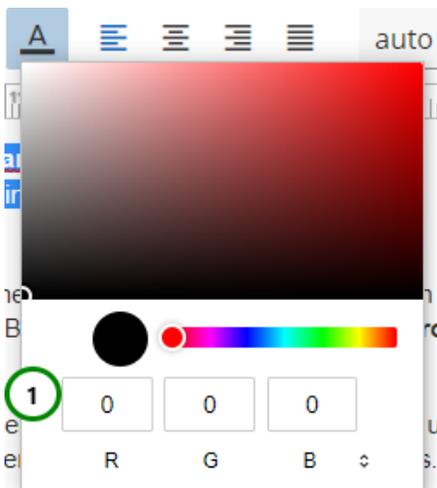


Abbildung D

Wählen Sie eine Farbe aus dem Fenster oder geben Sie die RGB-Zahlen in das Eingabefeld Abb. D (1) ein.

- eine **Textausrichtung** geben (5). Sie können zwischen  linksbündigem,  zentriertem oder  rechtsbündigem Text wählen.

- Den **Zeilenumbruch** (6) der Designansicht einstellen. Das hat keine Auswirkung auf den späteren Druck.

Damit können Sie die maximale Breite des Inhaltsbereiches auf dem späteren Druck simulieren, um automatische Zeilenumbrüche besser zu erkennen.

**Beachten Sie**, dass diese Änderung keine Auswirkung auf den späteren Druck hat.

- **Tabstops** setzen (7) . Klicken Sie dazu an der gewünschten Position auf das Lineal.
- **Font** ist im Transaktions-Formular nicht erlaubt bzw. auf dem Server nicht vorhanden: Der Text wird **rot** hinterlegt.

Sie können hier die **Eingabe-Hilfe** verwenden

 Beachten Sie, dass die Umbenennung einer Ressource alle Verweise auf die Ressource ungültig macht, insbesondere in Erweiterungen. Wird die Ressource in einer Erweiterung oder in einer anderen Ressource verwendet, müssen Sie nach dem Umbenennen alle Verweise anpassen.

## Erforderliche Ressource erstellen und ändern

⚠ Dieser Artikel bezieht sich nur auf Bibliotheks-Versionen.

Erforderliche Ressourcen dienen dazu, Ressourcen einer Formular-/Schema-Version in einer Bibliotheks-Version bekannt zu machen, um dann damit innerhalb der Bibliotheks-Version arbeiten zu können, obwohl die eigentliche Ressource erst verfügbar ist, wenn die Bibliothek eingebunden wird. Falls Sie sich noch nicht mit erforderlichen Ressource vertraut gemacht haben, eine Beschreibung finden Sie [hier](#).

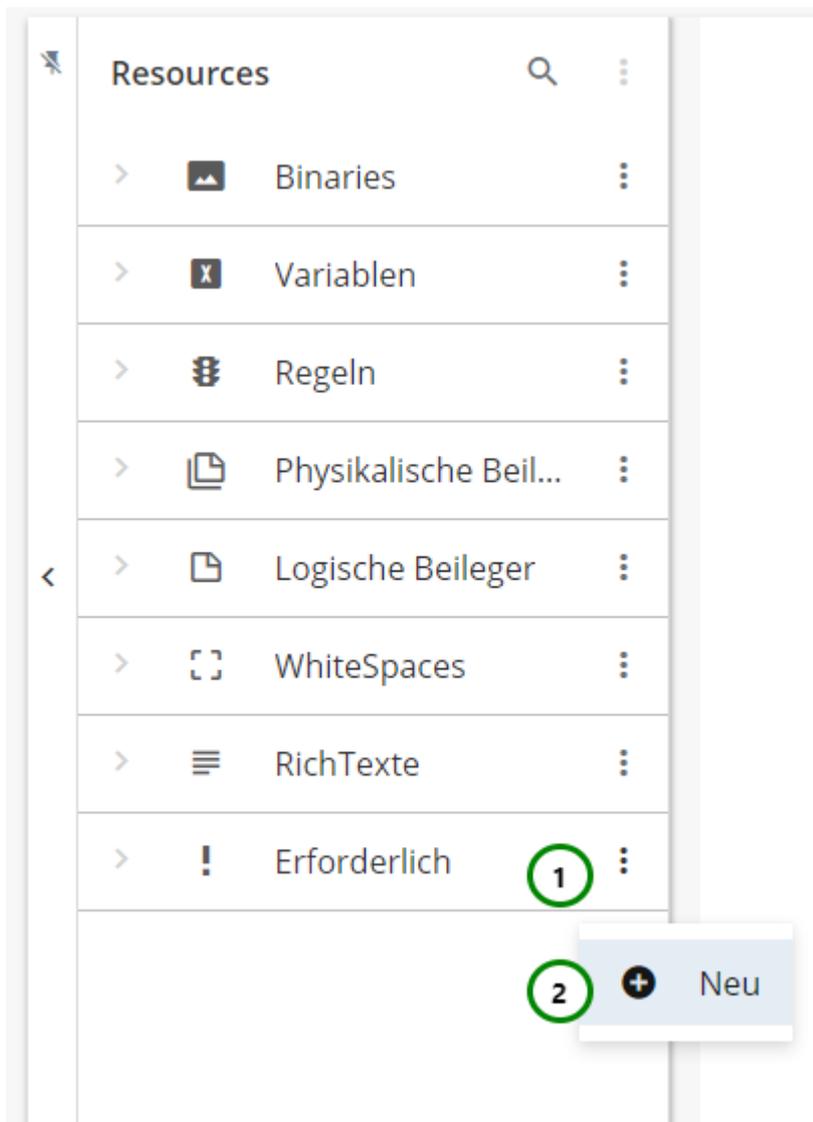


Abbildung A

In dem Ressourcen-Typ **!** Erforderlich finden Sie alle bereits angelegt erforderlichen Ressourcen. Klappen Sie dazu den Ressourcen-Typ auf **>** .

Um eine neue erforderlichen Ressource zu erstellen, klicken Sie bitte auf das **⋮** Kontextmenü Abb. A (1) des Ressourcen-Typs und wählen Sie nun **+** Neu (2) aus.

Um Änderungen vorzunehmen, wählen Sie die Ressource, die Sie ändern möchten, aus der Übersichtsliste aus Abb. B (1). Nun sehen Sie in dem Inhaltsbereich die Informationen zu dieser Ressource:

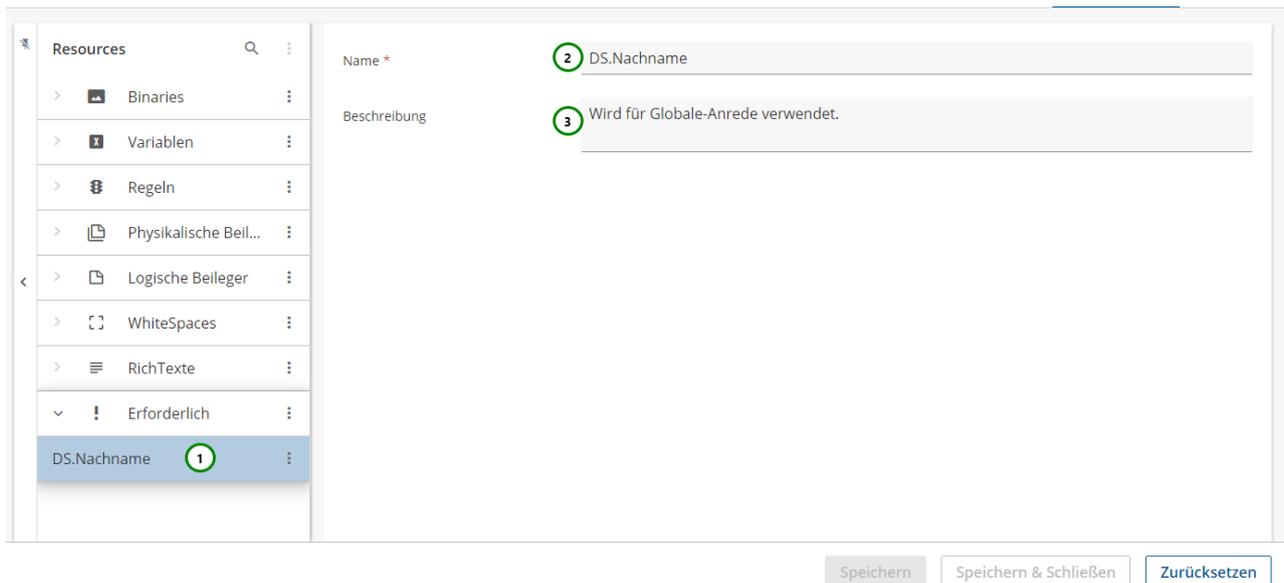


Abbildung B:

Die Erforderliche Ressource muss den gleichen Namen haben wie die Ressource aus der Schema-Version, die Sie verwenden möchten Abb. B (2). Optional können Sie der erforderlichen Ressource auch eine Beschreibung geben (3). Sie wird z.B. in Auswahl-Fenstern angezeigt.

Nachdem Sie die erforderliche Ressource erstellt haben, können Sie sie in der Bibliothek nutzen. Dazu sollten Sie aber im vornherein wissen, welchen Ressource-Typ die Ressource später haben wird, da es sonst zu unerwünschtem Verhalten kommen kann. In anderen Bibliotheks-Ressourcen nutzen Sie die erforderliche Ressource jetzt ganz normal.

Name der erforderlichen Ressource: DS.Nachname

Name \*

Variable

Beschreibung

Wert

Sehr geehrte Frau \${DS.Nachname}

Der Wert der erforderlichen Ressource wird später, beim Einbinden in das Schema, durch die dort vorhandene Ressource aufgefüllt. Existiert die Ressource im Schema nicht, kommt es zu einer Fehlermeldung.

## Ressourcen löschen

In der Ressourcenverwaltung steht Ihnen folgende Aktion zur Verfügung, mit der Sie Ressourcen löschen können.

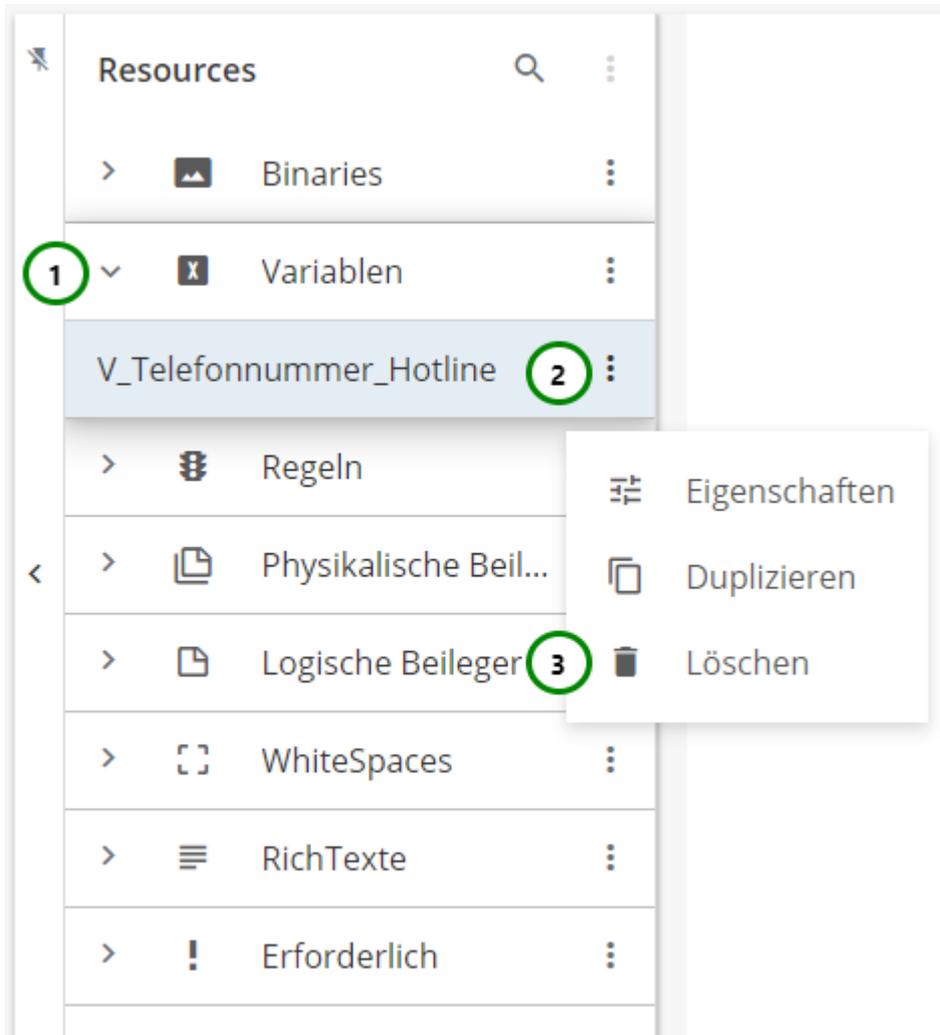


Abbildung A

Klappen Sie einen Ressourcen-Typ auf [Abb. A \(1\)](#). Wählen Sie die Ressource aus, die Sie löschen möchten, und öffnen Sie das  Kontextmenü (2). Klicken Sie anschließend auf  löschen (3), um die Ressource zum löschen vorzumerken.

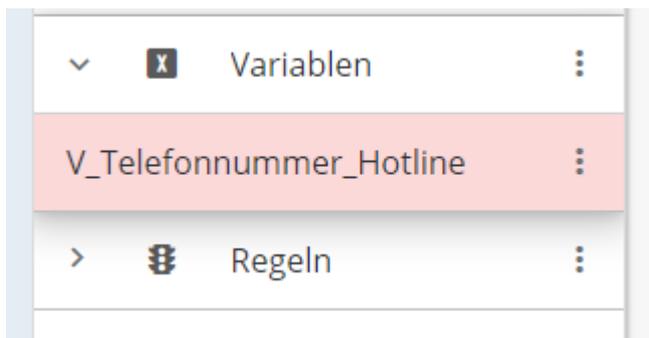


Abbildung B

Die Ressource erscheint jetzt rot in der Übersichtstabelle [Abb. B](#). Mehr dazu finden Sie [hier](#). Die Ressource ist nun zum löschen vorgemerkt und wird erst endgültig gelöscht, wenn Sie die Schema- bzw. Bibliotheksversion speichern. Wird Schema- bzw. Bibliotheksversion verworfen, so bleibt die Ressource bestehen.

Sie können das Löschen einzelner Ressourcen auch rückgängig machen. Mehr dazu finden [hier](#).

**i** Sie können auch mehrere Ressourcen markieren und löschen, indem Sie <STRG> gedrückt halten.

Abbildung C

Öffnen Sie das Kontextmenü [Abb. C \(1\)](#) der Selektion (Kopf der Übersicht) und klicken Sie anschließend auf Selektion löschen (2) .

**!** Wenn Sie eine Ressource löschen, die durch eine Erweiterung in das Dokument eingebunden wird, wird der Verweis in der Erweiterung ungültig. Ein ungültiger Verweis wird gekennzeichnet. Weitere Informationen dazu finden Sie unter [Konflikte und Markierungen bei Erweiterungen](#).  
**Beachten Sie außerdem:** Wird eine Schema-Ressource, die eine Bibliotheks-Ressource überschreibt, gelöscht, verwenden Erweiterungen nach dem Löschen die **überschriebene** Ressource. Dies kann zu unerwünschtem Verhalten führen, da die überschriebene Ressource durch die Erweiterung in das Dokument eingebunden wird, ohne einen Fehler zu generieren, da ja eine Ressource mit dem geforderten Namen vorhanden ist.

## Ressourcen duplizieren

In der Ressourcenverwaltung steht Ihnen folgende Aktion zur Verfügung, mit der Sie Ressourcen duplizieren können.

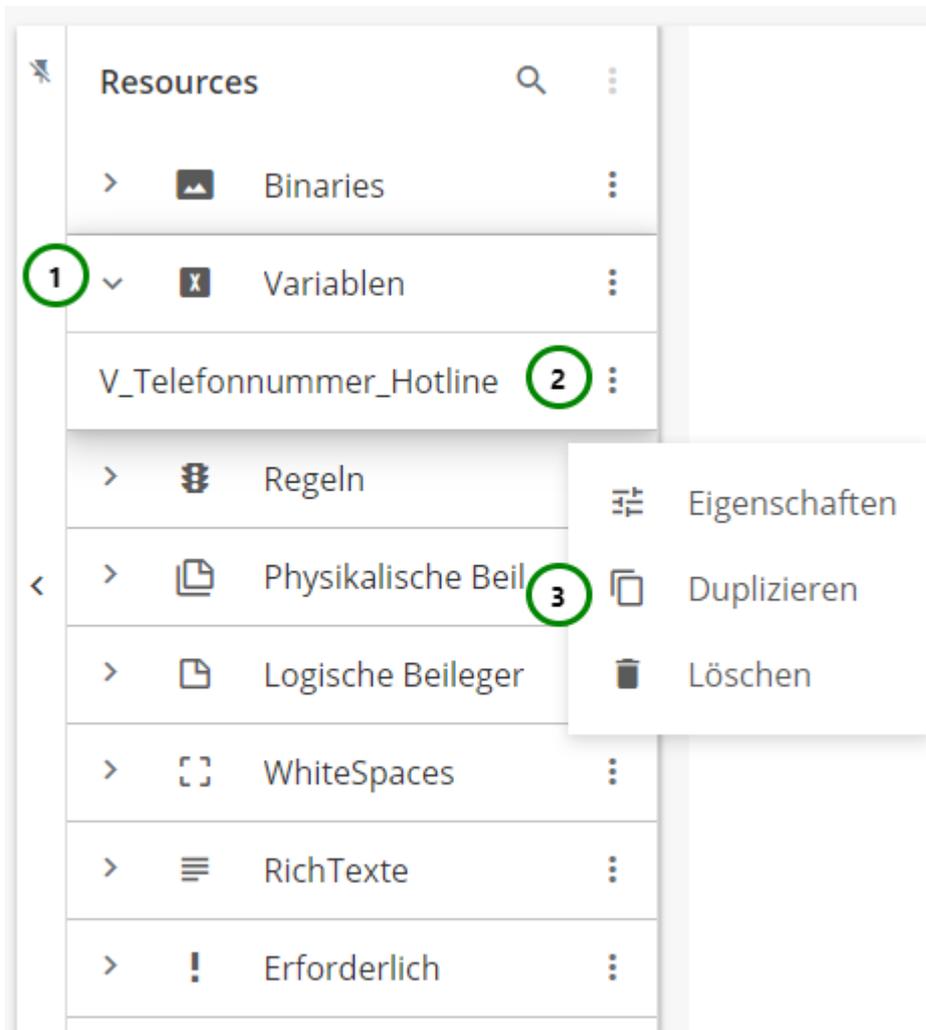


Abbildung A

Klappen Sie hierzu einen Ressourcen-Typ auf  Abb. A (1). Wählen Sie die Ressource aus, die Sie duplizieren möchten und öffnen Sie das  Kontextmenü (2). Anschließend klicken Sie auf  Duplizieren (3).

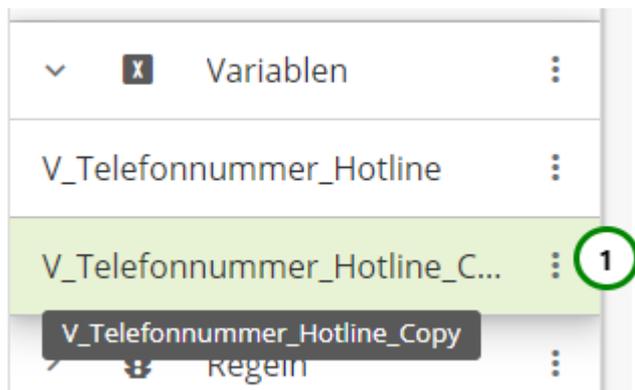


Abbildung B

Die ausgewählte Ressource ist nun dupliziert [Abb. B \(1\)](#) und hat den Namenszusatz "\_Copy" bekommen. Der Typ wurde übernommen. Die duplizierte Ressource übernimmt alle Inhalte und Eigenschaften des Originals. Sie wird nun in grün dargestellt. Mehr dazu finden Sie [hier](#).  
Klicken Sie auf Speichern, um die Schema- bzw. Bibliotheks-Version zu speichern oder Verwerfen Sie alle Änderungen mit Zurücksetzen.

Sie können das Duplizieren einer Ressourcen auch rückgängig machen. Mehr dazu finden [hier](#).

**i** Sie können auch mehrere Ressourcen markieren und duplizieren, indem Sie <STRG> gedrückt halten.

**2 Ressourcen gewählt** **1** :

- 2** Selektion duplizieren
- Selektion löschen
- Selektion zurücksetzen

*Abbildung C*  
Öffnen Sie das **:** Kontextmenü [Abb. C \(1\)](#) der Selektion (Kopf der Übersicht) und klicken Sie anschließend auf **Selektion duplizieren (2)**.

## Ressourcen zurücksetzen

Sie können einzelne Ressourcen während der Bearbeitung zurücksetzen, ohne andere Ressourcen zu beeinflussen.

Ressourcen, die seit dem letzten Speichern modifiziert wurden, werden farblich markiert. Diese Ressourcen können in den Zustand des letzten Speicherns zurückgesetzt werden.

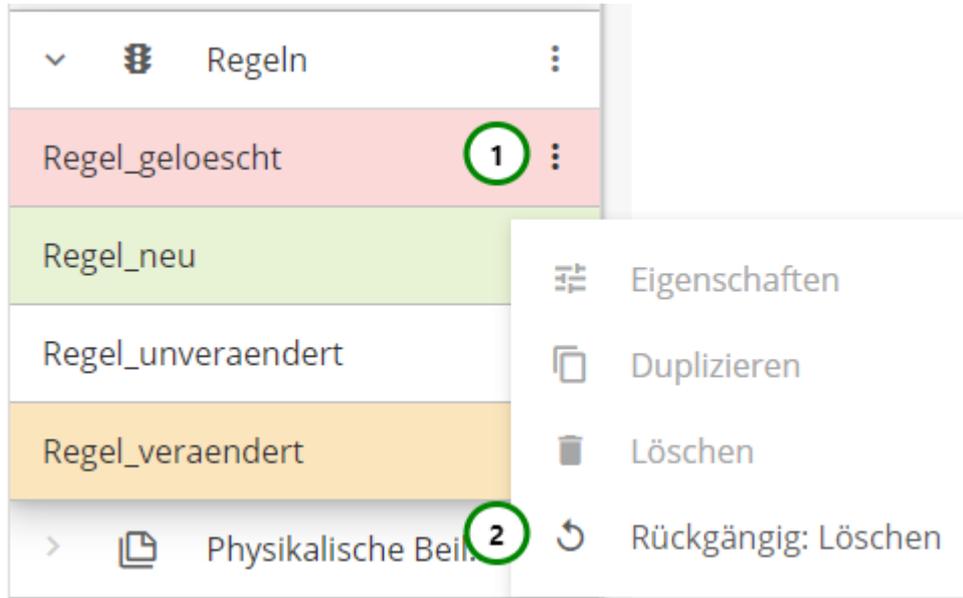


Abbildung A

Um eine einzelne Ressource zurückzusetzen, öffnen Sie das  Kontextmenü [Abb. A \(1\)](#) der Ressource und klicken Sie auf  Rückgängig: ... (2).

Die Ressource wurde erfolgreich zurückgesetzt, wenn die farbliche Markierung aufgehoben wurde [Abb. B](#):

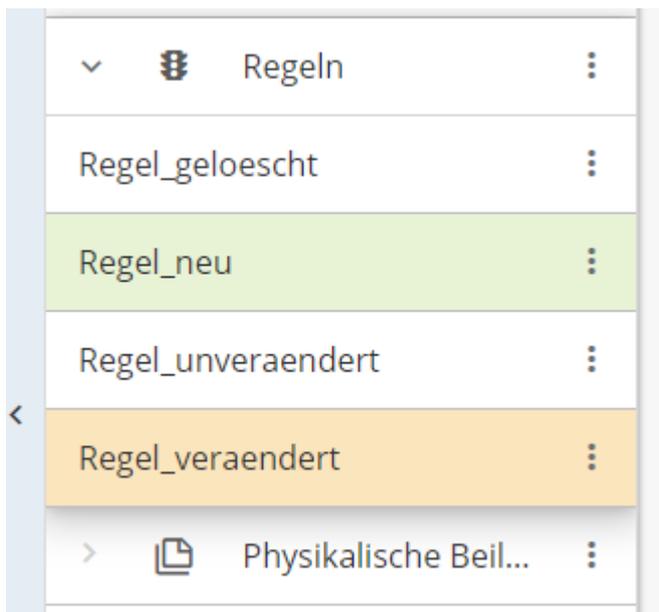


Abbildung B

⚠️ **Besonderheit:** Haben Sie eine vorhandene Ressource erst verändert (gelb) und anschließend gelöscht (rot), so kehren Sie beim ersten Zurücksetzen zur veränderten Ressource zurück.

## Ressourcen exportieren

In der Ressourcenverwaltung steht Ihnen folgende Aktion zur Verfügung, mit der Sie Ressourcen exportieren können.

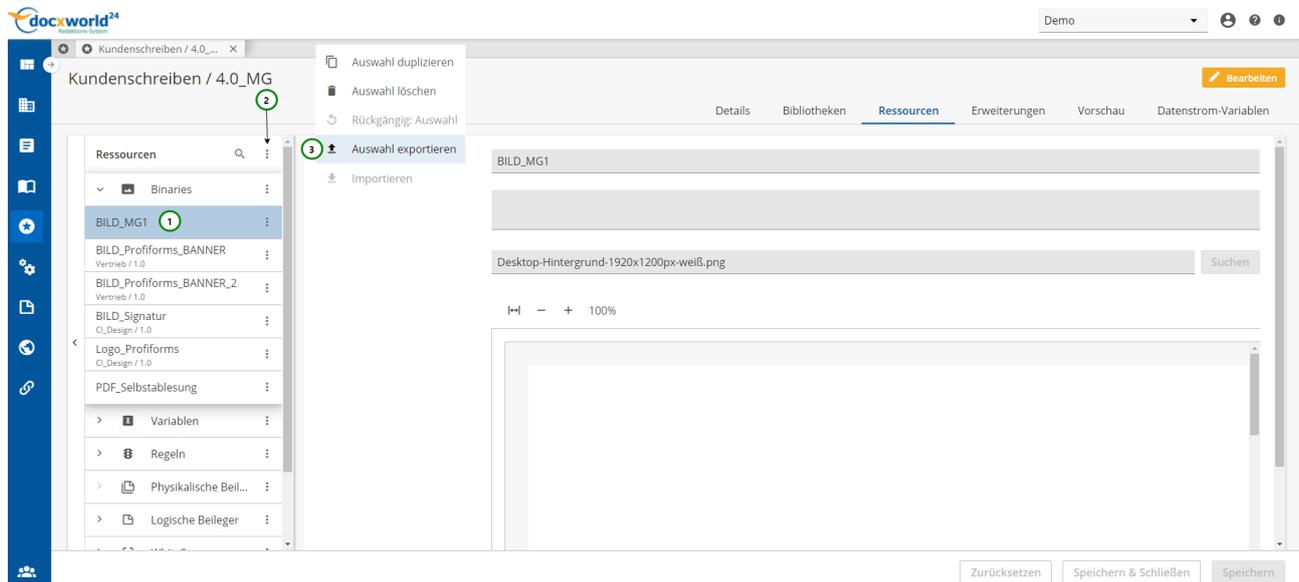
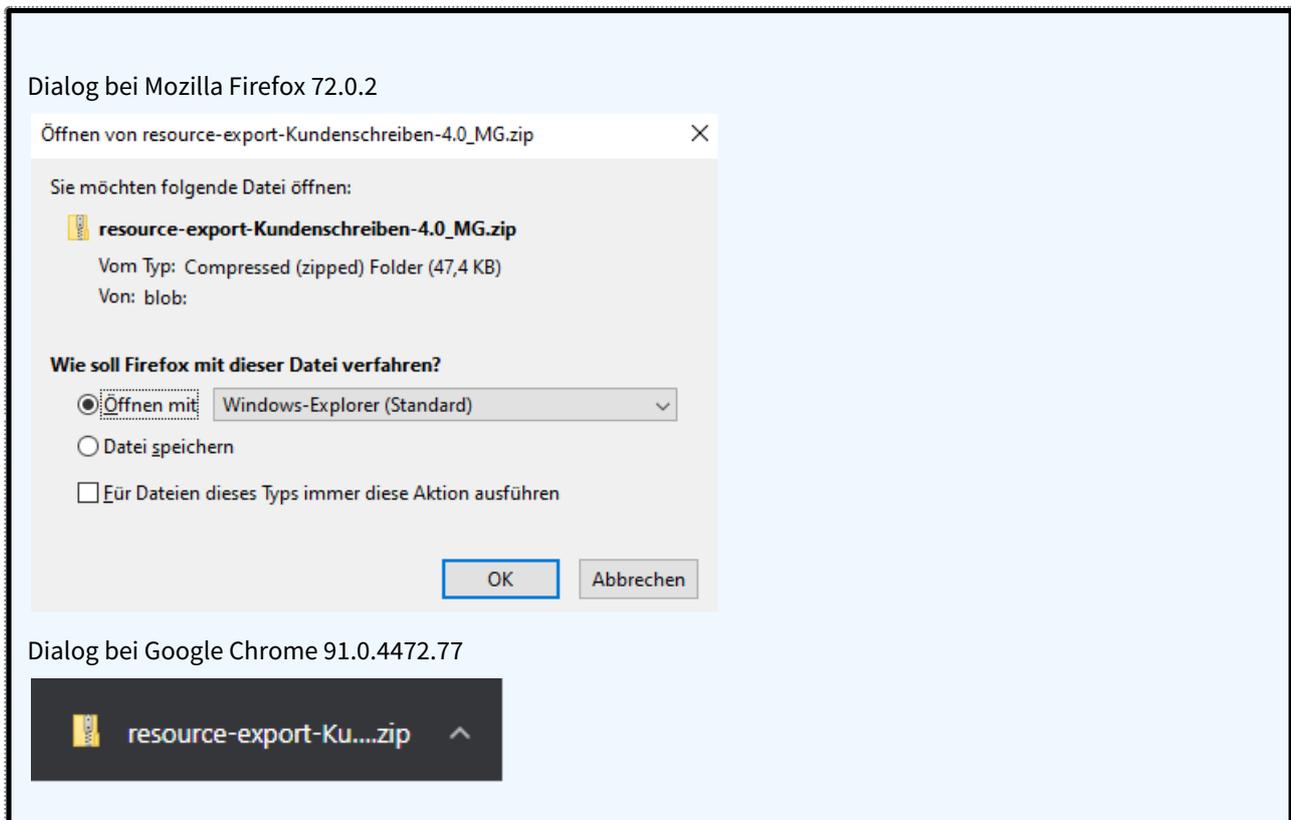


Abbildung A

Sie können Ressourcen exportieren, um Sie außerhalb einer Schema-Version/Bibliotheks-Version verfügbar zu machen. Sie können die Ressource z.B. in einem anderen Arbeitsbereich wieder **importieren**, um damit weiterzuarbeiten. Wählen Sie dazu aus Abb. A (1) eine Ressource aus, die Sie exportieren möchten. Öffnen Sie anschließend das  Kontextmenü (2) und wählen  Auswahl exportieren (3) aus.

 Sie können auch mehrere Ressourcen markieren, indem Sie <strg> + <alt> gedrückt halten.

Abhängig von dem Browser, den Sie verwenden, werden Ihnen eventuell unterschiedliche Dialoge angezeigt:



- Ressourcen, die in der Bibliothek definiert werden, können nicht im Schema exportiert werden.
- **Erforderliche Ressourcen** können nicht exportiert werden, da es sich dabei nicht um Ressourcen im eigentlichen Sinne handelt.
- **Datenstrom-Variablen** können nicht exportiert werden.

Die Ressource wird (nach Bestätigung) automatisch in Ihr Standard Download-Verzeichnis exportiert (*Unter Windows: User-Verzeichnis/Downloads*). Sie finden die Datei, indem Sie zum Download-Verzeichnis wechseln oder sich den Download in Ihrem Browser anzeigen lassen. Die Datei wird standardmäßig unter dem Namen "export-ressourcen" abgespeichert. Wenn Sie mehrere Ressourcen exportieren, werden diese zusammen in einer Datei abgespeichert. Beim Import der Datei werden alle Ressourcen importiert.



Sie können auch komplette Schema-Versionen / Bibliotheks-Versionen exportieren. Klicken Sie für weitere Informationen darüber bitte [hier](#).

## Ressourcen importieren

In der Ressourcenverwaltung steht Ihnen folgende Aktion zur Verfügung, mit der Sie Ressourcen importieren können.

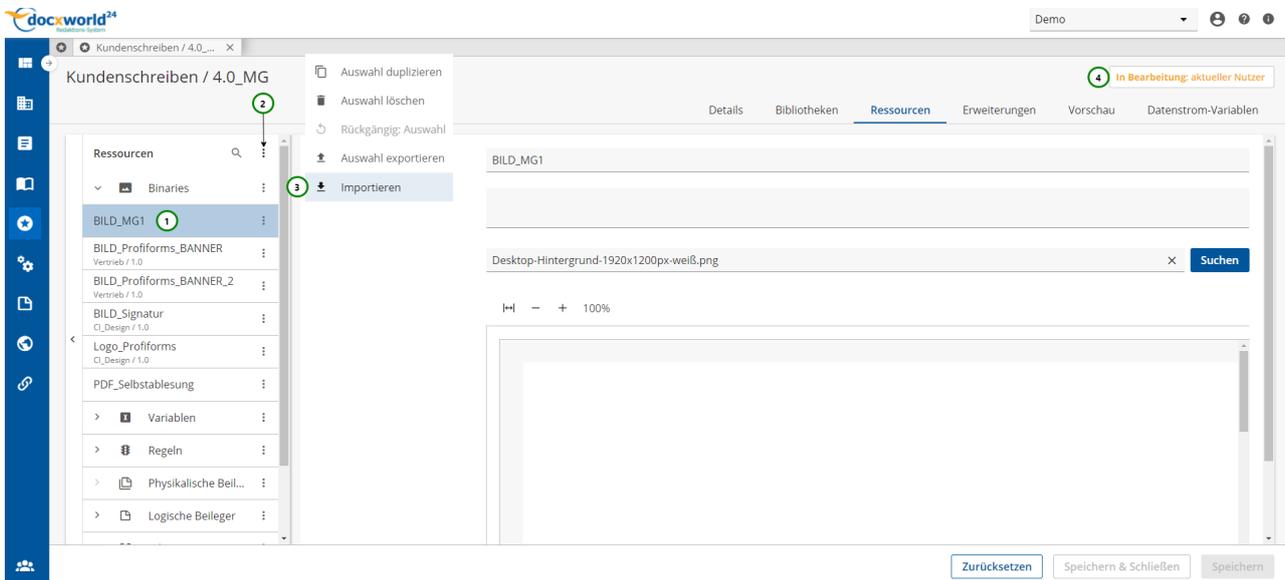


Abbildung A

Sie können Ressourcen importieren, um Sie innerhalb einer Schema-Version/Bibliotheks-Version verfügbar zu machen. Sie können die Ressource auch wieder **exportieren**. Öffnen Sie anschließend das

- Kontextmenü (2) und wählen Auswahl  Importieren (3) aus.

 Der Ressourcen-Import funktioniert nur im Bearbeitungsmodus Abb. A (4).

Folgender Dialog öffnet sich:

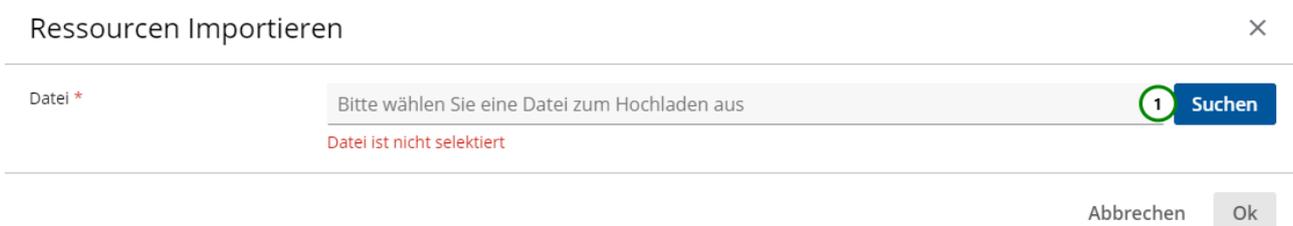


Abbildung B

Klicken Sie auf Abb. B (1) und wählen Sie dann die gewünschte Datei aus. Handelt es sich um eine importfähige Datei, schließt sich der Dialog automatisch und die importierten Ressourcen werden Ihnen in der Ressourcenliste angezeigt. Kann die Datei nicht importiert werden, weil es sich um keine gültige R-S-Import-Datei handelt, bricht das R-S mit einer Fehlermeldung ab und Sie müssen eine andere Datei auswählen.

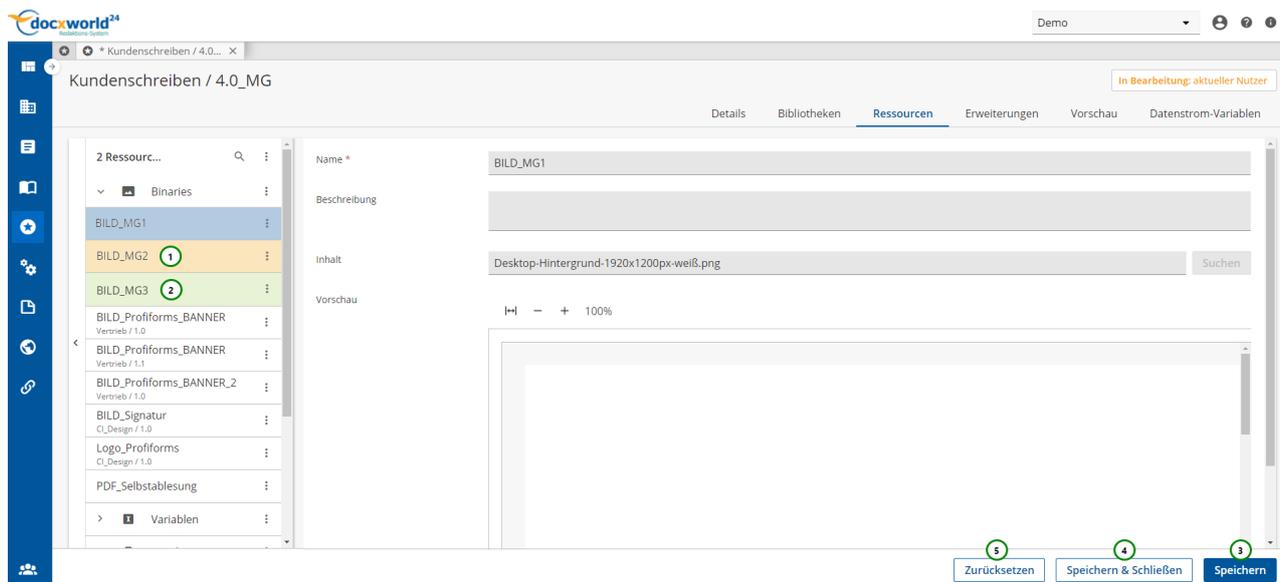


Abbildung C

**Orange markierte Zeilen** : Die Ressource war bereits in der Schema-Version vorhanden und wird von der importierten Ressource überschrieben Abb. C (1).

**Grün markierte Zeilen** : Die Ressource wurde erfolgreich importiert (2).

Klicken Sie zum Speichern nun auf Abb. C (3) oder auf (4). Die Ressourcen sind nun nicht mehr markiert und der Import wurde abgeschlossen.

Um den Import abzubrechen, importierten Dateien zu entfernen sowie die überschriebenen Dateien wiederherzustellen, klicken Sie bitte auf Abb. C (5). Beachten Sie dass, falls importierte Ressourcen Abhängigkeiten zu anderen Ressourcen besitzen, diese ebenfalls in der Schema-Version vorhanden sein müssen.

**i** Sie können auch ganze Schema-Versionen / Bibliotheks-Versionen importieren. Klicken Sie für weitere Informationen bitte [hier](#).

**i** Sie können auch einzelne Ressourcen zurücksetzen. Klicken Sie für weitere Informationen bitte [hier](#).

## Ressourcen Zustände

Ressourcen können 4 verschiedene Zustände während der Bearbeitung durchlaufen. Diese Zustände werden farblich hervorgehoben.

Regeln	
Regel_geloescht	
Regel_neu	
Regel_unveraendert	
Regel_veraendert	

### Weiß

- Die Ressource hat sich seit dem letzten Speichern der Schema- bzw. Bibliotheksversion **nicht verändert**.

### Grün

- Die Ressource ist seit dem letzten Speichern der Schema- bzw. Bibliotheksversion **neu hinzugekommen**.

### Rot

- Die Ressource wurde zum löschen vorgemerkt und wird beim nächsten Speichern endgültig **gelöscht**.

### Gelb

- Die Ressource hat sich seit dem letzten Speichern der Schema- bzw. Bibliotheksversion **verändert**.

⚠️ Erst nach dem Speichern werden alle Änderungen an den Server übergeben und endgültig gespeichert. Danach wechseln alle Zustände auf "weiß" - nicht verändert, da keine Änderungen seit dem letzten Speichern mehr vorliegen.

## Arbeiten mit Produktions-Paketen

In diesem Kapitel wird das Arbeiten mit Produktions-Paketen behandelt. Um zu den Übersichtstabellen der Produktions-Pakete zu gelangen, wechseln Sie in den Navigationspunkt Produktion.

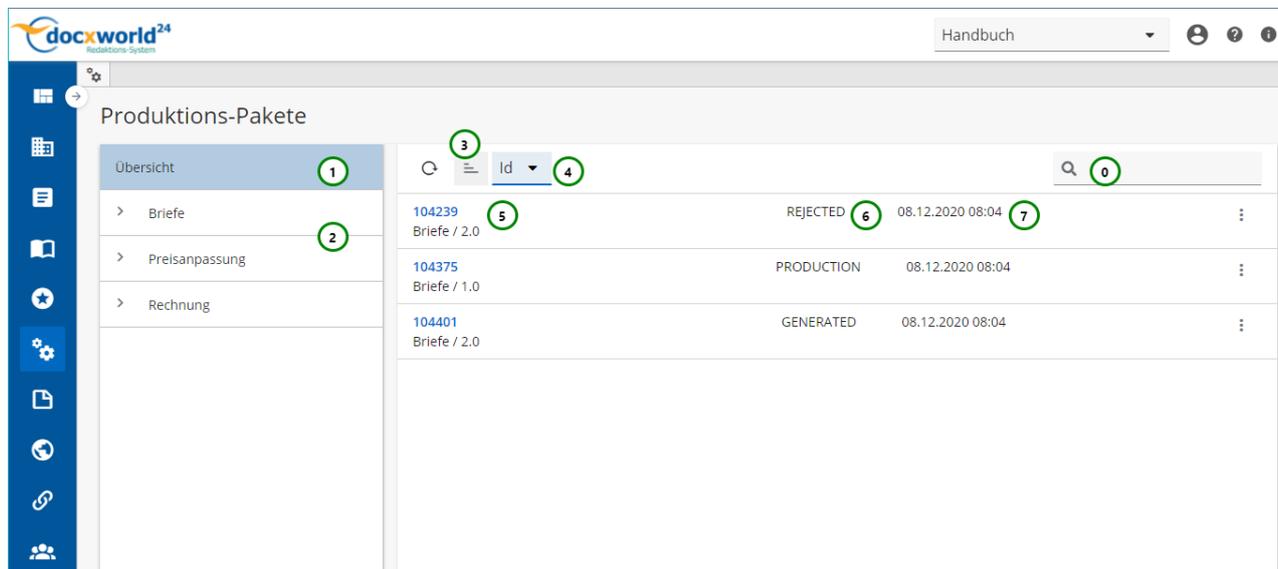


Abbildung A

Produktions-Pakete bündeln Formular und Schema in einem produktionsbereiten Paket, das im Dokument-Erstellungs-Prozess genutzt werden kann. Produktions-Pakete werden automatisch erstellt, wenn eine **Schema-Version** den Status **TEST** erreicht. Ein Paket ist für externe Produktionskomponenten erst verfügbar, wenn es PRODUCTION-Status hat und eine Verknüpfung auf das Paket besteht (**Docxworld-Link** oder **Produktions-Paket Verknüpfung**). Lesen Sie bitte den **Abschnitt über Produktions-Pakete**, bevor Sie beginnen, damit zu arbeiten.

Sie sehen in Abb. A:

- Die **Suchfunktion**, um nach Begriffen zu suchen (0).
- Die **Übersicht** aller vorhandenen Produktions-Pakete (1).
- Die **Übersicht** der vorhandenen Produktions-Pakete eines bestimmten **Schemas** (2).
- Die Liste **auf- / absteigend** sortieren (3) gemäß des Sortierkriteriums (4).
- Die **ID** des Produktions-Paketes inklusive der Herkunft (5).
- Den aktuellen **Status** des Produktions-Paketes (6).
- Das letzte **Änderungsdatum** des Produktions-Paketes (7).

## Paket ändern

In der Produktions-Paket-Übersichtstabelle steht Ihnen folgende Aktionen zur Verfügung, um ein Produktions-Paket zu ändern:

Id	Status	Datum	Aktionen
104239 Briefe / 2.0	REJECTED	08.12.2020 08:04	⋮
104375 Briefe / 1.0	PRODUCTION	08.12.2020 08:04	⋮

Abbildung A

Um ein Produktions-Paket zu **ändern**, wählen Sie bitte das Paket, das Sie ändern möchten und klicken Sie anschließend auf Abb. A (1). Es öffnet sich ein neuer Tab mit folgenden Registerkarten:

### Kurzbeschreibung der Registerkarten und ihrer Inhalte:

<b>Detail</b>	Sie beinhaltet allgemeine Informationen über die Schema-Version.
<b>Historie</b>	Zeigt aufgetretene Ereignisse, die das Objekt betreffen (z.B. Statusänderungen).
<b>Vorschau</b>	Mit Hilfe von Testdaten können Sie hier eine Dokument Vorschau generieren lassen.
<b>Test-Zyklus</b>	Beinhaltet eventuelle Anmerkungen, die während der Test-Phase des Objekts entstanden sind.
<b>Produktions-Zyklus</b>	Beinhaltet eventuelle Anmerkungen, die während der Produktions-Phase des Objekts entstanden sind.
<b>Paket-Prüfung</b>	Enthält Meldungen (Warnungen, Fehler), die bei der Prüfung des Pakets aufgetreten sind.

## Registerkarte - Details in Produktions-Pakete

Zeigt Ihnen Details über das ausgewählte Produktions-Paket an.

Briefe / 2.0 1

Details Historie Vorschau Test-Zyklus Produktions-Zyklus Paket-Prüfung

Status 2 REJECTED

Beschreibung 3 04.12.2020 - Anpassungen der Texte

Erstellt / Geändert 08.12.2020 07:31 4 04.01.2021 14:38 5

Speichern Speichern & Schließen Zurücksetzen

Abbildung A

**Sie sehen Informationen über das Produktions-Paket :**

- Durch welche **Schema-Version** welchen **Schemas** das Produktions-Paket generiert wurde Abb. A (1).
- den **Status** des Pakets (2).
- Wann das Paket **erstellt** (3) und wann zum letzten Mal **geändert** (4) wurde.
- i Sie können dem Produktions-Paket außerdem eine Beschreibung geben (3).

## Registerkarte - Historie

Zeigt Ihnen die durchgeführten Aktionen an dem Produktions-Paket in chronologischer Reihenfolge an.

Briefe / 2.0

Details Historie Vorschau Test-Zyklus Produktions-Zyklus Paket-Prüfung

<input type="text" value=""/>					
04.01.2021 14:38	57005:joey	A	UPDATE	bundle updated	
08.12.2020 08:04	57005:Joey		TEST -> REJECTED		
08.12.2020 08:04	57005:Joey		GENERATED -> TEST		
08.12.2020 07:31	SYSTEM		GENERATING -> GENERATED	generation complete	
08.12.2020 07:31	SYSTEM		SCHEDULED -> GENERATING		
08.12.2020 07:31	SYSTEM	B	NEW -> SCHEDULED		
08.12.2020 07:31	57005:Joey		CREATE	created by schema-revision[104165] call	

Speichern Speichern & Schließen Zurücksetzen

Abbildung A

In der Registerkarte Historien können Sie verfolgen, wann Änderungen am Paket stattgefunden haben. Solche Änderungen sind insbesondere Statuswechsel des Pakets oder Änderungen am Paket durch den Nutzer. Die Aktionen werden Ihnen in Abb. A (3) aufgelistet:

**z.B. bei Statuswechseln (B):**

ALT -> NEU

**z.B. Änderungen am Paket (A):**

UPDATE

Änderungen am Paket können z.B. Einträge im Test- oder Produktions-Zyklus sein.

**Sie sehen in der Historie:**

- Das **Datum** der Aktion inklusive der **Uhrzeit** Abb. A (1),
- welcher **Nutzer** die Aktion ausgeführt hat (z.B. SYSTEM für das R-S oder den Nutzernamen des Nutzers) (2),
- welche **Aktion** durchgeführt wurde (3) und
- eine **Beschreibung** der Aktion (4).

## Registerkarte - Vorschau in Produktions-Pakete

Zeigt Ihnen alle verfügbaren Vorschauen für das Paket an.

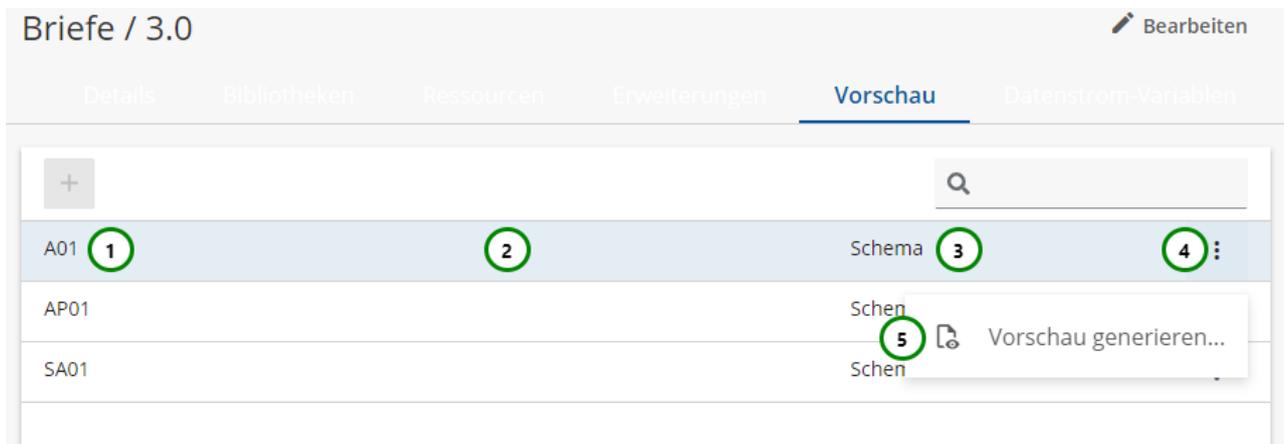


Abbildung A

### Sie sehen:

- Den **Formularnamen**, wie er in der Konfiguration der Test-Vorschau angegeben ist Abb. A (1),
- falls vorhanden die **Beschreibung** der Vorschau (2) und
- die **Herkunft** der Vorschau (3).
  - Vorschauen aus der Transaktions-Formular-Version sind feste, durch den Formular-Designer eingebrachte Testvorschauen.
  - Vorschauen aus der Schema-Version sind Testvorschauen, die Sie selbst erstellt haben. Wie Sie Testvorschauen selbst erstellen, lernen Sie [hier](#).

Wählen Sie eine Vorschau aus. Klicken Sie dazu auf das Kontextmenü Abb. A (4) und anschließend auf Vorschau generieren (4), um die Vorschau generieren zu lassen. Folgender Dialog öffnet sich:

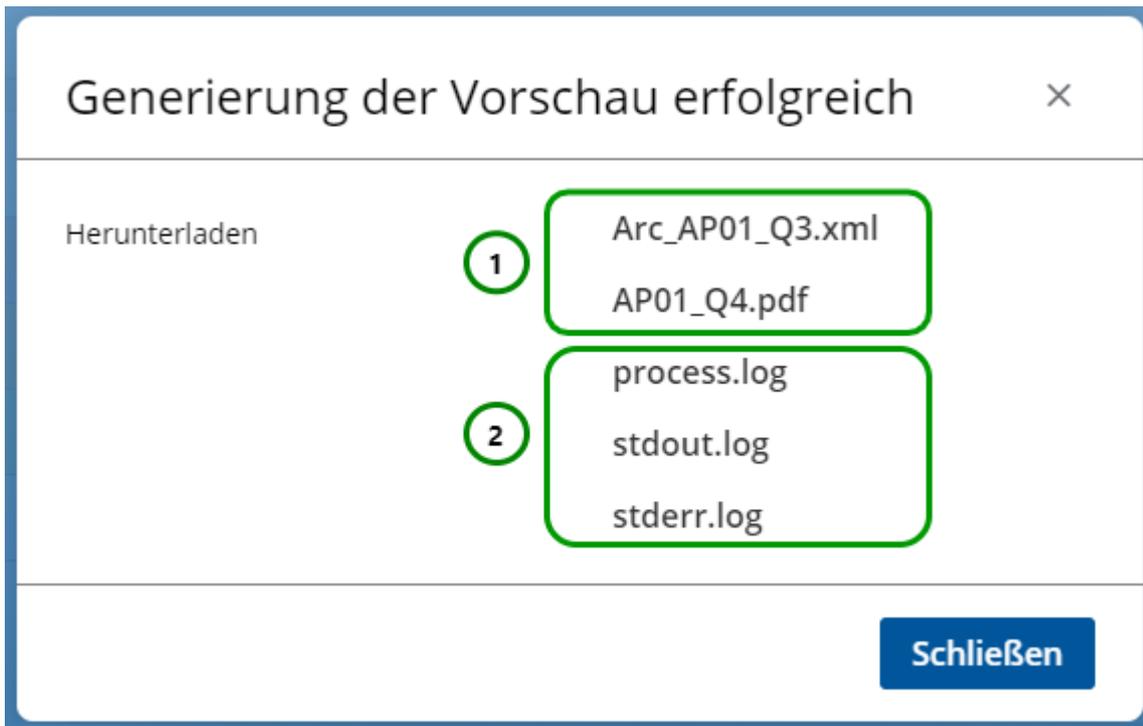


Abbildung B

Wenn beim Generierungsprozess keine Fehler aufgetreten sind, können Sie sich die Vorschau nun anzeigen lassen. Klicken Sie dazu auf den Link [Abb. B \(1\) \(PDF-Datei\)](#). Unter (2) wird eine genaue Beschreibung des Generierungsprozesses abgelegt. Die Testvorschau im Produktions-Paket wird durch die Programme des Binär-Pakets generiert, also im gleichen Verfahren wie später beim Druck (Wenn Sie sich noch nicht mit Binär-Paketen vertraut gemacht haben, lesen Sie bitte den Abschnitt über die [Umgebung des R-S](#) und den Abschnitt über [Binär-Pakete](#)). In der Testvorschau wird deshalb exakt das Ergebnis produziert, das Sie später im Druck erwarten können. Sie können sich also darauf verlassen, dass die Dokumente, so wie Sie in der Testvorschau generiert wurden, auch gedruckt werden. Sollten beim Generierungsprozess Fehler aufgetreten sein, so können Sie deren Beschreibung unter (2) einsehen.

Nachdem Sie die Vorschau begutachtet haben, schließen Sie die Vorschau durch Schließen.

## Registerkarte - Test-Zyklus

Die Registerkarte Test-Zyklus bietet eine Möglichkeit, Anmerkungen und Notizen zu dem Paket zu speichern, die sich aus der Test-Phase ergeben. Dies ist z. B. für QA-Teams gedacht, die in Abb. A Anmerkungen, Hinweise etc. aus der Test-Phase des Pakets/Schemas festhalten und somit an das Produktions-Team weitergeben können.

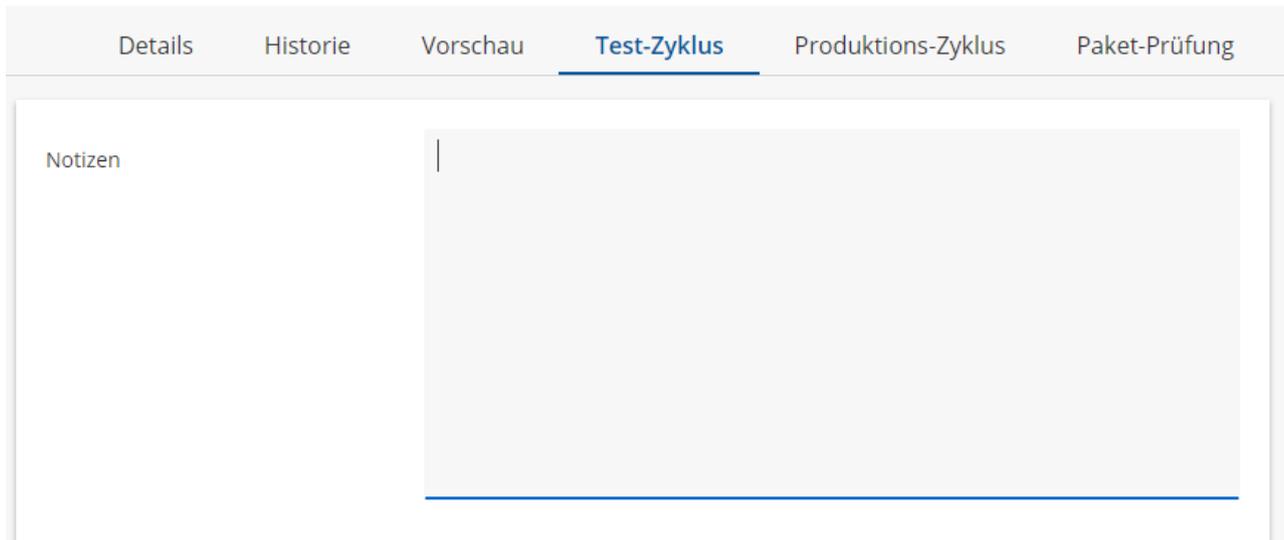


Abbildung A

## Registerkarte - Produktions-Zyklus

Die Registerkarte Produktions-Zyklus bietet analog zur **Registerkarte Test-Zyklus** die Möglichkeit, Anmerkungen und Notizen Abb. A zu dem Paket zu speichern. Im Gegensatz zur Registerkarte Test-Zyklus, in der Anmerkungen zur Test-Phase festgehalten werden sollten ist diese Registerkarte aber dazu gedacht, Anmerkungen zur Produktions-Phase zu speichern.



Abbildung A

## Registerkarte - Paket-Prüfung

In der Übersichtstabelle *Abb. A* (1) sehen Sie Meldungen, die bei der Validierung des Produktions-Paketes durch den R-S Server erstellt worden sind. Diese Meldungen werden in drei Typen aufgeteilt: **Error**, **Warning** und **Info**.

Der Typ der Meldung steht in der linken Spalte (1). In der rechten Spalte (2) sehen Sie die dazugehörige Beschreibung.

<span>Details</span> <span>Historie</span> <span>Vorschau</span> <span>Test-Zyklus</span> <span>Produktions-Zyklus</span> <span><b>Paket-Prüfung</b></span>	
①	②
{INFO}	extension[RS_LOGO_Rechts] has binding to extension-point[RS_LOGO_Rechts]
{INFO}	extension[RS_ABSENDER] has binding to extension-point[RS_ABSENDER]
{INFO}	extension[RS_Betreff] has binding to extension-point[RS_Betreff]
{INFO}	extension[RS_Infobox] has binding to extension-point[RS_Infobox]
{INFO}	extension[RS_Fuss] has binding to extension-point[RS_Fuss]
{INFO}	extension[Selbstablesung] has binding to extension-point[LOGICAL_SUPPLEMENT]
{INFO}	extension[Werbebanner] has binding to extension-point[WHITESPACE]
{INFO}	resource[BILD_Profiforms_BANNER] comes from[sr-104165]
{INFO}	resource[BILD_Signatur] comes from[sr-104165]
{INFO}	resource[R_Signatur_TEXT] comes from[sr-104165]
{INFO}	resource[RT_Brieftext_SA01] comes from[sr-104165]

Abbildung A

## Status eines Pakets setzen

In der Produktions-Paket-Übersichtstabelle steht Ihnen folgende Aktion zur Verfügung, um den Status eines Produktions-Pakets zu ändern:

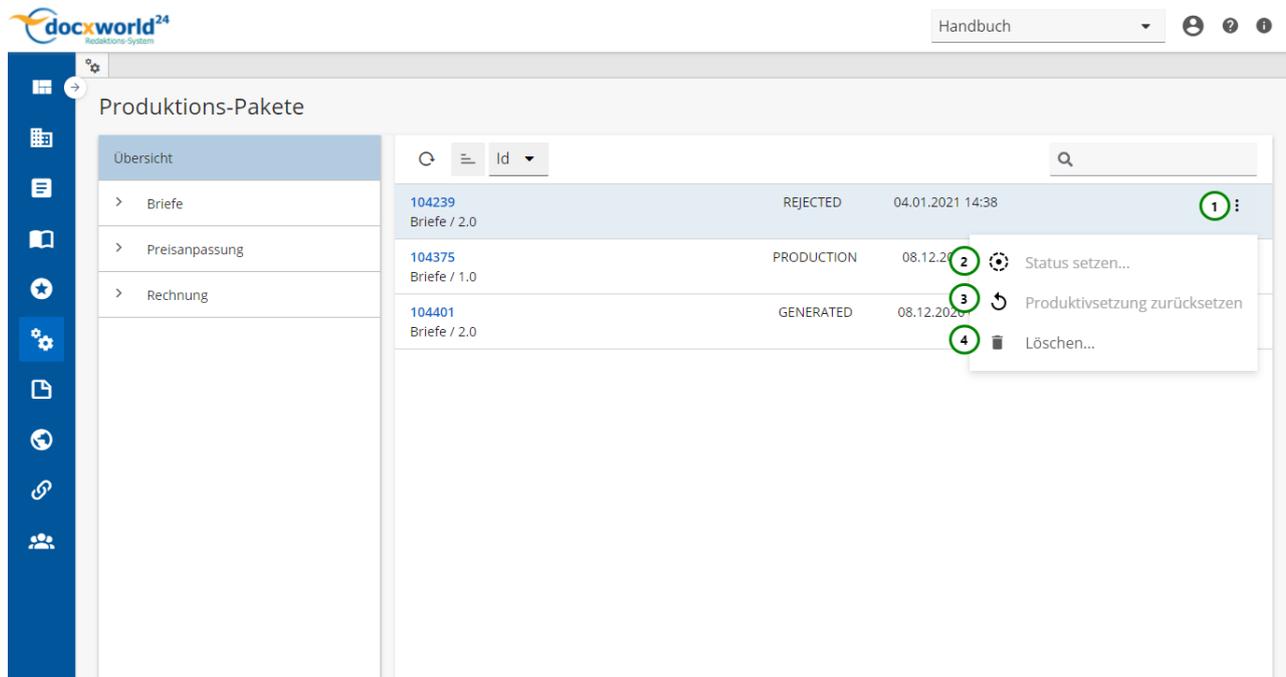


Abbildung A

Produktions-Pakete sind wie **Formulare**, **Schemas** oder **Bibliotheken** Teil des R-S Workflows. Sie sind deshalb der Versionierung unterworfen und besitzen einen **Status**. Lesen Sie dazu bitte den Abschnitt über den **R-S Workflow**.

Öffnen Sie das Kontextmenü  Abb. A (1) des Produktions-Pakets, dessen Status Sie ändern möchten und klicken Sie auf  (2). Folgendes Fenster öffnet sich:

## Status setzen ×

Aktueller Status ①

Neuer Status ②

Beschreibung ③

Abbrechen

Abbildung B

Bei Produktions-Paketen im Status **REJECTED** können Sie keinen Statuswechsel mehr durchführen.

**Sie sehen:**

- Den aktuellen **Status** des Produktions-Pakets Abb. B (1).

Wählen Sie in Abb. B (2) den neuen Status des Produktions-Pakets. Die Auswahl in (2) wird durch den aktuellen Status bestimmt:

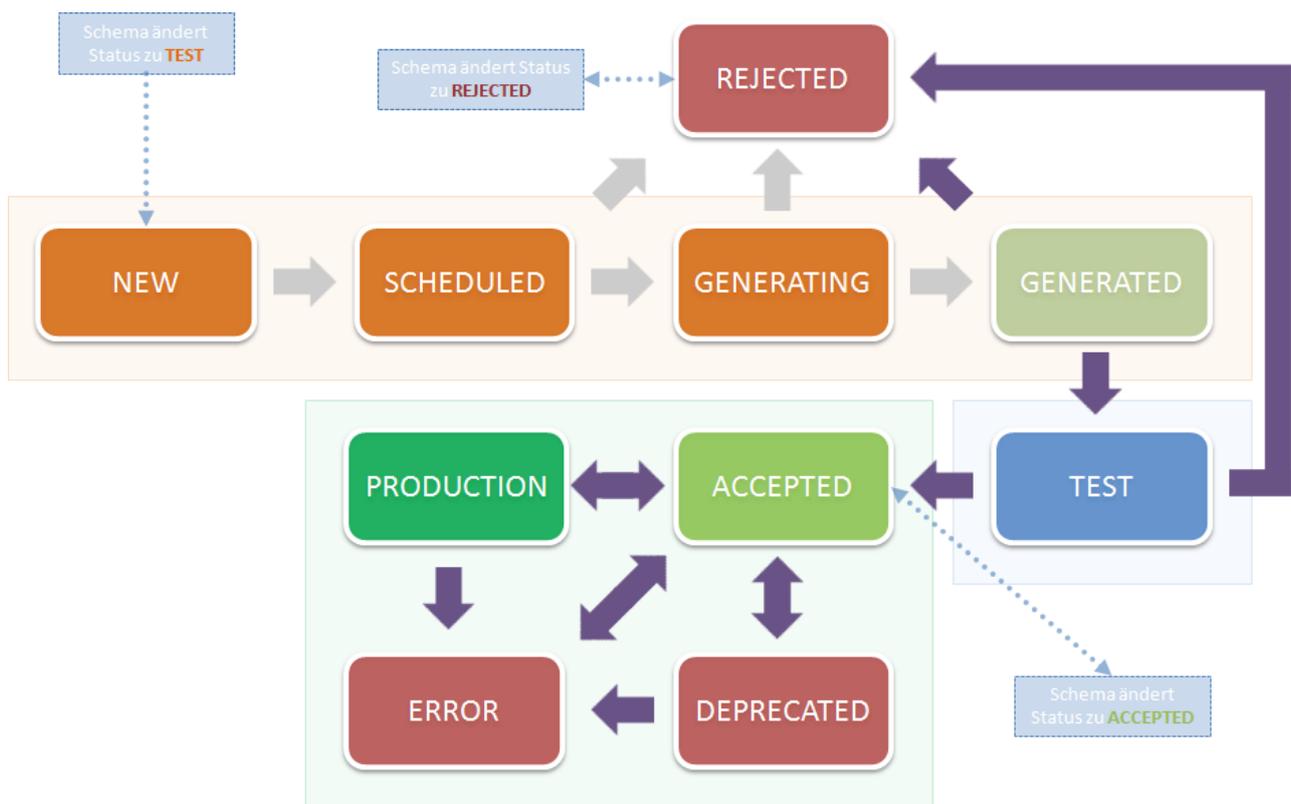


Abbildung C



**Statusänderung durch Nutzer möglich.**



**Statusänderung durch das R-S.**

Ändert sich der Status einer Schema-Version zu **TEST**, wird automatisch ein Produktions-Paket für die Schema-Version generiert. Tritt bei der Generierung des Pakets ein Fehler auf, wird der Status zu **REJECTED** geändert und das Paket wird unbrauchbar. Selbiges passiert, wenn der Status der Schema-Version auf **REJECTED** geändert wird. Wird die Schema-Version dann weiter auf **EDIT** und dann wieder auf **TEST** gesetzt, wird ein neues Paket generiert, das Alte bleibt im Status **REJECTED** und unbrauchbar. Ist der Generierungsprozess erfolgreich abgeschlossen, erhält das Paket den Status **GENERATED**.

Der Status des Pakets wird dann durch den Nutzer auf **TEST** und anschließend entweder auf **ACCEPTED** oder, um das Paket zurückzuweisen, auf **REJECTED** gesetzt. Ein Produktions-Paket mit dem Status **TEST** erhält außerdem automatisch den Status **ACCEPTED**, wenn die Schema-Version auf **ACCEPTED** gesetzt wird, gleiches gilt auch andersherum.

Um das Produktions-Paket als produktionsbereit zu markieren, ändern Sie den Status auf **PRODUCTION**. Produktions-Pakete können in der Produktion nur dann benutzt werden, wenn sie **PRODUCTION**-Status haben. Um das Paket als veraltet zu markieren, setzen Sie den Status **DEPRECATED**, den Sie wieder rückgängig machen können. Um das Paket als fehlerhaft zu markieren, setzen Sie den Status **ERROR**, den Sie ebenfalls wieder rückgängig machen können.

 Für eine genaue Beschreibung des Status-Workflows bei Produktions-Paketen lesen Sie bitte unbedingt den [Abschnitt über den R-S Workflow](#).

 Sie können außerdem eine **Beschreibung** hinzufügen [Abb. B \(3\)](#). Die Beschreibung wird beispielsweise in der [Historie](#) angezeigt.

**Produktions-Paket in den PRODUCTION-Status versetzen:**

The screenshot shows a dialog box titled "Status setzen" with a close button (X) in the top right corner. The dialog contains the following fields and controls:

- Aktueller Status:** A text field containing "ACCEPTED", marked with a green circle containing the number 1.
- Neuer Status:** A dropdown menu showing "PRODUCTION" with a downward arrow, marked with a green circle containing the number 2.
- Beschreibung:** A large empty text area, marked with a green circle containing the number 3.
- Zeitgesteuerte Produktivsetzung:** A checkbox that is checked, marked with a green circle containing the number 4.
- Zeitpunkt Produktivsetzung:** A date picker field, marked with a green circle containing the number 5. Below it, the text "Zeitpunkt ist nicht bestimmt" is displayed in red.

At the bottom of the dialog, there are two buttons: "Abbrechen" (disabled) and "Ok" (disabled).

Abbildung D

Wenn Sie ein Produktions-Paket in den PRODUCTION -Status versetzen [Abb. D](#), kann es im externen Produktions-Prozess verwendet werden (Siehe: [Umgebung des R-S](#)). Pakete können erst dann für externe Komponenten des Dokument-Erstellungs-Prozesses sicht- und nutzbar gemacht werden, wenn Sie PRODUCTION -Status haben (Ausnahme: Testproduktion). Von einem Schema kann immer nur ein Produktions-Paket den PRODUCTION -Status besitzen. Besitzt schon ein anderes Produktions-Paket des Schemas PRODUCTION -Status, wird das alte Paket in den ACCEPTED -Status zurückversetzt. Wenn Sie [Docxworld](#) von profiforms für Ihre Dokumenterstellung nutzen, müssen Sie das Paket mit Hilfe von [Docxworld-Links](#) verfügbar machen. Ansonsten nutzen Sie dazu bitte [Produktions-Paket Verknüpfungen](#).

Wenn Sie das Paket in den PRODUCTION -Status versetzen, können Sie neben der sofortigen Produktivsetzung (Das Paket wird sofort in den neuen Status versetzt und ist für die Produktion sofort verfügbar) auch eine zeitgesteuerte Produktivsetzung festlegen [Abb. D](#) (4). Das Paket wird dann erst am angegebenen Zeitpunkt in den PRODUCTION -Status versetzt  (5). Damit können Sie punktgenau Pakete in die Produktion schicken.

So kann z.B. während laufender Produktion ein altes Paket des Schemas zu einer festgelegten Zeit durch das neue Paket als das produktives Paket abgelöst werden. Im laufenden Produktions-Prozess wird das alte Paket ab dem Zeitpunkt, ab dem das neue Paket **PRODUCTION** -Status hat, abgelöst. Wenn das neue Paket den **PRODUCTION** -Status zugewiesen bekommt, wird das alte in den **ACCEPTED** -Status versetzt. Es kann später auch wieder produktiv gesetzt werden. Lesen Sie dazu bitte den [Abschnitt über Produktions-Pakete](#) .

 Damit externe Komponenten des Dokument-Erstellungs-Prozesses (wie der **Spooler**) das Paket der richtigen Produktion zuordnen können, müssen Sie entweder einen **Docxworld-Link** oder eine **Produktions-Paket Verknüpfung** für das Paket definieren!

Um die Änderungen zu speichern, klicken Sie bitte auf Ok . Das Produktions-Paket hat jetzt den neuen Status. Je nach Status hat sich auch der Status der Schema-Version geändert.

Um die Änderungen zu verwerfen, klicken Sie bitte auf Abbrechen.

Nachdem Sie einem Produktions-Paket eine zeitgesteuerte Produktivsetzung zugeordnet haben, sehen Sie die geplante Produktivsetzung in der Übersicht [Abb. E \(1\)](#) :

104401 Briefe / 2.0	ACCEPTED	04.01.2021 15:55	<span style="border: 1px solid green; border-radius: 50%; padding: 2px;">1</span>	05.01.2021 12:00	⋮
------------------------	----------	------------------	---	------------------	---

Abbildung E

### **Zeitgesteuerte Produktivsetzung zurücksetzen**

Nachdem Sie für ein Paket einen Zeitplan (Zeitpunkt für Produktivsetzung) eingestellt haben (siehe [Status eines Pakets setzen](#) ), können Sie diesen auch wieder aus der Übersichtstabelle löschen, wenn Sie ihn verwerfen möchten.

Öffnen Sie das Kontextmenü  [Abb. A \(1\)](#) des Produktions-Pakets, dessen Zeitplan Sie zurücksetzen möchten, und klicken Sie auf  Produktivsetzung zurücksetzen (3).

Tragen Sie in der nachfolgenden Abfrage den Grund Ihres Zurücksetzen ein. Dieser wird z.B. in der Historie des Pakets erfasst.

### **Produktions-Paket löschen**

Um ein Produktions-Paket zu löschen, darf das Paket nicht mehr den Status **TEST** oder **PRODUCTION** haben.

Öffnen Sie das Kontextmenü  [Abb. A \(1\)](#).

des Produktions-Paketes, dass Sie löschen möchten, und klicken Sie auf  löschen (4). Bestätigen Sie den nachfolgenden Dialog mit Ok, um den Löschvorgang abzuschließen.

## Arbeiten mit Materialien

In diesem Kapitel wird das Arbeiten mit Materialien behandelt. Materialien werden verwendet um als physikalische oder logische Beileger Transaktions-Dokumenten zugeordnet zu werden. Diese Zuordnung erfolgt über die Verwendung in Schema-Versionen.

**! WICHTIG:** Im Gegensatz zu Versionen von Transaktions-Formularen oder Bibliotheken, werden Materialien **NICHT** versioniert. Dies bedeutet, dass bei der Zuordnung von Materialien zu Schema-Versionen, die Material-Informationen in das Schema kopiert werden. Dies stellt sicher, dass sich Änderungen von Material-Informationen nicht automatisch in Schema-Versionen auswirken. Sollen sich Änderungen im Material auch in Schema-Versionen niederschlagen, so muss das entsprechende Material in den Schema-Versionen neu selektiert werden!

Um zu den Übersichtstabellen der Produktions-Pakete zu gelangen, wechseln Sie in den Navigationspunkt  Material. Sie können für logische und physikalische Beileger jeweils unterschiedliche Materialien anlegen. Wechseln Sie dazu in den Reiter für Physikalische Beileger (1) oder Logische Beileger (2). Ihnen werden anschließend nur Materialien der ausgewählten Kategorie angezeigt (3).

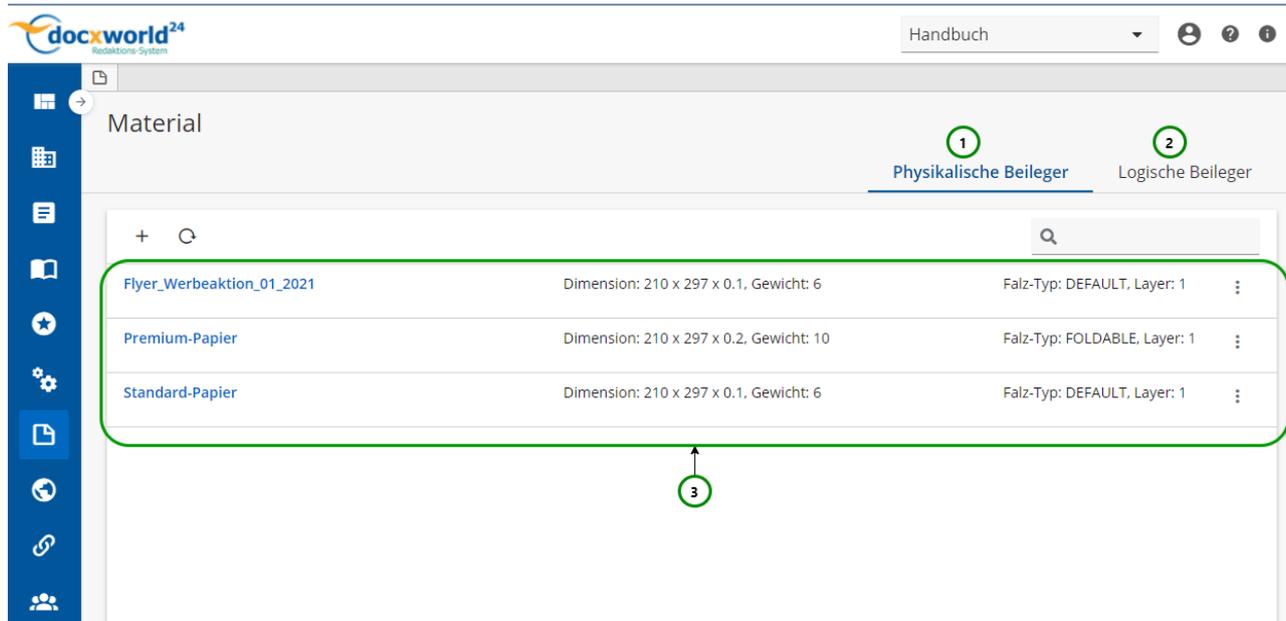


Abbildung A

Materialien dienen als Grundlage für Beileger. In Ihnen werden Maße, Gewicht und andere Eigenschaften des Beilegers festgelegt, was später für den Druck und den Versand wichtig ist (z.B. zur Berechnung des Portos). Wenn Sie einem Formular einen Beileger geben, müssen Sie immer ein zuvor erstelltes Material auswählen, das die physischen Eigenschaften des Beilegers angibt.



**Ausnahme:** Wenn das Formular für logische Beileger nur Formular eigenes Material erlaubt, können Sie nur die durch das Formular zur Verfügung gestellten Materialien auswählen.

Falls Sie den Abschnitt über Materialien noch nicht gelesen haben, tun Sie dies bitte [hier](#), bevor Sie beginnen, mit Materialien zu arbeiten.

## Material für physikalischen Beileger erstellen, ändern und löschen

In der Material-Übersichtstabelle für physikalische Beileger stehen Ihnen folgende Aktionen zur Verfügung, um ein Material für physikalische Beileger neu zu erstellen, zu bearbeiten oder zu löschen.

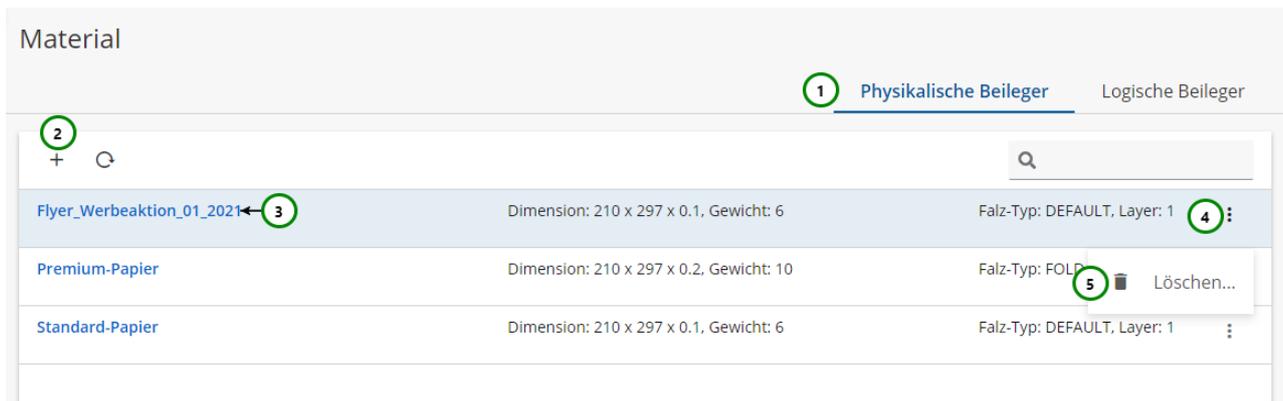


Abbildung A

Falls Sie sich noch nicht mit Materialien für physikalische Beileger vertraut gemacht haben, lesen Sie bitte zuerst diesen [Abschnitt](#). Mit Hilfe von [Abb. A \(5\)](#) können Sie die Materialliste durchsuchen. Ausführliche Informationen zur Suchfunktion finden Sie [hier](#).

### Material erstellen und ändern

Wechseln Sie auf den Reiter für Physikalische Beileger [Abb. A \(1\)](#).

Um ein neues Material für einen physikalischen Beileger zu **+** **erstellen**, klicken Sie bitte auf (2).

Um das Material für einen physikalischen Beileger zu **ändern**, wählen Sie das Material, das Sie ändern möchten aus und klicken Sie auf dessen Namen (3).

In beiden Fällen öffnet sich folgende Seite:

The screenshot shows a web interface for entering material data. The title is 'Material für physikalische Beileger'. The form has the following fields and callouts:

- Name \***: Input field with value 'Flyer\_Werbeaktion\_01\_2021' (callout 1).
- Größe \***: Three input fields for 'Breite' (210, callout 2), 'Höhe' (297, callout 3), and 'Stärke' (0.1, callout 4).
- Gewicht \***: Input field with value '6' (callout 5).
- Falzung \***: Dropdown menu with 'Standard' selected (callout 6) and an input field with value '1' (callout 7).
- Vorschau**: File upload area with text 'Bitte wählen Sie eine Datei zum Hochladen' (callout 8) and a 'Suchen' button (callout 9).
- Beschreibung**: Large empty text area (callout 10).

Buttons at the bottom: 'Speichern', 'Speichern & Schließen', 'Zurücksetzen'.

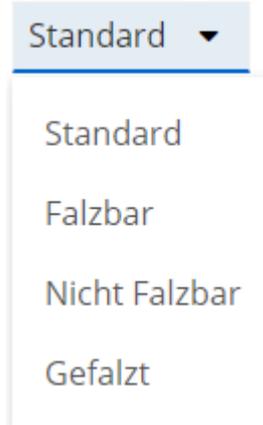
Abbildung B

Da physikalische Beileger den Dokumenten beim Druck fertig hinzugefügt werden, müssen vorher die Eigenschaften des Beilegers bekannt sein, um z.B. Porto oder Umschlaggröße berechnen zu können. Auch, ob ein optionaler Beileger noch beigelegt wird, wird durch diese Angaben bestimmt. Lesen Sie dazu den Abschnitt über [Materialen physikalischer Beileger](#).

**Sie müssen dem Material:**

- ★ einen **Namen** geben *Abb. B (1)*.
- ★ eine **Dimension** geben. Anzugeben sind die physischen Dimensionen des Beilegers:
  - Breite: Die Breite des Beilegers in Millimeter (2).
  - Höhe: Die Höhe des Beilegers in Millimeter (3).
  - Stärke: Die Dicke des Beilegers in Millimeter (4).
  - ⊖ Die hier gemachten Angaben müssen unbedingt korrekt sein, da sonst eine ordnungsgemäße Berechnung des zu verwendenden Umschlagformats etc. nicht möglich ist.
  - ⚠ Die Eingabe-Feld-Validierung erfolgt nach **DIMENSION**
- ★ ein **Gewicht** geben (5).
  - Das Gewicht des Beilegers in Gramm (g).
  - ⊖ Die hier gemachten Angaben müssen unbedingt korrekt sein, da sonst eine ordnungsgemäße Berechnung z.B. des Portos nicht möglich ist.
  - ⚠ Die Eingabe-Feld-Validierung erfolgt nach **WEIGHT**.

★ einen **Falz-Typ** zuordnen. Klicken Sie auf (6) . Es klappt ein Auswahlmü aus:



Wählen Sie einen Falz-Typ aus.

Der Falz-Typ gibt an, wie der Beileger vorliegt und wie mit ihm verfahren werden darf:

**DEFAULT** - Standardeinstellung, bedeutet der Beileger darf gefalzt werden.

**FOLDABLE** - Bedeutet, der Beileger darf gefalzt werden.

**NOT\_FOLDABLE** - Bedeutet, der Beileger darf nicht gefalzt werden. Beachten Sie, das großformatige Beileger (z.B. A4) dann auch eine entsprechende Umschlaggröße benötigen.

**FOLDED** - Bedeutet, dass der Beileger bereits gefalzt vorliegt.

★ eine **Falz-Layer** geben (7).

Der Falz-Layer gibt an, wie viele Layer (Papierschichten) der gefalzte Beileger hat. Im Leporellofalz gefalztes Papier hat beispielsweise den Falz-Layer 3.

⚠ Die Eingabe-Feld-Validierung erfolgt nach **INTEGER** .

★ eine **Beispiel-Vorschau** zuordnen.

Beispiel-Vorschauen dienen dazu, im R-S eine Ansicht des fertigen Beilegers anhängen zu können, um den R-S Beileger besser auf den realen Beileger abstimmen zu können.

- Beim Erstellen/Ändern klicken Sie bitte auf Suchen (9). Es öffnet sich der Windows-Explorer. Wählen Sie nun eine Datei aus.
- bereits zugeordnete Vorschauen finden Sie unter (8). Um die hochgeladene Vorschau zu entfernen, klicken Sie bitte auf ✕ .

★ eine Beschreibung geben (10). Beim Erstellen und Ändern müssen Sie das Feld nicht zwingend ausfüllen.

Um die gemachten Änderungen anzunehmen und das Material zu speichern, klicken Sie auf Speichern.

Um die Änderungen zu verwerfen, klicken Sie auf Zurücksetzen oder Schließen Sie den Tab.

### **Material löschen**

Um ein Material für einen physikalischen Beileger zu **löschen**, wählen Sie bitte das Material, das Sie löschen möchten, öffnen Sie dessen  Kontextmenü *Abb. A (4)* und wählen Sie anschließend  Löschen (5). Folgender Dialog öffnet sich:



Abbildung C

 Beachten Sie, dass die Verweise aller Ressourcen innerhalb von Schema- und Bibliotheks-Versionen, die den Beileger nutzen, ungültig werden. Kontrollieren Sie deshalb vorher, ob Sie den Beileger wirklich löschen wollen.

Klicken Sie auf Ok, um das Löschen zu bestätigen, oder auf Cancel , um das Löschen abubrechen.

## Material für logischen Beileger erstellen, ändern und löschen

In der Material-Übersichtstabelle für logische Beileger stehen Ihnen folgende Aktionen zur Verfügung, um ein Material für logische Beileger neu zu erstellen, zu bearbeiten oder zu löschen:

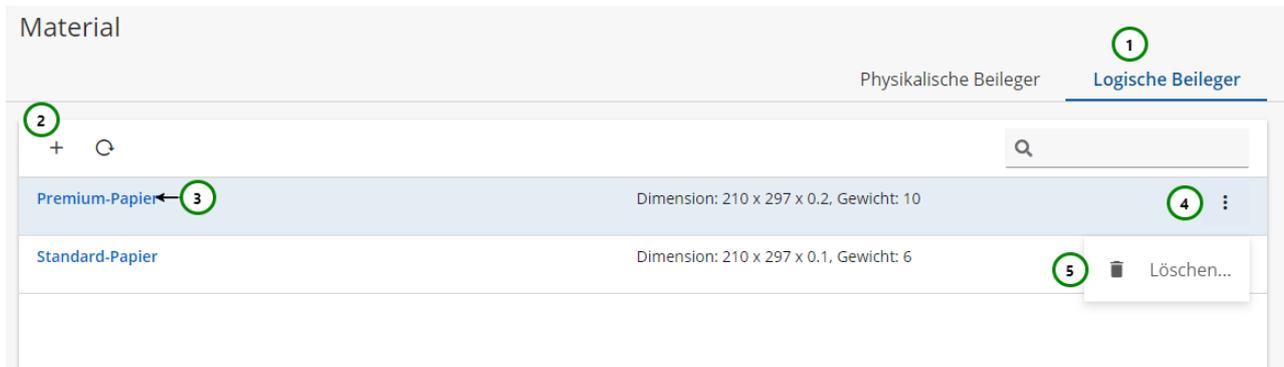


Abbildung A

Falls Sie sich noch nicht mit Materialien für logische Beileger vertraut gemacht haben, lesen Sie bitte zuerst diesen [Abschnitt](#). Mit Hilfe von [Abb. A](#) (5) können Sie die Materialliste durchsuchen. Ausführliche Informationen zur Suchfunktion finden Sie [hier](#).

### Material erstellen und ändern

Wechseln Sie auf den Reiter für Logische Beileger [Abb. A](#) (1).

Um ein neues Material für einen physikalischen Beileger zu **+** **erstellen**, klicken Sie bitte auf (2).

Um das Material für einen physikalischen Beileger zu **ändern**, wählen Sie das Material, das Sie ändern möchten aus und klicken Sie auf dessen Namen (3).

In beiden Fällen öffnet sich folgende Seite:

The screenshot shows a web application interface for managing materials. The title is 'Material für logische Beileger'. The form includes the following fields and callouts:

- 1**: Name \* (Premium-Papier)
- 2**: Breite (210)
- 3**: Höhe (297)
- 4**: Stärke (0.2)
- 5**: Gewicht \* (10)
- 6**: Vorschau (Bitte wählen Sie eine Datei zum Hochladen) with a **7** 'Suchen' button.
- 8**: Beschreibung (empty text area)
- 9**: Erstellt / Geändert (05.01.2021 08:20)
- 10**: Erstellt / Geändert (05.01.2021 08:20)

Buttons at the bottom: Speichern, Speichern & Schließen, Zurücksetzen.

Abbildung B

Logische Beileger werden zwar direkt mit den anderen Dokumenten mit gedruckt (im Gegensatz zu physikalischen Beilegern), es ist allerdings auch möglich, logische Beileger auf anderem Papier drucken zu lassen wie die restlichen Dokumente (Wenn das Transaktions-Formular dies erlaubt, lesen Sie hierzu die **Definition logischer Beileger**). Deshalb müssen auch bei logischen Beilegern Dimensionen und Eigenschaften angegeben werden, um in diesem Fall die Eigenschaften des Beilegers für Druck und Versand berechnen zu können (Porto, Umschlaggröße etc.). Auch, ob ein optionaler Beileger noch beigelegt wird, wird durch diese Angaben bestimmt. Lesen Sie dazu den Abschnitt über **Materialien logischer Beileger**.

**Sie müssen dem Material:**

- ★ einen **Namen** geben *Abb. B (1)*.
- ★ eine **Dimension** geben. Anzugeben ist sind die physischen Dimensionen des Beilegers:  
 Breite: Die Breite des Beilegers in Millimeter (2).  
 Höhe: Die Höhe des Beilegers in Millimeter (3).  
 Stärke: Die Dicke des Beilegers in Millimeter (4).  
 ⓪ Die hier gemachten Angaben müssen unbedingt korrekt sein, da sonst eine ordnungsgemäße Berechnung des zu verwendenden Umschlagformats etc. nicht möglich ist.  
 ⚠ Die Eingabe-Feld-Validierung erfolgt nach **DIMENSION**.
- ★ ein **Gewicht** geben (5).  
 Das Gewicht des Beilegers in Gramm (g).  
 ⓪ Die hier gemachten Angaben müssen unbedingt korrekt sein, da sonst eine ordnungsgemäße Berechnung z.B. des Portos nicht möglich ist.  
 ⚠ Die Eingabe-Feld-Validierung erfolgt nach **WEIGHT**.
- ★ eine **Beispiel-Vorschau** zuordnen.  
 Beispiel-Vorschauen dienen dazu, im R-S eine Ansicht des fertigen Beilegers anhängen zu können, um den R-S Beileger besser auf den realen Beileger abstimmen zu können.
  - Beim Erstellen/Ändern klicken Sie bitte auf Suchen (7). Es öffnet sich der Windows-Explorer. Wählen Sie nun eine Datei aus.
  - bereits zugeordnete Vorschauen finden Sie unter (6). Um die hochgeladene Vorschau zu entfernen, klicken Sie bitte auf .
- ★ eine **Beschreibung** geben (8). Die Beschreibung wird z.B. in Auswahldialogen angezeigt.

Um die gemachten Änderungen anzunehmen und das Material zu speichern, klicken Sie auf Speichern. Um die Änderungen zu verwerfen, klicken Sie auf Zurücksetzen oder Schließen Sie den Tab.

**Material löschen**

Um ein Material für einen logischen Beileger zu **löschen**, wählen Sie bitte das Material, das Sie löschen möchten, öffnen Sie dessen  Kontextmenü *Abb. A (4)* und wählen Sie anschließend  Löschen (5). Folgender Dialog öffnet sich:



Abbildung C

 Beachten Sie, dass die Verweise aller Ressourcen innerhalb von Schema- und Bibliotheks-Versionen, die den Beileger nutzen, ungültig werden. Kontrollieren Sie deshalb vorher, ob Sie den Beileger wirklich löschen wollen.

Klicken Sie auf Ok, um das Löschen zu bestätigen, oder auf Cancel, um das Löschen abubrechen.

## Arbeiten mit Docxworld-Links

In diesem Kapitel wird das Arbeiten mit Docxworld-Links behandelt. Um zur Übersichtstabelle der Docxworld-Links zu gelangen, wechseln Sie in den Navigationspunkt  docxworld.

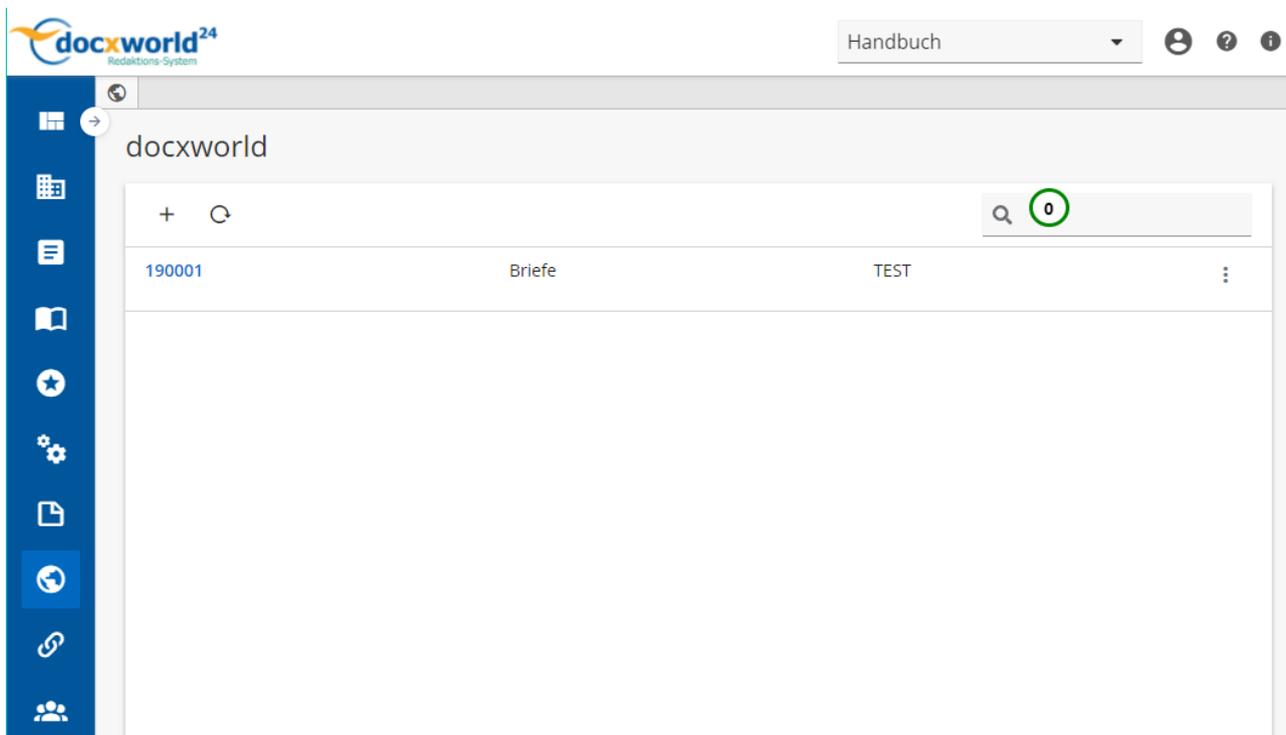


Abbildung A

Mit Docxworld-Links machen Sie Produktions-Pakete und die darin enthaltenen Dokumente für den Dokumenterstellung-Prozess verfügbar. Externe Komponenten (*Spooler*) können dann Produktions-Pakete durch die in den Datenströmen enthaltene **Vertragsnummer** zuordnen und in den Produktions-Prozess einschleusen. Lesen Sie bitte den [Abschnitt über Docxworld-Links](#), bevor Sie beginnen, damit zu arbeiten. Wenn Sie kein Docxworld für Ihre Produktion benutzen, müssen Sie **Produktions-Paket Verknüpfungen** benutzen.

Sie können die Docxworld-Link-Liste mit Hilfe von  (0) durchsuchen. Ausführliche Informationen zur Suchfunktion finden Sie [hier](#).

## Docxworld-Links erstellen, ändern und löschen

Sie sehen folgende Übersicht, die im Arbeitsbereich vorhandenen Docxworld-Links enthält und es Ihnen erlaubt, neue Docxworld-Verlinkungen zu erstellen oder vorhandene Links zu bearbeiten. Docxworld-Links binden Produktions-Pakete an einen Docxworld-Vertrag. Diesen Link nutzt bspw. der **Spooler** bei der Dokumenterstellung, um das den Datenströmen zugehörige Produktions-Paket zu finden. Lesen Sie den **Abschnitt über Docxworld-Links**, bevor Sie damit arbeiten.



Abbildung A

Sie sehen in **Abb. A** :

- (1) Docxworld-Link **+** erstellen.
- (2) Name bzw. Vertragsnummer des Docxworld-Link.
- (3) Das selektierte Schema des Docxworld-Link.
- (4) Das ausgewählte Produktions-Paket bzw. die Status Referenz auf ein Produktions-Paket.
- (5) Das **⋮** Kontextmenü des Docxworld-Link.

Um einen Docxworld-Link zu **+** erstellen, klicken Sie bitte auf **Abb. A (1)**.

Um einen Docxworld-Link zu **ändern**, klicken Sie bitte auf den Namen des Docxworld-Links (2).

In beiden Fällen öffnet sich folgende Ansicht:

### Vertragsbestandteile

Vertrag *	<b>1</b>	190001
Schema *	<b>2</b>	Briefe ▾
Paket *	<b>3</b>	Test
Beschreibung	<b>4</b>	Test
	<b>5</b>	Produktion
	<b>6</b>	Selektieren
Erstellt / Geändert	<b>7</b>	05.01.2021 10:59      05.01.2021 10:59

Abbildung B

**Sie können dem Docxworld-Link:**

- ★ eine **Vertrags-Nummer** geben *Abb. B (1)*.

Die Vertragsnummer gibt an, welchem Docxworld-Vertrag Sie dem Produktions-Paket zuordnen möchten. Die Vertragsnummer muss mit der Vertragsnummer übereinstimmen, die im eingehenden Datenstrom enthalten ist, den das Produktions-Paket verwendet. An einen Vertrag kann immer nur ein Produktions-Paket gebunden sein. Lesen Sie dazu den Abschnitt über die **Umgebung des R-S** und **Docxworld-Links**.

- ⊖ Die beiden Vertragsnummern müssen unbedingt übereinstimmen, da sonst eine Zuordnung von Datenströmen und Dokument nicht möglich ist!

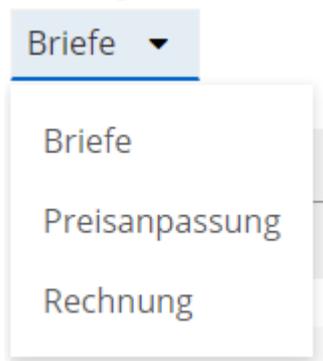
Ist der Vertragsnummer schon ein anderes Produktions-Paket zugeordnet, wird folgende Fehlermeldung angezeigt:



Da jede Produktion in Docxworld eindeutig einem Vertrag zugeordnet ist, muss die Vertragsnummer nicht unbedingt durch den aktuellen Nutzer belegt sein und kann eventuell trotzdem nicht verwendet werden. Setzen Sie sich mit dem Formular-Designer in Verbindung, wenn Sie die Vertragsnummer des ihrer Dokumente zugehörigen Datenstroms nicht wissen.

- ⚠ Die Eingabe-Feld-Validierung erfolgt nach **INTEGER**.

- ★ Dem Vertrag ein **Schema** zuordnen. Klicken Sie auf (2) . Es klappt ein Auswahlménü aus:



*Wählen Sie ein Schema aus.*

In der Auswahl werden Ihnen alle verfügbaren Schemas angezeigt. Da von jedem Schema immer nur ein produktionsbereites Produktions-Paket (im PRODUCTION -Status) existieren kann, ist mit Auswahl des Schemas das Paket eindeutig bestimmt. Sie können das Produktions-Paket allerdings auch manuell auswählen.

- ★ ein **Produktions-Paket** selektieren oder auf Produktion bzw. Test setzen.

Wenn Sie ein Produktions-Paket manuell auswählen möchten, können Sie das mit (6) tun. Hier ist es auch möglich, Produktions-Pakete mit DEPRECATED Status auszuwählen. Sie haben dann folgende Möglichkeiten:

- **Produktion (5):** Das Produktions-Paket im PRODUCTION -Status des selektierten Schemas wird für diesen Link verwendet.  
Der Vertrag ist damit für die Produktion freigegeben und kann im Produktions-Prozess genutzt werden.
- **Test (4):** Das Produktions-Paket im TEST -Status des selektierten Schemas wird für diesen Link verwendet.  
Der Vertrag ist damit ein Testvertrag und die Dokumente können einem Test-Produktions-Prozess durchlaufen. Hierzu benötigen Sie eine Vertragsnummer, die für Testproduktionen registriert ist.

★ eine **Beschreibung** geben (7).  
Die Beschreibung ist optional und wird z.B. in Auswahldialogen angezeigt.

Verwenden Sie zum Speichern des Docxworld-Links Speichern oder Speichern und Schließen.  
Um die Änderungen zu verwerfen, klicken Sie bitte auf Zurücksetzen oder schließen Sie den Tab.

### Vertrag löschen

Um einen Docxworld-Link zu löschen, öffnen Sie dessen  Kontextmenü *Abb. A (5)* und wählen Sie anschließend  Löschen (6). Folgender Dialog öffnet sich:



Abbildung C

Klicken Sie auf Ok, um das Löschen zu bestätigen oder auf Cancel, um das Löschen abzubrechen.

 Wenn ein Vertrag gelöscht wird, ist das zugehörige Produktions-Paket nicht mehr durch die Vertragsnummer ansprechbar und damit auch im Produktions-Prozess nicht mehr verfügbar.

## Arbeiten mit Produktions-Paket Verknüpfungen

In diesem Kapitel wird das Arbeiten mit Produktions-Paket Verknüpfungen behandelt. Um zu den Übersichtstabellen der Verknüpfungen zu gelangen, wechseln Sie in den Navigationspunkt  Produktions-Paket Links.

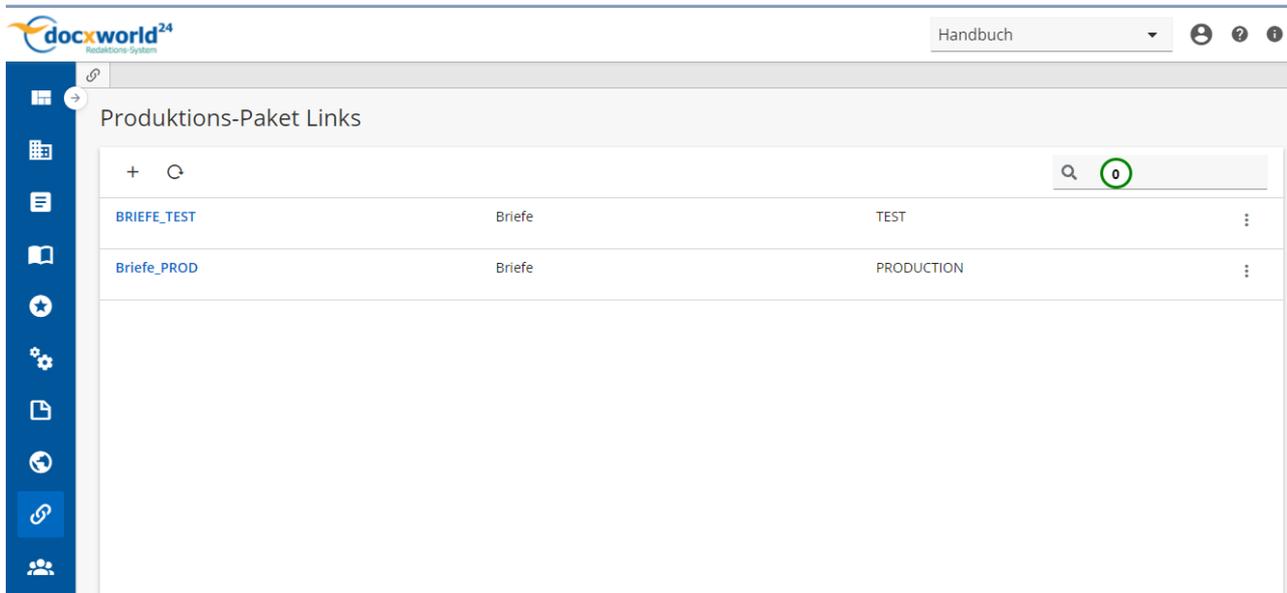


Abbildung A

Mit Produktions-Paket Verknüpfungen machen Sie **Produktions-Pakete** und die darin enthaltenen Dokumente für den Dokumenterstellung-Prozess verfügbar. Externe Komponenten (**Spooler**) können hiermit Produktions-Pakete durch den in den Datenströmen enthaltenen **Namen** zuordnen und in den Produktions-Prozess einschleusen. Lesen Sie bitte den **Abschnitt über Produktions-Paket Verknüpfungen**, bevor Sie beginnen, damit zu arbeiten. Produktions-Paket Verknüpfungen müssen Sie benutzen, wenn Sie kein **Docxworld** für Ihre Dokument-Erstellung benutzen. Wenn Sie Docxworld für Ihre Produktion benutzen, müssen Sie **Docxworld-Links** benutzen.

Sie können die Produktions-Paket-Verknüpfungs-Liste mit Hilfe von  (0) durchsuchen. Ausführliche Informationen zur Suchfunktion finden Sie [hier](#).

## Verknüpfung erstellen und ändern

Sie sehen folgende Oberfläche, mit der Sie Verknüpfungen erstellen und ändern können:



Abbildung A

Sie sehen in **Abb. A** :

- (1) Produktions-Paket Link **+** **erstellen**.
- (2) Name bzw. Vertragsnummer des Produktions-Paket Link.
- (3) Das selektierte Schema des Produktions-Paket Link.
- (4) Das ausgewählte Produktions-Paket bzw. die Status Referenz auf ein Produktions-Paket.
- (5) Das **⋮** Kontextmenü des Produktions-Paket Link.

Um einen Produktions-Paket Link zu **+** **erstellen**, klicken Sie bitte auf [Abb. A \(1\)](#).

Um einen Produktions-Paket Link zu **ändern**, klicken Sie bitte auf den Namen des Produktions-Paket Links (2).

In beiden Fällen öffnet sich folgende Ansicht:

Produktions-Paket Link

Name *	<input type="text" value="BRIEFE_TEST"/>			
Schema *	<input type="text" value="Briefe"/>			
Paket *	<input type="text" value="Test"/>			
Beschreibung	<table border="1"> <tr> <td><input type="button" value="Test"/></td> <td><input type="button" value="Produktion"/></td> <td><input type="button" value="Selektieren"/></td> </tr> </table>	<input type="button" value="Test"/>	<input type="button" value="Produktion"/>	<input type="button" value="Selektieren"/>
<input type="button" value="Test"/>	<input type="button" value="Produktion"/>	<input type="button" value="Selektieren"/>		
Erstellt / Geändert	<table border="1"> <tr> <td><input type="text" value="06.01.2021 10:36"/></td> <td><input type="text" value="06.01.2021 10:36"/></td> </tr> </table>	<input type="text" value="06.01.2021 10:36"/>	<input type="text" value="06.01.2021 10:36"/>	
<input type="text" value="06.01.2021 10:36"/>	<input type="text" value="06.01.2021 10:36"/>			

Abbildung B

**Sie können der Verknüpfung:**

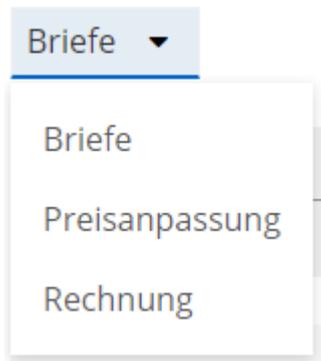
- ★ einen **Namen** geben *Abb. B (1)*.

Mit Hilfe des Namens kann das Paket einem Datenstrom zugeordnet werden. Der Name muss mit dem im Datenstrom enthaltenen Namen übereinstimmen, den das Produktions-Paket verwendet. Lesen Sie dazu den Abschnitt über die **Umgebung des R-S** und die **Produktions-Paket Verknüpfungen**.

⊖ Die beiden Namen müssen unbedingt übereinstimmen, da sonst eine Zuordnung von Datenströmen und Dokument nicht möglich ist!

⚠ Die Eingabe-Feld-Validierung erfolgt nach **STRING**.

- ★ Der Verknüpfung ein **Schema** zuordnen. Klicken Sie auf (2) . Es klappt ein Auswahlménü aus:



*Wählen Sie ein Schema aus.*

In der Auswahl werden Ihnen alle verfügbaren Schemas angezeigt. Da von jedem Schema immer nur ein produktionsbereites Produktions-Paket existieren kann, ist mit Auswahl des Schemas das Paket eindeutig bestimmt. Sie können allerdings das Produktions-Paket auch manuell auswählen.

- ★ ein **Produktions-Paket** selektieren, auf Produktion setzen und testen.

Wenn Sie ein Produktions-Paket manuell auswählen möchten, können Sie das mit (6) tun. Hier ist es auch möglich, Produktions-Pakete mit DEPRECATED-Status auszuwählen.

Sie haben dann folgende Möglichkeiten:

- **Produktion** (5): Das Produktions-Paket im PRODUCTION-Status des selektierten Schemas wird für diesen Link verwendet.

Das Paket ist damit für die Produktion freigegeben und kann im Produktions-Prozess genutzt werden.

- **Test** (4): Das Produktions-Paket im TEST-Status des selektierten Schemas wird für diesen Link verwendet.

Das Paket muss dazu auch nicht PRODUCTION-Status haben. Die Dokumente können damit einen Test-Produktions-Prozess durchlaufen. Eventuell müssen Sie sich dazu mit Ihrem Druckanbieter absprechen.

- ★ eine **Beschreibung** geben (7). Sie wird z.B. in Auswahldialogen angezeigt.

Verwenden Sie zum Speichern des Produktions-Paket Links Speichern oder Speichern und Schließen.  
Um die Änderungen zu verwerfen, klicken Sie bitte auf Zurücksetzen oder schließen Sie den Tab.

### **Link löschen**

Um einen Produktions-Paket Link zu löschen, öffnen Sie dessen  Kontextmenü *Abb. A (5)* und wählen Sie anschließend  Löschen (6) . Folgender Dialog öffnet sich:



Abbildung C

Klicken Sie auf Ok, um das Löschen zu bestätigen, oder auf Cancel, um das Löschen abubrechen.

 Wenn ein Vertrag gelöscht wird, ist das zugehörige Produktions-Paket nicht mehr durch die Vertragsnummer ansprechbar und damit auch im Produktions-Prozess nicht mehr verfügbar.

## Mehr zur Arbeit mit Objekten

In diesem Kapitel sollen Ihnen noch einige erweiterte Konzepte im Umgang mit dem R-S und insbesondere R-S Objekten vorgestellt werden. Dazu gehören:

- Das *Löschen von Objekten*
- Das *Wechseln zwischen Bearbeitungs- und Anzeigemodus*
- *EVAL-Anweisungen, Operatoren und Datentypen* und *Funktionen*
- *INLINE-Sequenzen*

## Löschen von Objekten

Wählen Sie dazu in der jeweiligen Übersicht das entsprechende Objekt aus und öffnen Sie dessen Kontextmenü *Abb. A* (1) und wählen Sie anschließend  Löschen (2).

Der Vorgang gleicht sich für alle Objekte und soll hier kurz exemplarisch am Beispiel einer Schema-Version dargestellt werden:

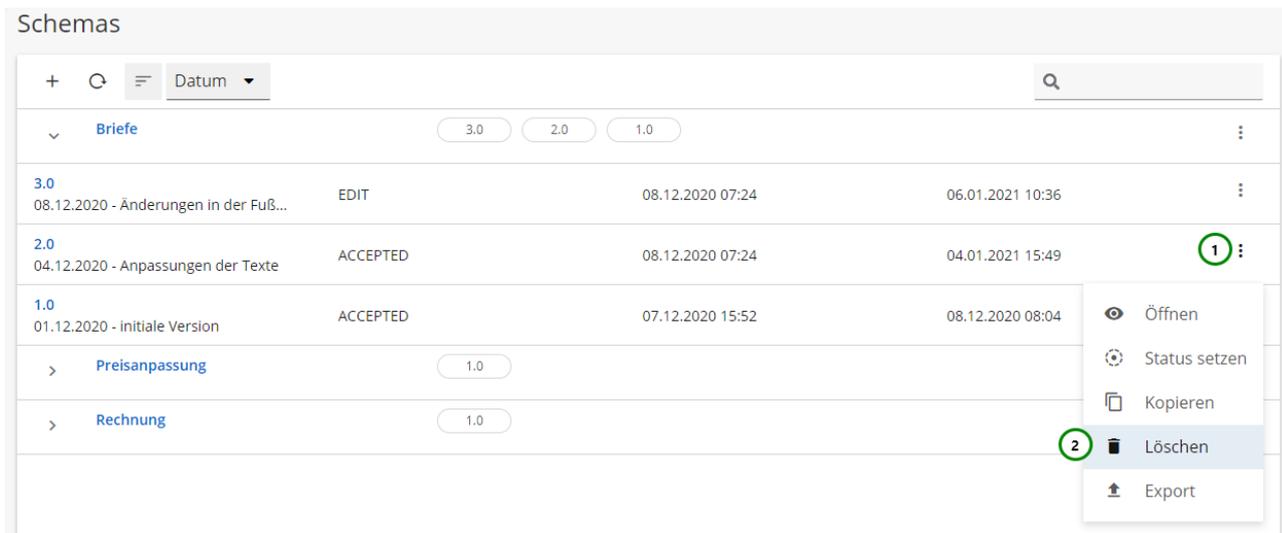


Abbildung A

Folgender Dialog erscheint:



Abbildung B

Bestätigen Sie mit Ok. Das Objekt wird jetzt nicht mehr in der Übersicht angezeigt.

 Sie können ein Objekt nur löschen, wenn es keine aktiven (existierenden) Versionen besitzt. Falls Sie versuchen, ein Transaktions-Formular, Schema oder Bibliothek mit aktiven Versionen zu löschen, erscheint folgender Fehler:



Abbildung C

Sie müssen dann zunächst alle Versionen des Formulars / der Bibliothek / des Schemas löschen, bevor Sie das Formular / die Bibliothek / das Schema selbst löschen.

⚠ Überprüfen Sie, ob eine Formular-Version in einer Schema-Version verwendet wird, bevor Sie diese löschen, da Sie beim Löschen nicht extra auf die Abhängigkeit hingewiesen werden. Die entsprechenden Schema-Version referenziert dann auf eine ungültige (da nicht mehr existierende Formular-Version), weshalb beim Erstellen eines Produktions-Paket (Statuswechsel zu **TEST**) folgende Fehlermeldung auftritt:

⚠ production-bundle has inactive txform-revision[{{txform-revision}}] -> can't create it

Abbildung D

Weisen Sie der Schema-Version ein neues Transaktions-Formular zu oder kontaktieren Sie ihren Administrator, damit dieser die gelöschte Formular-Version wiederherstellt.

⚠ Wenn Sie ein Objekt löschen (dazu zählen auch Bibliotheken) geht das entsprechende Objekt nicht endgültig verloren, sondern es wird "inaktiv" gesetzt und ist nicht mehr sichtbar / verwendbar (weder für den Nutzer noch für andere Objekte), ist aber physisch noch existent. Ein R-S Administrator kann gelöschte Objekte im R-S Admin-Client wiederherstellen. Wenden Sie sich an Ihren R-S Administrator, um das versehentlich gelöschte Objekte wiederherzustellen.

⚠ Inaktive Objekte werden nur eine bestimmte Zeit lang archiviert, bevor sie endgültig gelöscht werden. Danach ist eine Wiederherstellung des Objekts nicht mehr möglich.

ⓘ Informationen für Administratoren, wie Sie gelöschte Objekte wiederherstellen, finden Sie [hier](#).

## Wechseln des Modus

Wenn Sie eine Formular-, Schema- oder Bibliotheks-Version öffnen, befinden Sie sich zuerst im Anzeige-Modus (View-Modus). Hier können Sie alle Informationen der Version einsehen ohne die Bearbeitung anderer oder Status-Wechsel zu blockieren.

Um eine Version zu bearbeiten, müssen Sie den  Bearbeitungsmodus Abb. A (1) aktivieren:

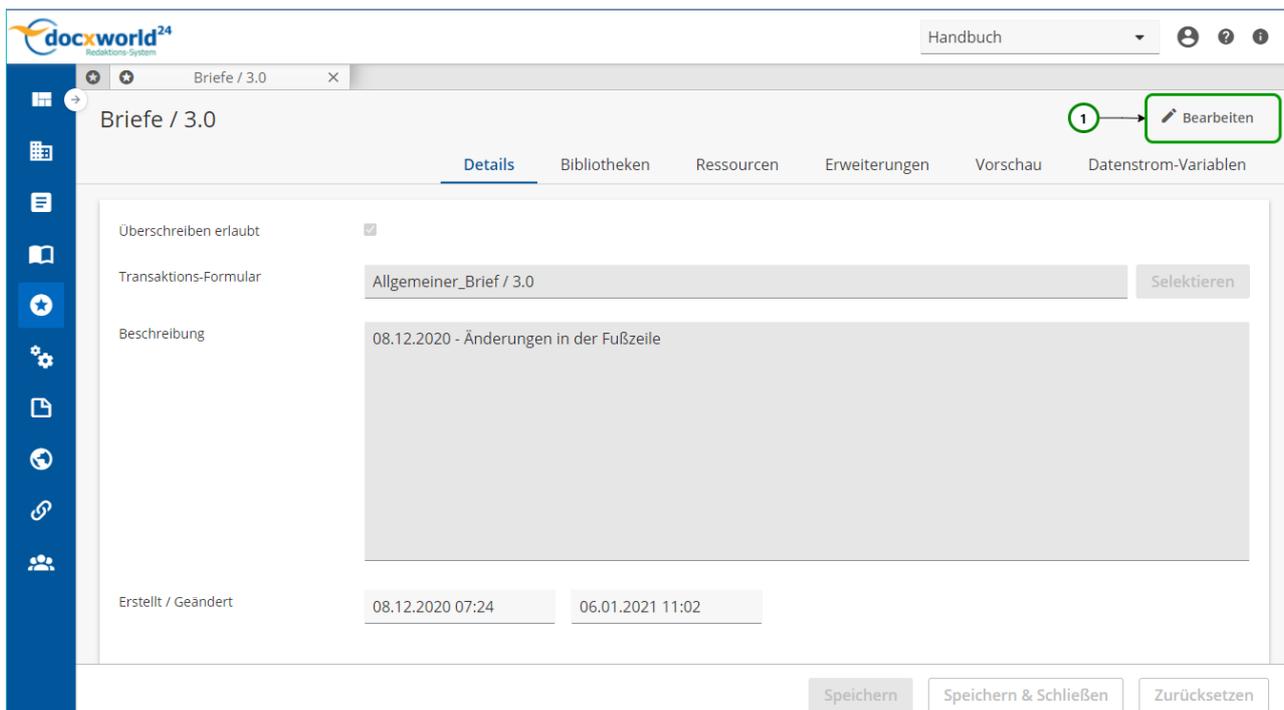


Abbildung A

Im Bearbeitungsmodus können Sie nun alle Inhalte der Version editieren. **Beachten Sie**, dass dadurch die Bearbeitung durch andere, sowie Statuswechsel gesperrt sind. Dies trifft auch auf Ihre eigenen Aktionen zu. Haben Sie z.B. noch die Version im Bearbeitungsmodus als Tab geöffnet, so können Sie auch keinen Statuswechsel vornehmen.

Sobald Sie im Bearbeitungsmodus sind, ändert sich Abb. A (1) zu:



Wenn der Bearbeitungsmodus nicht aktiviert werden kann, kann dies z.B. folgende Gründe haben:

- Das Objekt hat nicht den Status EDIT (nur Objekte mit Status EDIT können geändert werden),
- Sie haben keine Rechte zum Bearbeiten von Objekten dieses Typs,
- Ein anderer Nutzer bearbeitet gerade das Objekt.

 In Bearbeitung: Joey

 Um vom Bearbeitungs- in den Anzeigemodus zu wechseln, können Sie Zurücksetzen verwenden. Bitte Beachten Sie, dass hier alle nicht gespeicherten Änderungen verworfen werden.

## EVAL-Anweisungen, Operatoren und Datentypen

EVAL-Anweisungen (EVAL steht für Evaluierung) sind eine Möglichkeit, dem R-S dynamisch Befehle zur Berechnung und Einbindung von Objekten zu übergeben. Eine EVAL-Anweisung wird immer mit der Zeichenfolge `#{` eingeleitet und mit der Zeichenfolge `}` abgeschlossen.

Es gibt 2 Hauptgründe, um EVAL-Anweisungen zu verwenden:

### 1. Einbinden einer Ressource

Mit EVAL-Anweisungen können Sie Ressourcen ansprechen. Als Identifikator dient der Name der Ressource.

**i Syntax**  
`#{Meine_Ressource}`

Hier wird dann die Ressource "Meine Ressource" dynamisch verlinkt. Wenn Sie z.B. in einem RichText schreiben:

"Sehr geehrter Herr `#{DS.CUST_NAME}`"

wird, wenn der RichText verwendet wird, dynamisch die Datenstrom-Variable `DS.CUST_NAME` eingefügt und das Resultat ist z.B.:

"Sehr geehrter Herr Max Mustermann"

### 2. Funktionen und Bedingungen

Sie können innerhalb von EVAL-Anweisungen vorgefertigte Funktionen verwenden, die dann dynamisch einen Wert liefern. Eine Liste der vorhandenen Funktionen finden Sie [hier](#).

**i Syntax**  
`#{NameDerFunktion(Parameter1, Parameter2,...)}`

`#{IsNull(0)}` hat z.B. als Resultat **1**.

EVAL-Anweisungen führen also die eingebettete(n) Funktion(en) aus und liefern den Ergebniswert zurück.

## 1. Umgang mit Ressourcen in Texten

Wie oben beschrieben ist es möglich, mit EVAL-Anweisungen Ressourcen in Texte einzubinden. So ist es z.B. möglich, Variablen oder Regeln in andere Textressourcen einzubinden. Eine Ausnahme gilt für binäre Ressourcen. Da sie nicht zusammen mit anderem Text evaluiert werden können, ist es nur möglich, binäre Ressourcen in Variablen einzubinden, wenn außer der einbindenden EVAL-Anweisung sonst kein Text in der Variable enthalten ist. Ansonsten interpretiert das R-S den kompletten Inhalt als Text. Außerdem ist es grundsätzlich nicht möglich, binäre Ressourcen in RichTexte einzubinden.

## 2. Operatoren

Das R-S unterstützt auch boolesche Operationen, die Sie z.B. in einer Regel anwenden können. Boolesche Operationen sind Vergleiche, die entweder wahr oder falsch sind. Folgende Operatoren sind im R-S definiert:

Operator	Bedeutung	Beschreibung
*	ALWAYSTRUE	Ist immer wahr.
!=	NOTEQUAL	Ist wahr, wenn die Objekte nicht gleich sind.
<=	EQUALLESS	Ist wahr, wenn Objekt1 kleiner als Objekt2 ist oder die Objekte gleich sind.
>=	EQUALGREATER	Ist wahr, wenn Objekt1 größer als Objekt2 ist oder die Objekte gleich sind.
<	LESS	Ist wahr, wenn Objekt1 kleiner als Objekt2 ist.
>	GREATER	Ist wahr, wenn Objekt1 größer als Objekt2 ist.
[VAL1,VAL2,VALn]	INLIST	Ist wahr, wenn das zu vergleichende Objekt in der Liste vorhanden ist.
[VAL1,...,VAL2]	<b>INRANGE</b> (3 Argumente, das mittlere Argument ist "...")	Ist wahr, wenn das zu vergleichende Objekt zwischen <b>VAL1</b> und <b>VAL2</b> liegt.
![VAL1, VAL2, VALn]	NOTINLIST	Ist wahr, wenn das zu vergleichende Objekt <u>nicht</u> in der Liste vorhanden ist.
![VAL1,...,VAL2]	NOTINRANGE	Ist wahr, wenn das zu vergleichende Objekt <u>nicht</u> zwischen <b>VAL1</b> und <b>VAL2</b> liegt.

Wird kein Operator angegeben, wird EQUAL (=) verwendet. Der Vergleich ist also dann wahr, wenn die Objekte gleich sind. Alle Operatoren wurden für alle Datentypen umgesetzt. Praktische Beispiele finden Sie in der [Regeldefinition](#).

### 3. Datentypen

Das R-S unterstützt verschiedene Datentypen:

Datentyp	Beschreibung	Wertebereich	Beispiel
STRING	Speichert eine Zeichenkette	Alle Unicode-Zeichen	"Hallo Welt!"
NUMBER	Speichert eine Ganzzahl	-2.147.483.648 bis +2.147.483.648 bzw. 0 bis 4.294.967.295	+706512, -59102
REAL	Speichert eine Gleitkommazahl Trennzeichen ist '.' (Achtung keine localisierten Schreibweisen erlaubt und damit auch kein 1000-der Punkt)	3.4E-4932 1.1E+4932	+5481.4390156, -98042.4168640
BOOLEAN	Speichert einen Wahrheitswert (0/1)	0/false/no und 1/true/yes	true, false, 1, 0, yes, no
DATETIME	Speichert ein Datum nach ISO 8601	Datum ohne Unterstützung von Wochen, Wochentagen und Tagen im Jahr	2011-12-31, 2011-12-31T16:47Z

#### 4. Entities

Es werden alle HTML-Entities aufgelöst, die RW kennt. Zusätzliche ESCAPE-Zeichen sind:

ESCAPE-Sequenz	Bedeutung
\"	Doppeltes Anführungszeichen
\n	Zeilenvorschub (neue Zeile)

Da in RichTexten die interne Darstellung durch XML erfolgt, werden XML-Sonderzeichen im Text implizit umgewandelt, damit sie innerhalb des Codes gespeichert werden können. Sie können mit den folgenden Escape-Sequenzen aber auch eine explizite Umwandlung erzwingen:

ESCAPE-Sequenz	Stellt dar
\!	!
\=	=
\<	<
\>	>
\[	[
\]	]



EVAL-Anweisungen werden **NICHT** durch das Redaktions-System auf syntaktische und semantische Korrektheit geprüft! Fehlt zum Beispiel eine schließende Klammer oder ist eine Funktionsbezeichnung falsch geschrieben, so wird dies erst durch die Output Management Produkte geprüft, welche die Funktionen bei der Dokument-Generierung nutzen (Late Binding). So können diese Art von Fehlern erst bei der Ausführung der Preview-Funktion im Redaktions-System evaluiert werden.

## Funktionen in EVAL-Anweisungen

Wie bereits vorgestellt können innerhalb von EVAL-Anweisungen auch Funktionen angegeben werden. Dazu gibt es im R-S einen Satz vorgefertigter Funktionen, die Sie nutzen können.

Diese Funktionen sind syntaktisch wie folgt aufgebaut:

```
FUNKTIONSNAME (PN1, . . . , PNn)
```

Der FUNKTIONSNAME bezeichnet die Funktion. Der FUNKTIONSNAME kann in Klein- und Großbuchstaben oder gemischt in Klein-/Großbuchstaben geschrieben werden, eine Unterscheidung ergibt sich dadurch nicht. Nach dem FUNKTIONSNAMEN folgen dann die Zeichenfolge '()'. Innerhalb des Klammer-Paares können die Übergabeparameter der Funktion angegeben werden. **PN1** bis **PNn** sind die Übergabeparameter der Funktion. Parameter werden immer mit " " eingeschlossen. Die Anzahl der Übergabeparameter ist von der Funktion abhängig. Es müssen immer alle erwarteten Parameter angegeben werden, manche Funktionen haben aber auch optionale Parameter. Übergabeparameter werden durch ';' getrennt und können folgende Werte annehmen:

- **Statische Werte** (Statische Werte werden durch doppelte Anführungszeichen eingeschlossen):  
*Beispiel:* `concat("Hello", "World")`
- **Variablen** aus dem Datenstrom oder dem R-S:  
*Beispiel:* `concat("Betrag von ", DS.RECHNUNGS_SUMME)`
- **Funktionen**  
*Beispiel:* `concat("Zahlen Sie bis zum: ", AddDate(DS.RECHNUNGS_DATUM, "30", "NO"))`

Die meisten Funktionen erwarten Parameter von einem bestimmten Datentyp und geben einen Wert in einem bestimmten Datentyp zurück. Manche Funktionen existieren für mehrere Datentypen, z.B. Add für **INTEGER** (AddI), **REAL** (AddR) und **DATE** (AddDate). Sie müssen jeweils die Funktion verwenden, die für den vorhandenen Datentyp geeignet ist.

### **Kurz-Übersichtsliste der Funktionen**

Im Folgenden sind alle Funktionen aufgeführt und nach Typ geordnet. Von jeder Funktion werden die verschiedenen Ausführungen aufgelistet, die unterschiedliche Datentypen als Eingangsparameter verarbeiten können. Klicken Sie auf eine Funktion, um genauere Informationen zu erhalten.

### Einfache mathematische Funktionen

Funktion	INTEGER	REAL	DATE	Beschreibung
Add	AddI(PN1,...,PNn)	AddR(PN1,...,PNn)	AddDate(PN1,PN2,PN3)	Addiert 2 oder mehrere Werte bzw. addiert Tage zu einem gegebenen Datum.
Subtrac	<i>Subtracl(PN1,PN2,...,PNn)</i>	<i>SubtracR(PN1,PN2,...,PNn)</i>	<i>SubtDate(PN1,PN2,PN3)</i>	Subtrahiert 2 oder mehrere Werte bzw. subtrahiert Tage von einem gegebenen Datum.
Multiply	<i>MultiplyI(PN1,PN2,...,PNn)</i>	<i>MultiplyR(PN1,PN2,PN3,PN4)</i>	—	Multipliziert PN1 mit allen folgenden Parametern.
Divide	<i>Dividel(PN1,PN2)</i>	<i>DivideR(PN1,PN2)</i>	—	Teilt die Zahl von PN1 (Dividend) durch die Zahl von PN2 (Divisor).
Round	—	<i>RoundR(PN1,PN2,PN3,PN4)</i>	—	Rundet eine reelle Zahl mit Nachkommastellen

## Stringoperatoren

Funktion	STRING	Beschreibung
Empty	<i>EmptyS(PN1)</i>	Gibt 1 (true) zurück, wenn der String PN1 leer ist, ansonsten 0.
Index	<i>IndexS(PN1,PN2,PN3)</i>	Sucht in einem String das Vorkommen eines Matchstrings und gibt die Position des ersten Zeichens des Matchstrings im String zurück.
Length	<i>LengthS(PN)</i>	Gibt die Länge des Strings zurück.
Sub	<i>SubS(PN1,PN2,PN3)</i>	Bildet vom String PN1 einen SubString, der auf Position PN2 beginnt und PN3 Zeichen lang ist.
Token	<i>TokenS(PN1,PN2,PN3)</i>	Liefert den n-ten Token aus einem String zurück, wobei 1 der erste Token ist.

## Stringmanipulation

Funktion	STRING	Beschreibung
Clear	ClearS(PN1)	Der angegebene String wird von allen Zeichen bereinigt, die in einem File-Namen nicht vorkommen dürfen.
Fill	<i>FillS(PN1,PN2,PN3,PN4)</i>	Füllt einen String mit einem angegebenen Zeichen auf.
Placeholder	<i>PlaceholderS(PN1,...,PNn)</i>	Tauscht im String PN1 alle %x% (x = fortlaufende Nummer) durch PNx aus.
RemoveAll	<i>RemoveAllS(PN1,PN2)</i>	Entfernt jedes Vorkommen der Zeichenkette PN2 im String PN1.
ReplaceAll	<i>ReplaceAllS(PN1,PN2,PN3)</i>	Tauscht jedes Vorkommen der Zeichenkette PN2 im String PN1 gegen PN3.
ToLower	<i>ToLowerS (PN1)</i>	Wandelt die Großbuchstaben eines Strings in Kleinbuchstaben.
ToUpper	<i>ToUpperS (PN1)</i>	Wandelt die Kleinbuchstaben eines Strings in Großbuchstaben.
Trim	<i>TrimS(PN1,PN2,PN3)</i>	Entfernt nicht benötigte Zeichen vom Anfang und oder vom Ende eines Strings.

### Logische Vergleiche

Funktion	STRING	INTEGER	REAL	Beschreibung
IsLess	IsLessS(PN1,PN2)	IsLessI(PN1,PN2)	IsLessR(PN1,PN2)	Liefert 1 (true) zurück, wenn PN1 kleiner ist als PN2, ansonsten 0 (false)
IsEqual	<i>IsEqualS(PN1,PN2)</i>	<i>IsEqualI(PN1,PN2)</i>	<i>IsEqualR(PN1,PN2)</i>	Liefert 1 (true) zurück, wenn PN1 gleich PN2 ist, ansonsten 0 (false)
IsGreater	<i>IsGreaterS(PN1,PN2)</i>	<i>IsGreaterI(PN1,PN2)</i>	<i>IsGreaterR(PN1,PN2)</i>	Liefert 1 (true) zurück, wenn PN1 größer ist als PN2, ansonsten 0 (false)
IsNull	<i>IsNullS(PN1)</i>	<i>IsNullI(PN1)</i>	<i>IsNullR(PN1)</i>	Liefert 1 (true) zurück, wenn PN1 0 ist (bzw. ein leerer String), ansonsten 0 (false)

### Logische Operatoren

Funktion	BOOLEAN	Beschreibung
And	<i>And(PN1,...,PNn)</i>	Verknüpft alle PNs nach den Regeln des Logischen And. Gibt 1 zurück, wenn alle PN = true
If	<i>If(PN1,PN2,PN3)</i>	Liefert den Wert PN2 zurück, wenn PN1 gleich 1 (true) ist, ansonsten PN3.
Nand	<i>Nand(PN1,...,PNn)</i>	Verknüpft alle PNs nach den Regeln des Logischen Nand. Gibt 1 zurück, wenn min. 1 PN = true
Nor	<i>Nor(PN1,...,PNn)</i>	Verknüpft alle PNs nach den Regeln des Logischen Nor. Gibt 1 zurück, wenn alle PN = false
Not	<i>Not(PN1)</i>	Negiert PN1.
Or	<i>Or(PN1,...,PNn)</i>	Verknüpft alle PNs nach den Regeln des Logischen Or. Gibt 1 zurück, wenn min 1 PN = false

### Sonstige Funktionen

Funktion	STRING	INTEGER / REAL	Beschreibung
Concat	<i>ConcatS(PN1,PN2,...,PNn)</i>	—	Hängt die Werte aller Parameter aneinander und gibt das Resultat als String zurück.
Now	<i>NowS(format-string)</i>	—	Wird zum Einfügen von Datums- und Zeitangaben verwendet.
SepSign	—	<i>SepSign(PN1,PN2)</i>	Trennt bei einer Zahl mit Vorzeichen die Zahl und das Vorzeichen voneinander.

## AddDate

### **Funktionsbeschreibung**

Die AddDate-Funktion addiert zu einem als Input gegebenen Tagesdatum weitere Tage hinzu und gibt als Ausgabe das errechnete Datum zurück.  
Dabei können die Wochenenden ignoriert werden, so dass nur die Werktage gezählt werden.

### **Syntax**

AddDate(PN1,PN2,PN3)

### **Parameter (3)**

DATE PN1: InputDatum im Format DD.MM.YYYY oder DD.MM.YY

INTEGER PN2: Anzahl Tage, die addiert werden sollen

STRING PN3: Wochenende-Schalter mit YES/NO oder 1/0; bei YES/1 werden Tage an Wochenenden mitgezählt

### **Rückgabewert**

DATE

### **Beispiel**

```
AddDate("12.07.2002", "3", "NO")
```

**Ergebnis:** 17.07.2002

## AddI

### **Funktionsbeschreibung**

AddI addiert die Integerwerte aller resultierenden Strings zueinander.

### **Syntax**

AddI(PN1,...,PNn)

### **Parameter (beliebig viele, min. 2)**

INTEGER PN1: Wert 1

INTEGER PN2: Wert 2

INTEGER PNn: IWert n

### **Rückgabewert**

INTEGER

### **Beispiel**

```
AddI("12","3")
```

**Ergebnis:** 15

## AddR

### Funktionsbeschreibung

AddR addiert reelle Zahlen (Gleitkommazahlen) zueinander.

### Syntax

AddR(PN1,PN2,PN3,PN4)

### Parameter (4)

REAL PN1: 1. Summand

REAL PN2: 2. Summand

STRING PN3: Zeichen für die Trennung von ganzem und gebrochenem Teil (optional)

STRING PN4: Zeichen für die Tausender-Trennung (optional)

 Das Zahlenformat des Ergebnisses der Funktion ergibt sich aus den Format-Einstellungen von PN1.

Sind PN3 und ON4 nicht angegeben, so wird das Ergebnis im deutschen Zahlenformat ausgegeben.

 Bitte beachten Sie, dass die Default-Einstellung zur Interpretation des Formates für PN1 und PN2 das deutsche Zahlenformat ist (PN3="," und PN4=".").

Verarbeiten Sie andere Zahlenformate, so müssen die Parameter PN3 und PN4 unbedingt angegeben werden.

### Rückgabewert

REAL

### Beispiel

```
AddR("1200,00", "1300,00", ",", ".")
```

**Ergebnis:** 2.500,00

## And

### **Funktionsbeschreibung**

Verknüpft alle PN nach den Regeln des logischen And. Der Rückgabewert ist 1, wenn alle PN den booleschen Wert 1 haben, andernfalls ist der Rückgabewert 0 (*Umkehr: **Nor***).

### **Syntax**

And(PN1,...,PNn)

### **Parameter (beliebig viele, min. 2)**

BOOLEAN PN1: Wert 1

BOOLEAN PN2: Wert 2

BOOLEAN PNn: Wert n

### **Rückgabewert**

BOOLEAN

### **Beispiel**

```
And("1", "1", "1")
```

```
And("1", "0", "1")
```

**Ergebnis 1:** 1

**Ergebnis 2:** 0

## ClearS

### **Funktionsbeschreibung**

Der angegebene String wird von allen Leerzeichen, Zeilenumbrüchen, Tabulatoren und allen Zeichen bereinigt, die in einem File-Namen nicht vorkommen dürfen.

### **Syntax**

ClearS(PN1)

### **Parameter (1)**

STRING PN1: Zeichenkette

### **Rückgabewert**

STRING

### **Beispiel**

```
ClearS("Die Nadel im Heuhaufen")
```

**Ergebnis:** DieNadelimHeuhaufen

## ConcatS

### **Funktionsbeschreibung**

ConcatS ist eine Funktion, welche die Werte aller Platzhalter aneinander hängt und den Ergebniswert zurückgibt.

### **Syntax**

ConcatS(PN1, PN2,..., PNn)

### **Parameter (beliebig viele, min. 2)**

STRING PN1: Zeichenkette 1

STRING PN2: Zeichenkette 2

STRING PNn: Zeichenkette 3

### **Rückgabewert**

STRING

### **Beispiel**

```
ConcatS("Hallo", "Welt", "!")
```

**Ergebnis:** Hallo Welt !

## Dividel

### **Funktionsbeschreibung**

Dividel teilt die Integerzahl von PN1 (Dividend) durch die Integerzahl von PN2 (Divisor).

### **Syntax**

Dividel(PN1,PN2)

### **Parameter (2)**

INTEGER PN1: Wert 1

INTEGER PN2: Wert 2

### **Rückgabewert**

INTEGER

### **Beispiel**

```
DivideI ("12","3")
```

**Ergebnis:** 4

## DivideR

### Funktionsbeschreibung

DivideR teilt die reelle Zahl von PN1 durch die reelle Zahl von PN2.

### Syntax

DivideR(PN1, PN2, PN3, PN4)

### Parameter (4, min. 2)

REAL PN1: Dividend

REAL PN2: Divisor

STRING PN3: Zeichen für die Trennung von ganzem und gebrochenem Teil (optional)

STRING PN4: Zeichen für die Tausender-Trennung (optional)

 Das Zahlenformat des Ergebnisses der Funktion ergibt sich aus den Format-Einstellungen von PN1.

Sind PN3 und PN4 nicht angegeben, so wird das Ergebnis im deutschen Zahlenformat ausgegeben.

 Bitte beachten Sie, dass die Default-Einstellung zur Interpretation des Formates für PN1 und PN2 das deutsche Zahlenformat ist (PN3="," und PN4=".").

Verarbeiten Sie andere Zahlenformate, so müssen die Parameter PN3 und PN4 unbedingt angegeben werden.

### Rückgabewert

REAL

### Beispiel

```
DivideR("1500,00","300,00","",".")
```

**Ergebnis:** 5,00

## EmptyS

### **Funktionsbeschreibung**

EmptyS gibt an, ob Zeichen in dem String enthalten sind.

### **Syntax**

EmptyS(PN1)

### **Parameter (1)**

STRING PN1 = Zeichenkette

### **Rückgabewert**

BOOLEAN

### **Beispiel**

1. EmptyS("Hallo")
2. EmptyS("")

**Ergebnis 1:** 0 (false)

**Ergebnis 2:** 1 (true)

## FILLS

### **Funktionsbeschreibung**

FILLS ist ein Stringfüller. Strings, die unter der angegebenen Länge sind, werden mit dem ebenfalls angegebenen Füllzeichen gefüllt.

Dabei kann das Füllzeichen entweder links oder rechts angesetzt werden. Wie bei der Standardsyntax kann jeder PN-Wert als Variable, FixText oder Funktion angegeben werden.

### **Syntax**

FILLS(PN1,PN2,PN3,PN4)

### **Parameter (4)**

INTEGER,NUMBER,STRING PN1: Ist der zu füllende Wert

STRING PN2: ist das Füllzeichen (das erste Zeichen des Strings zählt)

STRING PN3: Orientierung des Befüllens (LEFT oder RIGHT)

INTEGER,NUMBER PN4: die Sollgröße des Wertes

### **Rückgabewert**

STRING

### **Beispiel**

```
FILLS("Hallo", "!", "RIGHT", "8")
```

**Ergebnis:** Hallo!!!

## If

### **Funktionsbeschreibung**

Wenn PN1 den Wert 1 zurück liefert (true), dann ist der Rückgabewert von If gleich dem Wert PN2.

Wenn PN1 den Wert 0 zurück liefert (false), dann ist der Rückgabewert von If gleich dem Wert PN3.

### **Syntax**

If(PN1,PN2,PN3)

### **Parameter (3)**

*BOOLEAN* PN1: Bedingung eingeben

*ANY DATATYPE* PN2: Rückgabewert 1

*ANY DATATYPE* PN3: Rückgabewert 2

### **Rückgabewert**

UNDEFINED (ANY DATATYPE)

### **Beispiel**

```
If(EmptyS(""), "Der String ist leer", "Der String ist nicht leer")
```

**Ergebnis:** Der String ist leer

## IndexS

### **Funktionsbeschreibung**

IndexS sucht in einem String das Vorkommen eines Matchstrings und gibt die Position des ersten Zeichens des Matchstrings im String zurück.  
Dabei bedeutet -1 nicht gefunden und 0 ist das erste Zeichen.

### **Syntax**

IndexS(PN1,PN2)

### **Parameter (2)**

STRING PN1: Zeichenkette

STRING PN2: Zu findende Zeichenkette in PN1

### **Rückgabewert**

INTEGER

### **Beispiel**

1. IndexS("Die Nadel im Heuhaufen","Geld")
2. IndexS("Die Nadel im Heuhaufen","Nadel")

**Ergebnis 1:** -1

**Ergebnis 2:** 4

## IsEqualI

### **Funktionsbeschreibung**

Liefert den Wert 1 zurück, wenn die ganze Zahl PN1 gleich der ganzen Zahl PN2 ist. Andernfalls ist der Rückgabewert 0.

### **Syntax**

IsEqualI(PN1,PN2)

### **Parameter (2)**

INTEGER PN1: Wert 1

INTEGER PN2: Wert 2

### **Rückgabewert**

BOOLEAN

### **Beispiel**

```
IsEqualI("1","1")
```

```
IsEqualI("1","0")
```

**Ergebnis 1:** 1

**Ergebnis 2:** 0

## IsEqualR

### Funktionsbeschreibung

Liefert den Wert 1 zurück, wenn die reelle Zahl PN1 gleich der reellen Zahl PN2 ist. Andernfalls ist der Rückgabewert 0.

### Syntax

IsEqualR(PN1,PN2)

### Parameter (2)

REAL PN1: Wert 1

REAL PN2: Wert 2

### Rückgabewert

BOOLEAN

### Beispiel

```
IsEqualR("1,5", "1,5")
```

```
IsEqualR("1,5", "1,0")
```

**Ergebnis 1:** 1

**Ergebnis 2:** 0

## IsEquals

### **Funktionsbeschreibung**

Liefert den Wert 1 zurück, wenn der String PN1 gleich dem String PN2 ist. Andernfalls ist der Rückgabewert 0.

### **Syntax**

IsEquals(PN1,PN2)

### **Parameter (2)**

STRING PN1: Zeichenkette 1

STRING PN2: Zeichenkette 2

### **Rückgabewert**

BOOLEAN

### **Beispiel**

```
IsEquals("HALLO","HALLO")
```

```
IsEquals("HALLO","TSCHUESS")
```

**Ergebnis 1:** 1

**Ergebnis 2:** 0

## IsGreaterI

### **Funktionsbeschreibung**

Liefert den Wert 1 zurück, wenn die ganze Zahl PN1 größer ist als die ganze Zahl PN2. Andernfalls ist der Rückgabewert 0.

### **Syntax**

IsGreaterI(PN1,PN2)

### **Parameter (2)**

INTEGER PN1: Wert 1

INTEGER PN2: Wert 2

### **Rückgabewert**

BOOLEAN

### **Beispiel**

```
IsGreaterI("7","5")
```

```
IsGreaterI("7","9")
```

**Ergebnis 1:** 1

**Ergebnis 2:** 0

## IsGreaterR

### **Funktionsbeschreibung**

Liefert den Wert 1 zurück, wenn die reelle Zahl PN1 größer ist als die reelle Zahl PN2. Andernfalls ist der Rückgabewert 0.

### **Syntax**

IsGreaterR(PN1,PN2)

### **Parameter (2)**

REAL PN1: Wert 1

REAL PN2: Wert 2

### **Rückgabewert**

BOOLEAN

### **Beispiel**

```
IsGreaterR("7,00","5,50")
```

```
IsGreaterR("7,25","7,75")
```

**Ergebnis 1:** 1

**Ergebnis 2:** 0

## IsGreaterS

### **Funktionsbeschreibung**

Liefert den Wert 1 zurück, wenn der String PN1 größer ist als der String PN2. Andernfalls ist der Rückgabewert 0.

### **Syntax**

IsGreaterS(PN1,PN2)

### **Parameter (2)**

STRING PN1: Wert 1

STRING PN2: Wert 2

### **Rückgabewert**

BOOLEAN

### **Beispiel**

```
IsGreaterS("A", "B")
```

```
IsGreaterS("B", "A")
```

**Ergebnis 1:** 1

**Ergebnis 2:** 0

## IsLessI

### **Funktionsbeschreibung**

Liefert den Wert 1 zurück, wenn die ganze Zahl PN1 kleiner ist als die ganze Zahl PN2. Andernfalls ist der Rückgabewert 0.

### **Syntax**

IsLessI(PN1,PN2)

### **Parameter (2)**

INTEGER PN1: Wert 1

INTEGER PN2: Wert 2

### **Rückgabewert**

BOOLEAN

### **Beispiel**

```
IsLessI("0", "1")
```

```
IsLessI("1", "0")
```

**Ergebnis 1:** 1

**Ergebnis 2:** 0

## IsLessR

### **Funktionsbeschreibung**

Liefert den Wert 1 zurück, wenn die reelle Zahl PN1 kleiner ist als die reelle Zahl PN2. Andernfalls ist der Rückgabewert 0.

### **Syntax**

IsLessR(PN1,PN2)

### **Parameter (2)**

REAL PN1: Wert 1

REAL PN2: Wert 2

### **Rückgabewert**

BOOLEAN

### **Beispiel**

```
IsLessR("1,0" , "1,25")
```

```
IsLessR("1,5" , "1,0")
```

**Ergebnis 1:** 1

**Ergebnis 2:** 0

## IsLessS

### **Funktionsbeschreibung**

Liefert den Wert 1 zurück, wenn der String PN1 kleiner ist als der String PN2. Andernfalls ist der Rückgabewert 0.

### **Syntax**

IsLessS(PN1,PN2)

### **Parameter (2)**

STRING PN1: Wert 1

STRING PN2: Wert 2

### **Rückgabewert**

BOOLEAN

### **Beispiel**

```
IsLessS("B", "A")
```

```
IsLessS("A", "B")
```

**Ergebnis 1:** 1

**Ergebnis 2:** 0

## IsNull

### **Funktionsbeschreibung**

Liefert den Wert 1 zurück, wenn die ganze Zahl PN1 0 ist. Andernfalls ist der Rückgabewert 0.

### **Syntax**

IsNull(PN1)

### **Parameter (1)**

INTEGER PN1: Wert

### **Rückgabewert**

BOOLEAN

### **Beispiel**

```
IsNull("0")
```

```
IsNull("1")
```

**Ergebnis 1:** 1

**Ergebnis 2:** 0

## IsNullR

### **Funktionsbeschreibung**

Liefert den Wert 1 zurück, wenn die reelle Zahl PN1 0 ist. Andernfalls ist der Rückgabewert 0.

### **Syntax**

IsNullR(PN1)

### **Parameter (1)**

REAL PN1: Wert

### **Rückgabewert**

BOOLEAN

### **Beispiel**

```
IsNullR("0,00")
```

```
IsNullR("0,01")
```

**Ergebnis 1:** 1

**Ergebnis 2:** 0

## IsNullS

### **Funktionsbeschreibung**

Liefert den Wert 1 zurück, wenn der String PN1 keine Zeichen enthält. Andernfalls ist der Rückgabewert 0 (Äquivalent zu **EmptyS**).

### **Syntax**

IsNullS(PN1)

### **Parameter (1)**

STRING PN1: Wert

### **Rückgabewert**

BOOLEAN

### **Beispiel**

```
IsNullS("")
```

```
IsNullS("Hallo")
```

**Ergebnis 1:** 1

**Ergebnis 2:** 0

## LengthS

### **Funktionsbeschreibung**

Gibt die Länge eines Strings zurück.

### **Syntax**

LengthS(PN1)

### **Parameter (1)**

STRING PN1: Wert

### **Rückgabewert**

INTEGER

### **Beispiel**

```
LengthS("Hallo Welt!")
```

**Ergebnis:** 11

## MultipliyI

### **Funktionsbeschreibung**

MultipliyI multipliziert die Integerwerte der Parameter 2 bis n mit dem Parameter 1.

### **Syntax**

MultipliyI(PN1,PN2,...,PNn)

### **Parameter (beliebig viele, min. 2)**

INTEGER PN1: Wert 1

INTEGER PN2: Wert 2

INTEGER PNn: Wert n

### **Rückgabewert**

INTEGER

### **Beispiel**

```
MultipliyI("2", "4", "5")
```

**Ergebnis:** 40

## MultiplyR

### Funktionsbeschreibung

MultiplyR multipliziert zwei reelle Zahlen (Gleitkommazahlen) zueinander.

### Syntax

MultiplyR(PN1,PN2,PN3,PN4)

### Parameter (4, min. 2)

REAL PN1: 1. Faktor

REAL PN2: 2. Faktor

STRING PN3: Zeichen für die Trennung von ganzem und gebrochenem Teil (optional)

STRING PN4: Zeichen für die Tausender-Trennung (optional)

 Das Zahlenformat des Ergebnisses der Funktion ergibt sich aus den Format-Einstellungen von PN1.

Sind PN3 und PN4 nicht angegeben, so wird das Ergebnis im deutschen Zahlenformat ausgegeben.

 Bitte beachten Sie, dass die Default-Einstellung zur Interpretation des Formates für PN1 und PN2 das deutsche Zahlenformat ist (PN3="," und PN4=".").

Verarbeiten Sie andere Zahlenformate, so müssen die Parameter PN3 und PN4 unbedingt angegeben werden

### Rückgabewert

REAL

### Beispiel

```
MultiplyR("2,5","5,0","",".")
```

**Ergebnis:** 12,5

## Nand

### **Funktionsbeschreibung**

Verknüpft alle PN nach den Regeln des logischen Nand. Der Rückgabewert ist 1, wenn mindestens ein PN den booleschen Wert 0 hat, andernfalls ist der Rückgabewert 0 (*Umkehr von **Nor***)

### **Syntax**

Nand(PN1,...,PNn)

### **Parameter (beliebig viele, min. 2)**

BOOLEAN PN1: Wert 1

BOOLEAN PN2: Wert 2

BOOLEAN PNn: Wert n

### **Rückgabewert**

BOOLEAN

### **Beispiel**

```
Nand("0", "1", "0")
```

```
Nand("1", "1", "1")
```

**Ergebnis 1:** 1

**Ergebnis 2:** 0

## Nor

### **Funktionsbeschreibung**

Verknüpft alle PN nach den Regeln des logischen Nor. Der Rückgabewert ist 1, wenn alle PN den boolschen Wert 0 haben, andernfalls ist der Rückgabewert 0 (*Umkehr von Or*)

### **Syntax**

Nor(PN1,...,PNn)

### **Parameter (beliebig viele, min. 2)**

BOOLEAN PN1: Wert 1

BOOLEAN PN2: Wert 2

BOOLEAN PNn: Wert n

### **Rückgabewert**

BOOLEAN

### **Beispiel**

```
Nor("0", "0", "0")
```

```
Nor("0", "0", "1")
```

**Ergebnis 1:** 1

**Ergebnis 2:** 0

# Not

## **Funktionsbeschreibung**

Negiert das Ergebnis von PN1.

## **Syntax**

Not(PN1)

## **Parameter (1)**

BOOLEAN PN1: Wert 1

## **Rückgabewert**

BOOLEAN

## **Beispiel**

```
Not(IsNullI("0"))
```

**Ergebnis:** 0

# NowS

## Funktionsbeschreibung

Zum Einfügen von Datums- und Zeitangaben wird die Funktion NowS im Calc-Objekt verwendet.

NowS ermittelt sowohl die aktuelle Uhrzeit als auch das aktuelle Datum und gibt diese als formatierten String zurück.

Es existiert ein optionaler Übergabeparameter, der den Formatierungsstring enthält. Die Formatierungstechnik wird über Platzhalter realisiert.

Dabei stehen folgende Platzhalter für folgenden Inhalt:

### 1. Die Uhrzeit:

Syntax	Beschreibung	Beispiel
\$HOUR\$	Stunde, in der die Datei erstellt wurde.	"15"
\$MIN\$	Minute, in der die Datei erstellt wurde.	"23"
\$SEC\$	Sekunde, in der die Datei erstellt wurde.	"10"

### 2. Die Tagesangabe:

Syntax	Beschreibung	Beispiel
\$DD\$	der Tag als zweistellige Zahl mit Punkt	"17"
\$WD\$	der Tag als zweistellige Zahl, Zuweisung erfolgt aufsteigend: 0=So, 1=Mo, ...	"0"
\$DAY\$	der englische Tagname als Text	"Monday"
\$TAG\$	der deutsche Tagname als Text	"Montag"

### 3. Die Monatsangabe:

Syntax	Beschreibung	Beispiel
\$MM\$	Monatsangabe als zweistellige Zahl.	"03"
\$MONTH\$	Monatsname in Englisch als Text.	"October"
\$MONAT\$	Monatsname in Deutsch als Text.	"Oktober"

**4. Die Jahresangabe:**

Syntax	Beschreibung	Beispiel
\$YYYY\$	vierstellige Jahreszahl.	"2001"
\$YY\$	zweistellige Jahreszahl .	"01"
\$Y\$	einstellige Jahreszahl.	"1"

**Syntax**

NowS(format-string)

**Parameter (optional 1)**

Wenn kein Parameter angegeben wird, ist dies gleichbedeutend mit folgenden Standard-Parametern:

Calc {

JETZT=NOW("\$DAY\$ \$DD\$. \$MM\$. \$YYYY\$ \$HOUR\$: \$MIN\$: \$SEC\$")

}

Rückgabewert

DATE

**Beispiel**

NowS("\$DAY\$ \$DD\$. \$MM\$. \$YYYY\$ \$HOUR\$: \$MIN\$: \$SEC\$")

**Ergebnis z.B.:** Thursday 31.01.2013 16:39:12

## Or

### **Funktionsbeschreibung**

Verknüpft alle PN nach den Regeln des logischen or. Der Rückgabewert ist 1, wenn mindestens ein PN den booleschen Wert 1 hat, andernfalls ist der Rückgabewert 0 (*Umkehr von **NorOr***)

### **Syntax**

Or(PN1,...,PNn)

### **Parameter (beliebig viele, min. 2)**

BOOLEAN PN1: Wert 1

BOOLEAN PN2: Wert 2

BOOLEAN PNn: Wert n

### **Rückgabewert**

BOOLEAN

### **Beispiel**

```
Or("1", "0", "0")
```

```
Or("0", "0", "0")
```

**Ergebnis 1:** 1

**Ergebnis 2:** 0

## PlaceholderS

### **Funktionsbeschreibung**

PlaceholderS nimmt den ersten PN als BasisString und tauscht in diesem String alle %x% gegen die weiteren PNs, wobei x für die Nummer des weiteren Parameters steht.

### **Syntax**

PlaceholderS(PN1,...,PNn)

### **Parameter (beliebig viele, min. 2)**

STRING PN1: Basisstring

STRING PN2: Platzhalter 1

STRING PNn: Platzhalter n

### **Rückgabewert**

STRING

### **Beispiel**

```
PlaceholderS("Hallo %1% %2%, herzlich willkommen bei uns %1% %2%","Herr","Müller")
```

**Ergebnis 1:** Hallo Herr Müller, herzlich willkommen bei uns Herr Müller

## RemoveAllS

### **Funktionsbeschreibung**

RemoveAllS entfernt jedes Vorkommen der Zeichenkette PN2 im String PN1.

### **Syntax**

RemoveAllS(PN1,PN2)

### **Parameter (2)**

STRING PN1: Basisstring

STRING PN2: Zu entfernender String

### **Rückgabewert**

STRING

### **Beispiel**

```
RemoveAllS("Automobil","mobil")
```

**Ergebnis:** Auto

## ReplaceAllS

### **Funktionsbeschreibung**

ReplaceAllS tauscht jedes Vorkommen von PN2 im String PN1 durch PN3 aus.

### **Syntax**

ReplaceAllS(PN1,PN2,PN3)

### **Parameter (3)**

STRING PN1: Basisstring

STRING PN2: Zu ersetzender String

STRING PN3: Neuer String

### **Rückgabewert**

STRING

### **Beispiel**

```
ReplaceAllS("Nadel", "d", "g")
```

**Ergebnis:** Nagel

## RoundR

### Funktionsbeschreibung

RoundR rundet die über PN1 angegebene reelle Zahl (Gleitkommazahlen).

### Syntax

RoundR(PN1,PN2,PN3,PN4)

### Parameter (4, min. 2)

REAL PN1: Originalzahl

INTEGER PN2: Rundungsrichtung als Zahl

STRING PN3: Zeichen für die Trennung von ganzem und gebrochenem Teil (optional)

STRING PN4: Zeichen für die Tausender-Trennung (optional)

### Die Rundungsrichtung wird als Zahl angegeben:

Ist die Rundungsrichtung 0, so wird ab einem Restwert von 0,5 auf, sonst aber abgerundet.

Ist die Rundungsrichtung kleiner als 0, so wird abgerundet.

Ist die Rundungsrichtung größer als 0, so wird aufgerundet.

 Das Zahlenformat des Ergebnisses der Funktion ergibt sich aus den Format-Einstellungen von PN1.

Sind PN3 und PN4 nicht angegeben, so wird das Ergebnis im deutschen Zahlenformat ausgegeben.

 Bitte beachten Sie, dass die Default-Einstellung zur Interpretation des Formates für PN1 und PN2 das deutsche Zahlenformat ist (PN3="," und PN4=".").

Verarbeiten Sie andere Zahlenformate, so müssen die Parameter PN3 und PN4 unbedingt angegeben werden

### Rückgabewert

STRING

### Beispiele

Beispiel für die Rundungsrichtung = 0:

```
RoundR("4,452", "0")
```

**Ergebnis:** 4,0

Beispiel für die Rundungsrichtung = 0:

```
RoundR("9,786","0")
Das Ergebnis ist 10,00.
```

**Ergebnis:** 10,0

Beispiel für die Rundungsrichtung > 0:

```
RoundR("2,28","1")
Das Ergebnis ist 3,00.
```

**Ergebnis:** 3,0

Beispiel für die Rundungsrichtung < 0:

```
RoundR("11,896","-1")
```

**Ergebnis:** 11,0

## SepSign

### **Funktionsbeschreibung**

SepSign dient dazu, in einer Zeichenkette bei einer Zahl mit Vorzeichen die Zahl und das Vorzeichen voneinander zu trennen.

### **Syntax**

SepSign(PN1,PN2)

### **Parameter (2)**

*INTEGER,REAL,STRING* PN1: Wert mit Vorzeichen

*INTEGER* PN2: 0 = Zahlenwert, 1 = Vorzeichen

### **Rückgabewert**

*INTEGER,REAL,STRING*

### **Beispiel**

```
SepSign("+80", "0")
SepSign("9,2-", "1")
SepSign("1,0+", "-5")
```

**Ergebnis 1:** 80

**Ergebnis 2:** -

**Ergebnis 3:** +

## SubS

### **Funktionsbeschreibung**

Bildet vom Platzhalter PN1 einen SubString, der auf Position PN2 beginnt und PN3 Zeichen lang ist.

### **Syntax**

SubS(PN1,PN2,PN3)

### **Parameter (3)**

*STRING* PN1: OriginalString

*INTEGER* PN2: Beginn des SubStrings (0 = Erstes Zeichen)

*INTEGER* PN3: Länge des SubStrings

### **Rückgabewert**

STRING

### **Beispiel**

```
SubS("Hallo Welt", "6", "4")
```

**Ergebnis:** Welt

## SubtDate

### **Funktionsbeschreibung**

Die SubtDate-Funktion subtrahiert von einem als Input gegebenen Tagesdatum eine Anzahl von Tagen und gibt als Ausgabe das errechnete Datum zurück. Dabei können die Wochenenden ignoriert werden, so dass nur die Werktage gezählt werden.

### **Syntax**

SubtDate(PN1,PN,PN3)

### **Parameter (3)**

DATE: InputDatum im Format DD.MM.YYYY oder DD.MM.YY

INTEGER PN2: Anzahl Tage, die subtrahiert werden sollen

BOOLEAN PN3: Wochenende-Schalter mit YES/NO oder 1/0; bei YES/1 werden Tage an Wochenenden mitgezählt

### **Rückgabewert**

DATE

### **Beispiel**

```
SubtDate("12.07.2002", "5", "YES")
```

**Ergebnis 3:** 07.07.2002

## SubtracI

### **Funktionsbeschreibung**

SubtracI zieht vom Integerwert von PN1 die Integerwerte aller folgenden PNs ab.

### **Syntax**

SubtracI(PN1,PN2,...,PNn)

### **Parameter (beliebig viele, min. 2)**

INTEGER PN1: Minuend

INTEGER PN2: Subtrahend 1

INTEGER PNn: Subtrahend n

### **Rückgabewert**

INTEGER

### **Beispiel**

```
SubtracI("30","5","10","-5")
```

**Ergebnis:** 20

## SubtracR

### Funktionsbeschreibung

SubtracR subtrahiert zwei reelle Zahlen (Gleitkommazahlen) voneinander.

### Syntax

SubtracR(PN1,PN2,PN3,PN4)

### Parameter (4, min. 2)

REAL PN1: 1. Minuend

REAL PN2: 2. Subtrahend

STRING PN3: Zeichen für die Trennung von ganzem und gebrochenem Teil (optional)

STRING PN4: Zeichen für die Tausender-Trennung (optional)

 Das Zahlenformat des Ergebnisses der Funktion ergibt sich aus den Format-Einstellungen von PN1.

Sind PN3 und PN4 nicht angegeben, so wird das Ergebnis im deutschen Zahlenformat ausgegeben.

 Bitte beachten Sie, dass die Default-Einstellung zur Interpretation des Formates für PN1 und PN2 das deutsche Zahlenformat ist (PN3="," und PN4=".").

Verarbeiten Sie andere Zahlenformate, so müssen die Parameter PN3 und PN4 unbedingt angegeben werden.

### Rückgabewert

REAL

### Beispiel

```
SubtracR("5000", "500,50", ",", ".")
```

**Ergebnis:** 4.499,5

## TokenS

### **Funktionsbeschreibung**

TokenS liefert den n-ten Token aus einem String zurück, wobei 1 der erste Token ist.

### **Syntax**

TokenS(PN1, PN2, PN3)

### **Parameter (3)**

*STRING* PN1: OriginalString

*INTEGER* PN2: Nummer des Tokens

*STRING* PN3: Separator zwischen den Token

### **Rückgabewert**

STRING

### **Beispiel**

```
Global.Address = "Firma\nHeinz GmbH\nWaldweg 5\n1111 Musterdorf"
Street=TokenS(Global.Address,"3","\n")
```

**Ergebnis:** Waldweg 5

## ToLowerS

### **Funktionsbeschreibung**

ToLowerS wandelt die Großbuchstaben eines Strings in Kleinbuchstaben. Alle anderen Zeichen bleiben unverändert.

Zurückgegeben wird der geänderte String (*Gegenfunktion: [ToUpperS](#)*).

### **Syntax**

ToLowerS(PN1)

### **Parameter (1)**

STRING PN1: String

### **Rückgabewert**

STRING

### **Beispiel**

```
ToLowerS("NaDeL")
```

**Ergebnis:** nadel

## ToUpperS

### **Funktionsbeschreibung**

ToLowerS wandelt die Kleinbuchstaben eines Strings in Großbuchstaben. Alle anderen Zeichen bleiben unverändert.

Zurückgegeben wird der geänderte String (*Gegenfunktion: [ToLowerS](#)*).

### **Syntax**

ToUpperS(PN1)

### **Parameter (1)**

STRING PN1: String

### **Rückgabewert**

STRING

### **Beispiel**

```
ToLowerS("NaDeL")
```

**Ergebnis:** NADEL

## TrimS

### **Funktionsbeschreibung**

TrimS entfernt nicht benötigte Zeichen vom Anfang und oder vom Ende eines Strings.

### **Syntax**

TrimS(PN1)

TrimS(PN1,PN2)

TrimS(PN1,PN2,PN3)

### **Parameter (1, 2 oder 3)**

STRING PN1: String, aus dem die Zeichen zu entfernen sind.

STRING PN2: Orientierung des Entfernens (BOTH, LEFT oder RIGHT), keine Angabe meint BOTH

STRING PN3: Das Füllzeichen (nur das erste Zeichen des Strings zählt), das entfernt werden soll

### **Rückgabewert**

STRING

### **Beispiel**

```
TrimS(" Hallo Welt ", "BOTH", " ")
```

**Ergebnis:** "Hallo Welt"

## INLINE-Sequenzen

**INLINE-Sequenzen** werden angewandt, um unformatierte Texte formatieren zu können. Dabei werden die entsprechenden Textpassagen nicht in der entsprechenden Textformatierung im R-S dargestellt, sondern die Interpretation dieser INLINE-Sequenzen erfolgt erst bei der Ausgabe der Dokumente (z.B. Druckvorschau). Die INLINE-Sequenzen sind damit lesbare und als Text editierbare Formatierungsanweisungen. Inline-Sequenzen sind besonders für Variablen wichtig, da sie die einzige Möglichkeit sind, Text in Variablen zu formatieren. Grundsätzlich sollten für formatierten Text aber *RichTexte* benutzt werden.

### ✔ Beispiel

Darstellung einer INLINE-Sequenz im R-S: Dieses \b.Wort\b0. wird in der Ausprägung Bold/Fett ausgegeben.

Darstellung einer INLINE-Sequenz im Dokument: Dieses **Wort** wird in der Ausprägung Bold/Fett ausgegeben.

Die INLINE-Sequenzen beginnen mit einem '\' und enden entweder mit einem Leerzeichen oder einem '!'.

### Kurz-Übersichtsliste der INLINE-Sequenzen

INLINE-Sequenz	Beschreibung
n	Einfügen eines Zeilenumbruchs. Ausführliche Beschreibung: <i>n - Sequenz</i>
b	Bold ist ein Schalter mit dem das Textattribut Fett ein- oder ausgeschaltet werden. Ausführliche Beschreibung: <i>b - Sequenz</i>
i	Italic ist ein Schalter mit dem das Textattribut Kursiv ein- oder ausgeschaltet werden. Ausführliche Beschreibung: <i>i - Sequenz</i>
ul	UnderLine ist ein Schalter für die Unterstreichung von Textattributen. Ausführliche Beschreibung: <i>ul - Sequenz</i>
ol	OverLine ist ein Schalter für die Überstreichung von Textattributen. Ausführliche Beschreibung: <i>ol - Sequenz</i>
sl	StrikeOutLine ist ein Schalter für die Durchstreichung von Textattributen. Ausführliche Beschreibung: <i>sl - Sequenz</i>
normal	Normal schaltet sämtliche Textattribute eines Fonts aus. Ausführliche Beschreibung: <i>normal - Sequenz</i>
plain	Zurücksetzen auf die im Design festgesetzten Font-Eigenschaften. Ausführliche Beschreibung: <i>plain - Sequenz</i>

INLINE-Sequenz	Beschreibung
dn	BaseLine-Shift versetzt die Zeichenlinie nach unten. Ausführliche Beschreibung: <b>dn - Sequenz</b>
up	BaseLine-Shift versetzt die Zeichenlinie nach oben. Ausführliche Beschreibung: <b>up - Sequenz</b>
fs	FontSize verändert die Schriftgröße. Ausführliche Beschreibung: <b>fs - Sequenz</b>
fn	Wechseln des Fonts über seinen Namen. Ausführliche Beschreibung: <b>fn - Sequenz</b>
f	Wechseln des Fonts über einen Alias-Namen. Ausführliche Beschreibung: <b>f - Sequenz</b>
cn	Setzen der Farbe. Ausführliche Beschreibung: <b>cn - Sequenz</b>
u+	Einfügen von Unicode-Zeichen in den Text, die nicht auf der Tastatur vorhanden sind. Ausführliche Beschreibung: <b>u+ - Sequenz</b>
t	Unterstützt Tabulatoren im Text. Ausführliche Beschreibung: <b>t - Sequenz</b>



INLINE-Sequenzen können bei Ressourcen vom Typ RichText nicht verwendet werden, da für RichText-Ressourcen die Formatierung über Funktionen erfolgt.

## b

### **Sequenzbeschreibung**

Bold ist ein Schalter mit den Werten 0 und 1, wobei 1 der Standard ist. Mit diesem Schalter kann das Textattribut Fett ein- oder ausgeschaltet werden.

### **Parameter**

b[0|1]

1 = eingeschaltet

0 = ausgeschaltet

### **Beispiel**

Eingabe:

\b1.Werbungs- und Transaktionsdruck\b0. werden über das \b1.R-S\b0. vereinigt.

Ausgabe:

**Werbungs- und Transaktionsdruck** werden über das **R-S** vereinigt.

## cn

### Sequenzbeschreibung

Color setzt die Farbe des Textes. Dabei können 2 unterschiedliche Syntax-Varianten zum Einsatz kommen.

In Variante 1 wird die Farbe über einen RGB-Wert mit dem Wertevorrat von 0 bis 255 gesetzt.

In Variante 2 kann eine Farbe über einen vordefinierten Namen gewählt werden. Die Farbnamen sind aus dem Standard HTML 4.0 abgeleitet.

Folgende Farbnamen sind bekannt: aqua, navy, black, olive, blue, purple, fuchsia, red, gray, silver, green, teal, lime, yellow, maroon, white

### Parameter

cnrgb(rot,grün,blau)

oder

cnName

### Beispiel

Eingabe:

```
\cnred.Werbungs- und Transaktionsdruck\cnblack. werden über das \cnrgb(255,0,0).R-
S\cnrgb(0,0,0). vereinigt.
```

Ausgabe:

Werbungs- und Transaktionsdruck werden über das R-S vereinigt.

## dn

### **Sequenzbeschreibung**

BaseLine-Shift down verschiebt die BaseLine, auf der die Buchstaben stehen, nach unten. Die BaseLine beschreibt den Fußpunkt der großgeschriebenen Buchstaben.

### **Parameter**

dn[points]

Points ist die Angabe der Verschiebung in ganzen Punkten, Kommawerte sind nicht zulässig.

### **Beispiel**

Eingabe:

\dn[2].Werbungs- und Transaktionsdruck werden über das R-S. vereinigt.

## f

### **Sequenzbeschreibung**

FontNumber: Wechseln des Fonts über einen Alias-Namen.

Soll der Font über einen Alias-Namen angesprochen werden, so ist dieser zuvor in der reportw.ini unter Fonts/Replacements zu definieren.

### **Parameter**

f[FontNumber]

### **Beispiel**

Eingabe:

```
\f1.Werbungs- und Transaktionsdruck\f1. werden über das \f2.R-S\f2. vereinigt.
```

Ausgabe:

**Werbungs- und Transaktionsdruck** werden über das **R-S**. vereinigt.

## fn

### Sequenzbeschreibung

FontName: Wechseln des Fonts über seinen Namen.

FontName setzt den Namen eines Fonts, wobei es eventuell vorkommende Hochkommas eliminiert.

Soll der Font über einen Alias-Namen angesprochen werden, so ist dieser zuvor in den Konfigurationen der Output-Management-Produkte (reportw.ini/logistics.ini) unter Fonts/Replacements zu definieren.

- i** Es können nur Font-Namen von Fonts verwendet werden, welche über das Transaktions-Formular (fonts.ini) bekannt gegeben werden.  
Bitte kontaktieren Sie den Entwickler des Transaktions-Formular bezüglich der Liste der möglichen Font-Namen.  
Desweiteren ist zu beachten, dass Fontnamen welche Leerzeichen enthalten immer in doppelten Hochkomma eingeschlossen werden müssen. Beispiel: \fn"Book Antiqua"

### Parameter

fn[FontName]

### Beispiel

Eingabe:

```
\fnArialRoundedMTBold.Werbungs- und Transaktionsdruck\plain werden über das R-S. vereinigt.
```

Ausgabe:

**Werbungs- und Transaktionsdruck** werden über das R-S. vereinigt.

## fs

### **Sequenzbeschreibung**

FontSize setzt die Größe des gewählten Fonts.

### **Parameter**

fs[points]

Points ist die Angabe der Schriftgröße in Dekapunkten, Kommawerte sind nicht zulässig.

 Die in Textverarbeitung typische Angaben von "12" oder "14" Punkt-Schriftgrößen ist immer mit dem Faktor 10 zu multiplizieren, um Dekapunkten zu erhalten. Damit ergeben sich die Beispiel-Werte "120" und "140".

### **Beispiel**

Eingabe:

```
\fs140.Werbungs- und Transaktionsdruck\fs100. werden über das \b1.R-S\b0. vereinigt.
```

Ausgabe:

**Werbungs- und Transaktionsdruck** werden über das **R-S** vereinigt.

## i

### **Sequenzbeschreibung**

Italic ist ein Schalter mit den Werten 0 und 1, wobei 1 der Standard ist. Mit diesem Schalter kann das Textattribut Kursiv ein- oder ausgeschaltet werden.

### **Parameter**

i[0|1]

0 = ausgeschaltet

1 = eingeschaltet

### **Beispiel**

Eingabe:

\i1.Werbungs- und Transaktionsdruck\i0. werden über das \i1.R-S\i0. vereinigt.

Ausgabe:

*Werbungs- und Transaktionsdruck* werden über das *R-S* vereinigt.

## n

### **Sequenzbeschreibung**

n ist ein NewLine-Zeichen. Es bewirkt einen Zeilenumbruch.

### **Parameter**

n

### **Beispiel**

Eingabe:

Werbungs- und Transaktionsdruck\nwerden über das R-S vereinigt.

Ausgabe:

Werbungs- und Transaktionsdruck  
werden über das R-S vereinigt.

## normal

### **Sequenzbeschreibung**

Normal schaltet sämtliche Textattribute eines Fonts aus. Der danach folgende Text erfolgt als regulärer Text ohne Verschiebung der BaseLine.

### **Beispiel**

Eingabe:

```
\b1.\i1.Werbungs- und Transaktionsdruck\normal. werden über das \b1.R-S\b0. vereinigt.
```

Ausgabe:

***Werbungs- und Transaktionsdruck*** werden über das **R-S** vereinigt.

## ol

### Sequenzbeschreibung

OverLine ist ein Schalter mit den Werten w, db, 0 oder 1, wobei 1 der Standard ist. Hier können die Eigenschaften für die Überstreichung gesteuert werden.

### Parameter

ol[w|db|0|1]

w = Wortüberstreichung, die Zwischenräume zwischen den Worten werden nicht überstrichen

db (DoubleLine) = es werden zwei Überstreichungslinien gezeichnet

0 = Ausschalten der Überstreichungen

1 = Einfache Überstreichung

### Beispiele

Eingabewerte 1 und 0:

```
\ol1.Werbungs- und Transaktionsdruck\ol0. werden über das \ol1.R-S\ol0. vereinigt.
```

Ausgabewerte 1 und 0:

Werbungs- und Transaktionsdruck werden über das R-S vereinigt.

Eingabewert w:

```
\olw1.Werbungs- und Transaktionsdruck\olw0. werden über das \olw1.R-S\olw0. vereinigt.
```

Ausgabewert w:

Werbungs- und Transaktionsdruck werden über das R-S vereinigt.

Eingabewert db:

```
\oldb1.Werbungs- und Transaktionsdruck\oldb0. werden über das R-S vereinigt.
```

Ausgabewert db:

Werbungs- und Transaktionsdruck werden über das R-S vereinigt.

## plain

### **Sequenzbeschreibung**

Plain ist ein Schalter, der alle gewählten Textattribute zurücksetzt auf den Font mit all seinen Eigenschaften, der im Design für dieses Objekt vorgegeben war.

### **Parameter**

-

### **Beispiel**

Eingabe:

```
\cngreen.Werbungs- und Transaktionsdruck\plain. werden über das \cngreen.R-S\plain. vereinigt.
```

Ausgabe:

Werbungs- und Transaktionsdruck werden über das R-S vereinigt.

## sl

### Sequenzbeschreibung

StrikeOutLine ist ein Schalter mit den Werten w, db, 0 oder 1, wobei 1 der Standard ist. Hier können die Eigenschaften für die Durchstreichung gesteuert werden.

### Parameter

sl[w|db|0|1]

w - Wortunterstreichung, die Zwischenräume zwischen den Worten werden nicht durchgestrichen

db (DoubleLine) - es werden zwei Durchstreichungslinien gezeichnet

0 - Ausschalten der Durchstreichungen

1 - Einfache Durchstreichung

### Beispiele

Eingabewerte 1 und 0:

```
\sl1.Werbungs- und Transaktionsdruck\sl0. werden über das \sl1.R-S\sl0. vereinigt.
```

Ausgabewerte 1 und 0:

~~Werbungs- und Transaktionsdruck~~ werden über das ~~R-S~~ vereinigt.

Eingabewert w:

```
\slw1.Werbungs- und Transaktionsdruck\slw0. werden über das \slw1.R-S\slw0. vereinigt.
```

Ausgabewert w:

~~Werbungs- und Transaktionsdruck~~ werden über das ~~R-S~~ vereinigt.

Eingabewert db:

```
\sldb1.Werbungs- und Transaktionsdruck\sldb0. werden über das R-S vereinigt.
```

Ausgabewert db:

~~Werbungs- und Transaktionsdruck~~ werden über das R-S vereinigt.

## t

### **Sequenzbeschreibung**

Anhand von \t gelangt man zum nächsten Tabulator. Durch Wiederholen des Befehls kommt man zu weiteren Tab-Stopps.

Dieser Schalter dient als Alternative zum Setzen eines ASCII Dezimal 9-Tabulators in den Datenfluss, um zum nächsten Tab-Stopp zu gelangen.

### **Parameter**

t

### **Beispiel**

Eingabe:

```
Werbungs- und Transaktionsdruck \t.werden\t. \t.über\t. \t.\t.das\t.\t. R-S vereinigt.
```

Ausgabe:

Werbungs- und Transaktionsdruck    werden    über    das    R-S vereinigt.

## u+

### **Sequenzbeschreibung**

u+ ist ein Insert Unicode Code Point. Mit \u+ können Unicode-Zeichen in den Text eingefügt werden, die nicht auf der Tastatur vorhanden sind.

### **Parameter**

u+[codepoint]

Codepoint steht als hexadezimale Zahl für ein Unicode-Zeichen.

### **Beispiel**

Eingabe:

Werbungs- und Transaktionsdruck werden \u+00FC ber das R-S vereinigt.

Ausgabe:

Werbungs- und Transaktionsdruck werden über das R-S vereinigt.

## ul

### Sequenzbeschreibung

UnderLine ist ein Schalter mit den Werten w, db, 0 oder 1, wobei 1 der Standard ist. Hier können die Eigenschaften für die Unterstreichung gesteuert werden.

### Parameter

ul[w|db|0|1]

w - Wortunterstreichung, die Zwischenräume zwischen den Worten werden nicht unterstrichen

db (DoubleLine) - es werden zwei Unterstreichungslinien gezeichnet

0 - Ausschalten der Unterstreichungen

1 - Einfache Unterstreichung

### Beispiel

Eingabewerte 1 und 0:

```
\ul1.Werbungs- und Transaktionsdruck\ul0. werden über das \ul1.R-S\ul0. vereinigt.
```

Ausgabewerte 1 und 0:

Werbungs- und Transaktionsdruck werden über das R-S vereinigt.

Eingabewert w:

```
\ulw1.Werbungs- und Transaktionsdruck\ulw0. werden über das \ulw1.R-S\ulw0. vereinigt.
```

Ausgabewert w:

Werbungs- und Transaktionsdruck werden über das R-S vereinigt.

Eingabewert db:

\uldb1.Werbungs- und Transaktionsdruck\uldb0. werden über das R-S vereinigt.

Ausgabewert db:

Werbungs- und Transaktionsdruck werden über das R-S vereinigt.

## up

### **Sequenzbeschreibung**

BaseLine-Shift up verschiebt die BaseLine, auf der die Buchstaben stehen, nach unten. Die BaseLine beschreibt den Fußpunkt der großgeschriebenen Buchstaben.

### **Parameter**

up[points]

Points ist die Angabe der Verschiebung in ganzen Punkten, Kommawerte sind nicht zulässig.

### **Beispiel**

```
\dn[2].Werbungs- und Transaktionsdruck. werden über das R-S. vereinigt.
```

# Administrator-Dokumentation

Dokumentation des Redaktions-Systems (R-S) für den System-Administrator. In der Administrator-Dokumentation finden Sie Hilfe zur Installation, Wartung und Administration des R-S.

Als Nutzer beachten Sie bitte die **Nutzer-Dokumentation**.

 Für die meisten in dieser Dokumentation gezeigten Aktionen benötigen Sie Administratorrechte.

## 1. Installation

- **System-Voraussetzungen**
  - Welche Voraussetzungen Ihr System erfüllen muss, um einen reibungslosen Betrieb des R-S zu gewährleisten.
- **Installation**
  - Informationen zur Installation und Einrichtung des R-S.
- **Systemsicherheit**
  - Grundlegende Informationen zur Systemsicherheit.

## 2. Systembetrieb

- **Administrator-Client**
  - Mit dem Administrator-Client wird das R-S in seinen Grundfunktionen administriert.
- **Backup**
  - Erläutert, wie Sie ein Backup des R-S Systems anfertigen.
- **Binär-Pakete**
  - Binär-Pakete enthalten die Programme, Skripte und Konfigurationen um aus Produktions-Paketen Dokumente zu generieren.
- **R-S Server Konfigurationsdatei**
  - Sie definiert die wesentlichen Einstellungen und Verhaltensweisen des R-S und ermöglicht deren Konfiguration.

## 3. Berechtigungen

- **Arbeitsbereich-Administrator**
  - Ein R-S Nutzer mit weitgehenden Administratorrechten in einem bestimmten Arbeitsbereich
- **Nutzer**
  - Erläutert die Verwaltung der Nutzer des R-S.
- **Gruppen**
  - Erläutert die Verwaltung von Gruppen, in denen Sie R-S Nutzer organisieren können.
- **Rechte**
  - Beschreibt, wie Rechte vergeben werden und welche Einschränkungsmöglichkeiten bestehen.

#### **4. Systemadministration**

- **Administrator-Client UI**
  - Ab Vers. 2.1.1000 des R-S steht für administrative Aufgaben die komfortable grafische Oberfläche des Admin-Clients zur Verfügung.
- **Administrator-Client Konsole**
  - Wenn Sie mit einer älteren Version als der aktuellen 2.1 arbeiten, müssen Sie zum Administrieren des R-S die klassische Konsole verwenden.

## Administrator-Client UI

Für administrative Aufgaben wie Anlegen von Nutzern, Arbeitsbereichen, Schriftarten oder Binär-Paketen, der Rechteverwaltung etc. steht Ihnen im R-S der Administrator-Client (Admin-Client) zur Verfügung.

Bei laufendem Redaktions-System ist das Admin-Client UI (*User Interface*) standardmäßig unter folgender URL erreichbar: **http://<ip>:<port>/rs-admin**, wobei **ip** = die IP-Adresse des Rechners ist, auf dem der R-S Server läuft (z.B. localhost) und **port** der Port, auf dem das R-S lauscht (Standard: 8080).

Die grafische Oberfläche des Admin-Clients folgt dem gleichen Aufbau wie die Oberfläche des R-S Clients, was Ihnen die Arbeit damit deutlich erleichtern wird (siehe dazu auch **Nutzeroberfläche und Navigation** in der R-S Dokumentation). Hilfe zum Schnelleinstieg finden Sie auf den folgenden Seiten, dazu gehört:

- **Anmelden** am Admin-Client UI,
- **Navigieren** durch die verschiedenen Funktionen,
- Verwalten von **Nutzern, Schriftarten** und **Binär-Paketen**,
- Die **graphische Objektübersicht** (inkl. Löschen / Wiederherstellen von Objekten).

## Anmeldung am R-S Admin-Client

Nachdem Sie den R-S Admin-Client in Ihrem Browser geöffnet haben werden Sie als erstes gebeten, sich anzumelden.

Dazu erscheint, wie bei der Anmeldung am R-S Client, folgender Dialog:

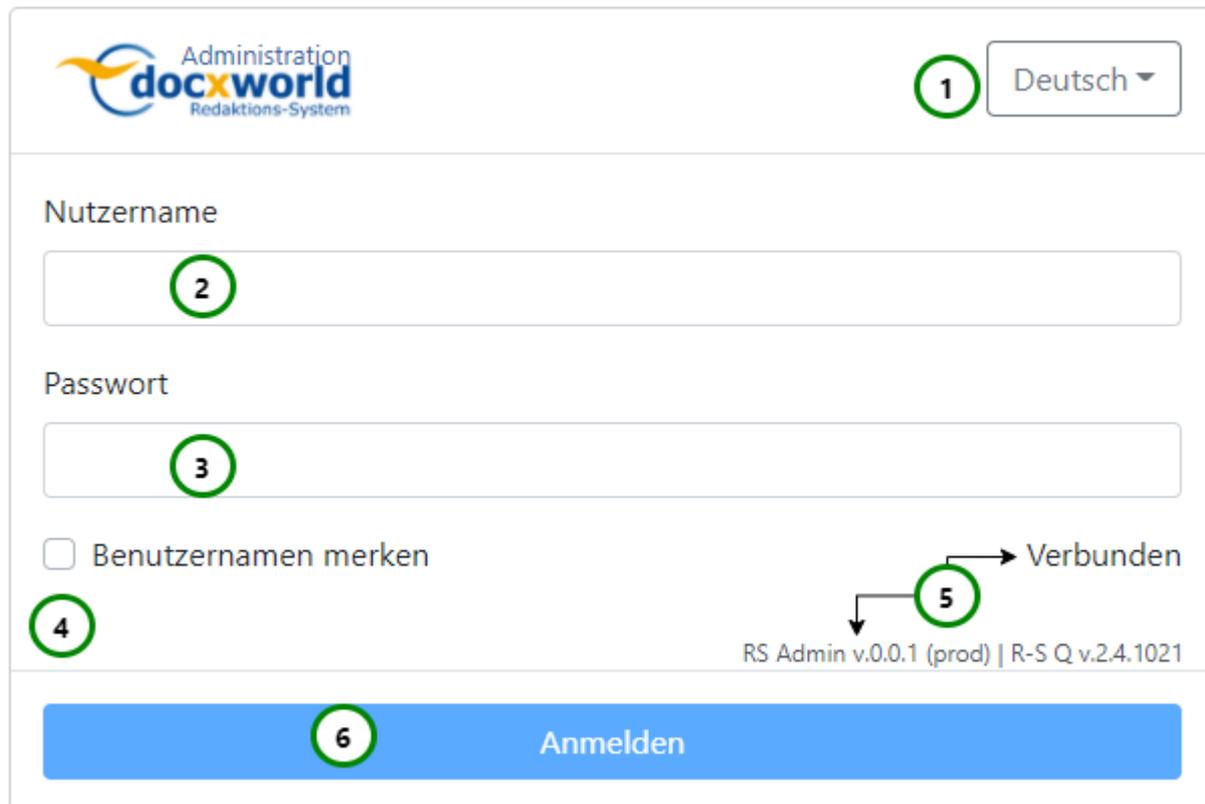


Abbildung A

Um sich anzumelden, müssen Sie Ihren Nutzernamen Abb. A (2) und Ihr Passwort (3) eingeben.

**!** Um sich am R-S Admin-Client anmelden zu können, benötigen Sie Administratorrechte. Im Auslieferungszustand ist bereits ein Administratorkonto angelegt (Benutzer: **admin** | Passwort: **admin**). Ebenso wird, wenn kein Benutzerkonto existiert, automatisch eines bei Systemstart generiert. Bitte verwenden Sie für das Administratorkonto nicht die Standardanmeldedaten, sondern ändern Sie diese umgehend (siehe **Nutzer ändern**). Eine Anmeldung ist nur über das Administratorkonto möglich, normale Nutzerkonten haben keinen Zugriff auf den Admin-Client.

**Sie können außerdem:**

- eine **Sprache** wählen *Abb. A (1)*.

Um die Sprache zu ändern, klicken Sie bitte auf  (1) und wählen eine andere Sprache aus. Die Standardsprache ist Deutsch. Momentan verfügbare Sprachen sind Deutsch und Englisch.

- den **Benutzernamen** merken (4).

Der Benutzername ist beim nächsten Einloggen bereits eingetragen (Nur in Ihrem Browser).

Nachdem Sie sich erfolgreich angemeldet haben *Abb. A (6)* gelangen Sie initial zur **Nutzer-Übersicht**.

In *Abb. A (5)* sehen Sie die Admin-Client und Server-Version, sowie den Status der Verbindung.

## Nutzeroberfläche des RS-Admin-Clients

In diesem Kapitel machen wir Sie mit dem Aufbau der Bildschirm-Masken und den Möglichkeiten zur Navigation vertraut. Es wird außerdem gezeigt, welche Ansichten Ihnen im R-S-Admin-Client zur Verfügung stehen und welche Objekte sie jeweils beinhalten.

Nachdem Sie sich angemeldet haben, werden Sie, je nach Berechtigungen, in den ersten für Sie freigegebenen Bereich weitergeleitet.

The screenshot shows the 'Nutzer' (Users) management interface. At the top, there is a navigation bar (1) with 'Nutzer', 'Arbeitsbereiche', 'Schriftarten', and 'Binär-Pakete'. To the right, a user profile dropdown (2) shows 'admin' and options for 'Einstellungen' and 'Hilfe'. The main area (3) is titled 'Nutzer' and contains a table of users. The user 'Joey Lauterbach' (ID 57005) is highlighted. To the right of the table is a 'Nutzer Vorschau' (User Preview) panel showing details for the selected user.

ID	Vorname	Nachname	Benutzername	Geändert	Status	Onli...
1002	admin	admin	admin	08.01.21 06:43:28	active	■
57005	Joey	Lauterbach	Joey	08.01.21 06:50:21	active	■
104014	Monique	Oehme	Monique	09.12.20 06:58:56	active	■
83018	Nurcan	Günes-Balci	Nurcan	16.11.20 12:32:01	active	■
1001	production	production	production	09.08.11 13:58:13	active	■
100342	Roman	Nesterko	Roman	25.11.20 13:56:59	active	■

The 'Nutzer Vorschau' panel shows the following details for user ID 57005:

- ID: 57005
- Erstellt: 29.10.20 09:14:23
- Geändert: 08.01.21 06:50:21
- Vorname: Joey
- Nachname: Lauterbach
- Email: Joey
- Status: ACTIVE
- Online: ■
- Rechte: MANAGER, USER, BINARY-MANAGER
- Arbeitsbereiche: Handbuch, ProfiForms

Abbildung A

### Der Aufbau der Bildschirm-Masken besteht immer aus drei Teilen:

- **Navigation:** Mit dem Navigations-Pfad wechseln Sie durch die Funktionen *Abb. A (1)*.
- **Optionen:** Hier finden Sie grundlegende Steuerungsoptionen des R-S-Admins (2)
  - Sie können Ihr Profil verwalten, um so ein neues Passwort zu vergeben oder um zu sehen, auf welche Arbeitsbereiche Sie Zugriff haben.
  - Die Sprache zu wechseln oder
  - sich abmelden.
- **Inhalt:** Hier finden Sie die Inhalte und Aktionen zu den ausgewählten Navigationspunkten (3).

## Funktionen des Admin-Clients

In der Navigation finden Sie, je nach Berechtigungen, folgende Punkte:



Nutzer Arbeitsbereiche Schriftarten Binär-Pakete

- **Nutzer**

Hier können Sie Nutzer anlegen, ändern, entfernen und ihnen Arbeitsbereiche zuweisen.

- **Arbeitsbereiche**

Hier können Sie Arbeitsbereiche erstellen und ändern. Außerdem können Sie sich hier die grafische Objektübersicht für einen Arbeitsbereich anzeigen lassen, in der Sie auch R-S Objekte löschen und wiederherstellen können.

- **Schriftarten**

Hier können Sie Schriftarten für Bibliotheken anlegen und ändern.

- **Binär-Pakete**

Hier können Sie Binär-Pakete erstellen, löschen und ändern.

## Navigation - Nutzer

Um R-S Nutzer und R-S Admin-Rechte zu verwalten, wählen Sie in der Navigation Nutzer aus. Sie gelangen dann zu folgender Übersicht:

ID	Vorname	Nachname	Benutzername	Geändert	Status	Onli...
1002	admin	admin	admin	08.01.21 06:43:28	active	■
57005	Joey	Lauterbach	Joey	08.01.21 06:50:21	active	■
104014	Monique	Oehme	Monique	09.12.20 06:58:56	active	■
83018	Nurcan	Günes-Balci	Nurcan	16.11.20 12:32:01	active	■
1001	production	production	production	09.08.11 13:58:13	active	■
100342	Roman	Nesterko	Roman	25.11.20 13:56:59	active	■

10 Benutzer

Profiforms R-S Q

Abbildung A

In Abb. A (3) finden Sie eine Auflistung der Nutzer mit folgenden Informationen:

- **ID:** Die interne, eindeutige ID des Nutzers
- **Vor-/ Nachname**
- **Benutzername:** Der Benutzername wird beim Login abgefragt  
 ⚠ Der Benutzername entspricht dem Feld Email bei der Erstellung des Nutzers und wird bei der Anmeldung benötigt.
- **Änderungsdatum:** Letzte Änderung mit Datum und Uhrzeit.
- **Status :** Der Status des Benutzers (*NEW* | *ACTIVE* | *INACTIVE* | *BLOCKED*) , siehe hierzu auch [erstellen / ändern eines Nutzers](#) .  
 ⚠ Benutzerpasswörter werden aus Sicherheitsgründen nicht angezeigt.  
 ⚠ Nutzer mit Status AKTIV / ACTIVE können nicht gelöscht werden.
- **Online:** Zeigt an, ob der Nutzer gerade am R-S angemeldet ist ■ (grün) oder nicht ■ (rot).

Verwenden Sie die entsprechende Suchleiste (0), um einen Nutzer oder Arbeitsbereich zu suchen. Dabei werden alle Spalten mit in die Suche einbezogen. Mehr zur Suchfunktion finden Sie [hier](#).

Wählen Sie einen Nutzer / Arbeitsbereich aus, um dieses Objekt zu bearbeiten.

**Sie können in Abb. A:**

- Nutzer mit Rechten erstellen oder die Rechte ausgewählter Nutzer ändern (1), mehr dazu [hier](#).
- Den ausgewählten Nutzer löschen (2).  
⚠ Sie können nur Nutzer löschen, die nicht den Status active haben.

Außerdem werden Ihnen auf der rechten Seite Abb. A (4) Informationen zum ausgewählten Nutzer und seinen Rechten angezeigt:

- Die **ID** des Nutzers.
- Wann der Nutzer **erstellt** wurde.
- Wann die Rechte des Nutzers **zuletzt geändert** wurden.
- **Vor-** und **Nachname** des Nutzers.
- Den **Benutzernamen**.
- Den **Status** (*NEW | ACTIVE | INACTIVE | BLOCKED*).
- Ob der Nutzer gerade **Online** ist ( ■ oder ■ ).
- Die **Rechte** des Nutzers.
- Die **zugewiesenen Arbeitsbereiche** des Nutzers.

## Nutzer erstellen, ändern und Rechte vergeben

Verwenden Sie den Erstellen- bzw. Ändern-Button in der Nutzerübersicht, um einen neuen Benutzer-Account anzulegen oder die Rechte des in der Liste ausgewählten Benutzers zu ändern. Die hier zugewiesenen Rechte regeln die Rechte innerhalb des R-S Admin Clients und die Nutzerrolle.

Sie geben dem Nutzer Befugnisse über den Zugriff auf den Admin-Client und seine allgemeine Rolle im R-S. Diese Rechte sind nicht zu verwechseln mit den Rechten, die Sie im R-S Client **vergeben** können und die die Befugnisse innerhalb dieses Arbeitsbereichs im R-S Client definieren.

ID	Vorname	Nachname	Benutzername	Geändert	Status	Onli...
1002	admin	admin	admin	08.01.21 06:43:28	active	■
57005	Joey	Lauterbach	Joey	08.01.21 06:50:21	active	■
104014	Monique	Oehme	Monique	09.12.20 06:58:56	active	■
83018	Nurcan	Günes-Balci	Nurcan	16.11.20 12:32:01	active	■
1001	production	production	production	09.08.11 13:58:13	active	■
100342	Roman	Nesterko	Roman	25.11.20 13:56:59	active	■

Abbildung A

Wenn Sie einen neuen Nutzer anlegen Abb. A (1) werden Sie direkt zur Eigenschaftsseite des neu angelegten Nutzers geleitet. Durch Ändern (2) öffnet sich die Eigenschaftsseite des in der Nutzerliste (3) ausgewählten Nutzers.

Nutzer / Ändere Nutzer "Joey Lauterbach"

Speichern 1 | Speichern und schließen 2 | Verlassen 3 | Passwort zurücksetzen 3

Vorname: Joey | ID: 57005

Nachname: Lauterbach | Erstellt: 29.10.20 09:14:23 5

Email: Joey | Geändert: 08.01.21 06:50:21

Status: ACTIVE 4

Arbeitsbereiche 6 | Rechte 7

Arbeitsbereiche filtern 0

Verfügbare Arbeitsbereiche			Ausgewählte Arbeitsbereiche			
ID	Name	Geändert	ID	Name	Geändert	Ad...
2193989	Profiforms_DEV	08.01.21 06:45:38	104010	Handbuch	07.12.20 14:46:53	<input checked="" type="checkbox"/>
2193985	Profiforms_QA	08.01.21 06:44:51	627409	ProfiForms	29.10.20 14:16:25	<input checked="" type="checkbox"/>
707662	Profiforms_Training	08.01.21 06:44:42				

27 Arbeitsbereiche | 2 Arbeitsbereiche

Abbildung B

Wenn Sie einen neuen Nutzer angelegt haben, sind die Elemente Abb. B (4 & 5) leer (da der Nutzer noch keine Werte besitzt). Andernfalls enthalten diese Elemente die aktuellen Werte des Nutzers, den Sie ändern möchten.

Bei einem bereits bestehenden Nutzer werden Ihnen folgende Details angezeigt Abb. B (5):

- **ID:** Die interne ID des Nutzers (siehe auch Erweiterte Ansicht).
- **Erstellt:** Der Zeitpunkt, an dem der Nutzer erstellt wurde.
- **Geändert:** Der Zeitpunkt, an dem der Nutzer zuletzt geändert wurde.

Sowohl beim Erstellen als beim Ändern eines Nutzers können Sie **Abb. B (4)**:

- Vorname des Nutzers angeben / ändern.
- Nachname des Nutzers angeben / ändern.
- Den Status des Nutzers ändern:
  - **ACTIVE**: Das Benutzerkonto ist aktiviert und kann verwendet werden (Standard). Nur mit einem aktivierten Konto können Sie sich am R-S anmelden.
  - **NEW**: Das Benutzerkonto wurde neu angelegt und ist noch nicht aktiviert, eine Anmeldung am R-S mit dem Konto ist noch nicht möglich.
  - **BLOCKED**: Sperrt das Benutzerkonto, es kann dann nicht mehr verwendet werden (Keine Anmeldung über dieses Konto möglich.).
  - **INACTIVE**: Das Benutzerkonto ist inaktiv, es kann nicht verwendet werden (Keine Anmeldung über dieses Konto möglich.).

 Systemnutzer können weder gelöscht, noch INACTIVE gesetzt werden.

Wenn Sie einen Nutzer neu anlegen, unterscheidet sich Ansicht **Abb. C** im Gegensatz zum Ändern eines Nutzers **Abb. B (4)** in folgenden Punkten:

- Bei einem neu angelegten Nutzer müssen Sie eine E-Mail Adresse angeben. Die E-Mail Adresse fungiert auch als Benutzername bei der Anmeldung.
-  Bei einem bereits bestehenden Nutzer können Sie die E-Mail Adresse **nicht mehr** ändern.
- Sie müssen ein Passwort angeben.

Vorname    
 Kann nicht leer sein

Nachname    
 Kann nicht leer sein

Email    
 Kann nicht leer sein

Passwort    
 Kann nicht leer sein

Status

Abbildung C

Hingegen können Sie beim Ändern eines Nutzers das Nutzerpasswort zurücksetzen Abb. B (3).

Abbildung D

Geben Sie das neue Passwort und Passwort wiederholen ein und speichern Sie es. Das Passwort wurde jetzt zurückgesetzt und der Nutzer kann sich (nachdem Sie die Änderungen mit Abb. B (1) gespeichert haben) mit dem neuen Passwort anmelden.

 Standardmäßig gibt es keine Mindestlänge und Validierung für Passwörter. Dies kann aber in der [rs.server.properties](#) geändert werden.

Wenn Sie die Bearbeitung beendet haben, speichern Sie den Nutzer Abb. B (1) (neu erstellte Nutzer werden erst dann angelegt) oder verwerfen Sie die Änderungen (2) (neu angelegte Nutzer werden wieder verworfen).

## Zuweisung von Arbeitsbereichen

Arbeitsbereiche Rechte

Arbeitsbereiche filtern

Verfügbare Arbeitsbereiche				Ausgewählte Arbeitsbereiche			
ID	Name	↓	Geändert	ID	Name	Geändert	Admin
2193989	Profiforms_DEV		08.01.21 06:45:38	104010	Handbuch	07.12.20 14:46:53	<input checked="" type="checkbox"/>
2193985	Profiforms_QA		08.01.21 06:44:51	627409	ProfiForms	29.10.20 14:16:25	<input checked="" type="checkbox"/>
707662	Profiforms_Training		08.01.21 06:44:42				

27 Arbeitsbereiche

2 Arbeitsbereiche

Abbildung E

Um Arbeitsbereiche zuzuordnen, müssen Sie bei der Erstellung oder Änderung eines Nutzers auf Arbeitsbereiche gehen [Abb. B \(6\)](#) bzw. [Abb. E \(1\)](#).

Über die Arbeitsbereichslisten [Abb. E \(2 & 3\)](#) können Sie dem Nutzer Arbeitsbereiche, zu denen er Zugang haben soll, zuweisen. Der Nutzer hat nur Zugang zu Arbeitsbereichen, die sich in der Liste Ausgewählte Arbeitsbereiche (3) befinden, nicht aber zu den Arbeitsbereichen in der Liste Verfügbare Arbeitsbereiche (2).

- Um dem Nutzer einen Arbeitsbereich zuzuweisen, wählen Sie in der Liste der verfügbaren Arbeitsbereiche (2) den gewünschten Arbeitsbereich aus und verschieben diesen mit **>** (4) in die Liste ausgewählter Arbeitsbereiche.
- Um einen zugewiesenen Arbeitsbereich wieder zu entfernen, wählen Sie in der Liste ausgewählter Arbeitsbereiche (3) den gewünschten Arbeitsbereich aus und verschieben diesen mit **<** (4) zurück in die Liste verfügbarer Arbeitsbereiche.

**!** Wenn Sie in Spalte Admin [Abb. E \(5\)](#) einen Haken setzen, gewähren Sie dem Nutzer dadurch **Administrator-Rechte** für den betreffenden Arbeitsbereich.

Verwenden Sie die entsprechenden Suchleisten ([Abb. A \(0\)](#), [Abb. B \(0\)](#) und [Abb. E \(0\)](#)), um einen Nutzer oder Arbeitsbereich zu suchen. Dabei werden alle Spalten in die Suche mit einbezogen. Mehr zur Suchfunktion finden Sie [hier](#).

## Rechte auf den Admin-Bereich vergeben

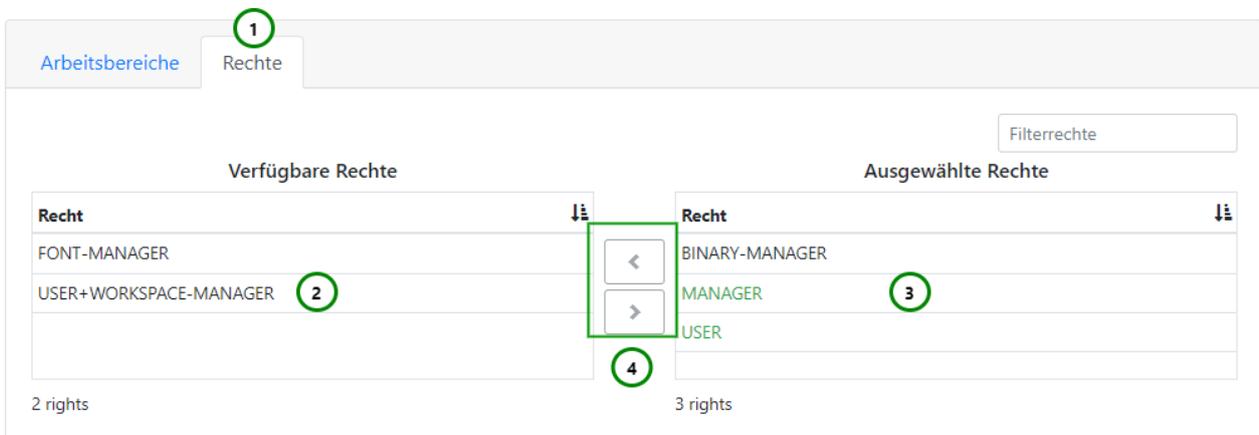


Abbildung F

Um Rechte zuzuordnen, müssen Sie bei der Erstellung oder Änderung eines Nutzers auf Rechte gehen [Abb. B \(7\)](#) bzw. [Abb. F \(1\)](#).

Nutzer können insgesamt 3 Admin-Client Rechte besitzen:

- **BINARY-MANAGER**  
Der Nutzer darf im Administrator-Client nur über Binär-Pakete verfügen. Er hat also lediglich Zugriff auf die [Binär-Pakete Übersicht](#).
- **FONT-MANAGER**  
Der Nutzer darf im Administrator-Client nur über Fonts verfügen. Er hat also lediglich Zugriff auf die [Schriftarten-Übersicht](#).
- **USER+WORKSPACE-MANAGER**  
Der Nutzer darf im Administrator-Client nur über Nutzer und Arbeitsbereiche verfügen. Er hat also lediglich Zugriff auf die [Nutzer- und Arbeitsbereiche-Übersicht](#).

Sie können einem Nutzer die 3 Rechte einzeln zuweisen.

- Um dem Nutzer das jeweilige Recht zuzuweisen, wählen Sie das Recht in Verfügbare Rechte [Abb. F \(2\)](#) aus und verschieben Sie es mit  (4) nach Ausgewählte Rechte (3).
- Um einem Nutzer ein Recht wieder zu entziehen, wählen Sie das Recht in Ausgewählte Rechte (3) aus und verschieben es mit  (4) nach Verfügbare Rechte (2).

Außerdem gibt es 4 Rollen:

- **USER:** Zugriff auf Arbeitsbereiche im R-S Client.
- **PRODUCTION:** Kein Zugriff auf Arbeitsbereiche im R-S Client, Produktion.
- **MANAGER:** Zugriff auf Teile des R-S Admin.
- **ADMIN:** Zugriff auf gesamten R-S Admin.

Nutzer können auch mehrere Rollen besitzen. Dabei hat jeder manuell neu im R-S Admin-Client erstellte Nutzer automatisch die Rolle USER. Rollen können im Admin-Client nicht direkt zugewiesen werden, Sie werden impliziert vergeben. So haben Administrator-Nutzer automatisch alle Rollen und Rechte, Nutzer mit BINARY-, FONT-, oder USER+WORKPSACE-MANAGER - Rechten automatisch die Rolle MANAGER, etc. Administratoren (ADMIN) können nicht zugewiesen werden, sondern müssen explizit als solche erstellt werden, z.B. über die Admin-Client Konsole.

USER und PRODUCTION haben keinen Zugriff auf den Admin-Client, MANAGER nur auf bestimmte Ansichten (Je nach vergebenen Rechten, siehe oben) und ADMIN auf den gesamten Admin-Client.

## Navigation - Arbeitsbereiche

Um R-S Nutzer und Arbeitsbereiche zu verwalten, wählen Sie in der Navigation Arbeitsbereiche aus. Sie gelangen dann zu folgender Übersicht:

Administration **docxworld** Redaktions-System

Nutzer Arbeitsbereiche Schriftarten Binär-Pakete admin ▾ Einstellungen ▾ Hilfe ▾

Arbeitsbereiche

Erstellen **1** Ändern **2** Objekte anzeigen **3** Arbeitsbereiche filtern **0**

ID	Name	Geändert
627409	Profiforms	29.10.20 14:16:25
2193989	Profiforms_DEV	08.01.21 06:45:38
2193985	Profiforms_QA	08.01.21 06:44:51
707662	Profiforms_Training	08.01.21 06:44:42

Arbeitsbereich Vorschau

ID	2193989
Erstellt	18.02.16 13:22:00
Geändert	08.01.21 06:45:38 <b>4</b>
Name	Profiforms_DEV
Benutzer	

29 Arbeitsbereiche | Profiforms R-S Q

Abbildung A

Die Übersichten für Nutzer und Arbeitsbereiche sind identisch aufgebaut.

In Abb. A (3) finden Sie eine Auflistung der bereits angelegten Arbeitsbereiche mit folgenden Informationen:

- **ID:** Die interne, eindeutige ID des Arbeitsbereichs.
- **Name:** Der Name des Arbeitsbereichs.
- **Geändert:** Letzte Änderung an den Einstellungen des Arbeitsbereiches

In Abb. A (4) sehen Sie weitere Informationen zum ausgewählten Arbeitsbereich. Darunter auch, welche Benutzer Zugriff auf den Arbeitsbereich haben.

Sie können in **Abb. A**:

- Arbeitsbereiche **erstellen** und **ändern** (1).
- **Objekte** des Arbeitsbereiches anzeigen lassen (2).

Verwenden Sie die entsprechende Suchleiste (0), um einen Arbeitsbereich zu suchen. Dabei werden alle Spalten in die Suche mit einbezogen. Mehr zur Suchfunktion finden Sie **hier**.

Wählen Sie einen Arbeitsbereich aus, um dieses Objekt zu bearbeiten.

## Arbeitsbereich erstellen und ändern

Verwenden Sie den Erstellen- bzw. Ändern-Button *Abb. A (1)*, um einen neuen **Arbeitsbereich** anzulegen oder den in der Liste ausgewählten Arbeitsbereich zu ändern:

Administration  
Redaktions-System

Nutzer
Arbeitsbereiche
Schriftarten
Binär-Pakete

admin ▾
Einstellungen ▾
Hilfe ▾

Arbeitsbereiche

Erstellen 1
Ändern
Objekte anzeigen

Arbeitsbereiche filtern 0

ID	Name	↓	Geändert
627409	ProfiForms		29.10.20 14:16:25
2193989 <span style="border: 1px solid green; border-radius: 50%; padding: 0 2px;">2</span>	Profiforms_DEV		08.01.21 06:45:38
2193985	Profiforms_QA		08.01.21 06:44:51
707662	Profiforms_Training		08.01.21 06:44:42

Arbeitsbereich Vorschau

ID	2193989
Erstellt	18.02.16 13:22:00
Geändert	08.01.21 06:45:38
Name	Profiforms_DEV
Benutzer	

29 Arbeitsbereiche
Profiforms R-S Q

Abbildung A

Wenn Sie einen neuen Arbeitsbereich anlegen *Abb. A (1)* werden Sie direkt zur Eigenschaftsseite des neu angelegten Arbeitsbereichs geleitet. Durch Ändern *(1)* öffnet sich die Eigenschaftsseite des in der Arbeitsbereichsliste *(2)* ausgewählten Arbeitsbereichs.

Arbeitsbereiche / Ändere Arbeitsbereiche "ProfiForms"

Speichern Speichern und schließen Verlassen

Name

ID 627409  
Erstellt 06.06.13 06:15:44  
Geändert 29.10.20 14:16:25

Benutzer

Benutzer filtern

Alle Nutzer

ID	Vorname	Nachname	Benutzername
1002	admin	admin	admin
104014	Monique	Oehme	Monique

9 Benutzer

Nutzer im Arbeitsbereich

ID	Admin	Vorname	Nachname	Benutzername
57005	<input checked="" type="checkbox"/>	Joey	Lauterbach	Joey

1 Nutzer

Profiforms R-S Q

:

Abbildung B

Wenn Sie einen neuen Arbeitsbereich angelegt haben, sind die Elemente Abb. B (3 & 4) leer (da der Arbeitsbereich noch keine Werte besitzt). Andernfalls enthalten diese Elemente die aktuellen Werte des Arbeitsbereichs, den Sie ändern möchten.

Bei einem bereits bestehenden Arbeitsbereich werden Ihnen folgende Details angezeigt Abb. B (4):

- **ID:** Die interne ID des Arbeitsbereichs .
- **Erstellt:** Der Zeitpunkt, an dem der Arbeitsbereich erstellt wurde.
- **Geändert:** Der Zeitpunkt, an dem der Arbeitsbereich zuletzt geändert wurde.

Wenn Sie einen Arbeitsbereich neu anlegen müssen Sie ihm einen Namen geben Abb. B (3), beim Ändern eines Arbeitsbereichs können Sie hier einen neuen Namen eintragen.

### **Zuweisung von Nutzern**

Über die Nutzerlisten Abb. B (5 & 6) können Sie dem Arbeitsbereich Nutzer zuweisen, die Zugang zu diesem Arbeitsbereich haben sollen. Nur Nutzer, die sich in der Liste "Nutzer im Arbeitsbereich" befinden (6) haben Zugang zu diesem Arbeitsbereich.

- Um einem Nutzer, Zugang zum Arbeitsbereich zu gewähren, wählen sie in der Liste aller Nutzer (5) den gewünschten Nutzer aus und verschieben Sie diesen mit  (7) auf die Liste der Nutzer im Arbeitsbereich.
- Um einem Nutzer den Zugang zum Arbeitsbereich wieder zu entziehen, wählen Sie in der Liste der Nutzer im Arbeitsbereich (6) den gewünschten Nutzer aus und verschieben Sie diesen mit  (7) zurück auf die Liste aller Nutzer.



Wenn Sie in Spalte Admin (8) einen Haken setzen, gewähren Sie dem Nutzer dadurch **Administrator-Rechte** für den betreffenden Arbeitsbereich. Der Nutzer erhält dadurch im entsprechenden Arbeitsbereich alle verfügbaren Rechte.

Verwenden Sie die entsprechende Suchleiste, um einen Arbeitsbereich Abb. A (0) oder Nutzer Abb. B (0) zu suchen. Dabei werden alle Spalten in die Suche mit einbezogen. Mehr zur Suchfunktion finden Sie [hier](#).

Wenn Sie die Bearbeitung beendet haben, speichern Sie den Arbeitsbereich Abb. B (1) (neu erstellte Arbeitsbereiche werden erst dann angelegt) oder verwerfen Sie die Änderungen (2) (neu angelegte Arbeitsbereiche werden wieder verworfen).

## Grafische Objektübersicht

In der **Nutzer und Arbeitsbereiche-Ansicht** können Sie sich für einen ausgewählten Arbeitsbereich eine grafische Übersicht aller Objekte im Arbeitsbereich und deren Beziehungen / Abhängigkeiten untereinander anzeigen lassen. In dieser Übersicht ist es z.B. auch möglich, gelöschte Objekte wiederherzustellen.

Wenn Sie in der Nutzer und Arbeitsbereiche Ansicht in der Arbeitsbereichsliste einen Arbeitsbereich auswählen und dann auf "Objekte anzeigen" klicken (*siehe hier*), öffnet sich die Objektübersicht für den Arbeitsbereich:

The screenshot shows the 'Grafische Objektübersicht' interface. It features a navigation bar at the top with 'Administration docxworld Redaktions-System' and 'Nutzer Arbeitsbereiche Schriftarten Binär-Pakete'. Below the navigation bar, there are breadcrumb links: 'Arbeitsbereiche / Navigiere Arbeitsbereich "Handbuch" / Transaktions-formulare / TForm "Allgemeiner\_Brief" / TForm Revision "Allgemeiner\_Brief" (1.0)'. The interface is divided into three main sections: 1. Object Tree (left), 2. Object Graph (center), and 3. Information Box (right). The Object Tree shows a hierarchy of objects under 'Handbuch' > 'Transaktions-formulare' > 'Allgemeiner\_Brief'. The Object Graph shows a central node 'Allgemeiner\_Brief (1.0) TXR [104036], DEPRECATED' connected to various other nodes. The Information Box displays details for the selected object, including ID, creation date, status, and version.

Abbildung A

Die grafische Objektübersicht untergliedert sich in 3 Hauptbereiche:

- den **Objektbaum** zum Auswählen des gewünschten Objekts Abb. A (1),
- den **Objektgraph**, der Abhängigkeiten und Beziehungen für das ausgewählte Objekt darstellt Abb. A (2),
- die **Informationsbox**, die die wichtigsten Informationen über das ausgewählte Objekt enthält Abb. A (3),

### Objektbaum (1)

Der Objektbaum enthält alle Objekte des Arbeitsbereichs in hierarchischer Ordnung. Wenn Sie die grafische Objektübersicht während einer Sitzung zum ersten Mal laden, sind alle unteren Ebenen des Objektbaums eingeklappt:

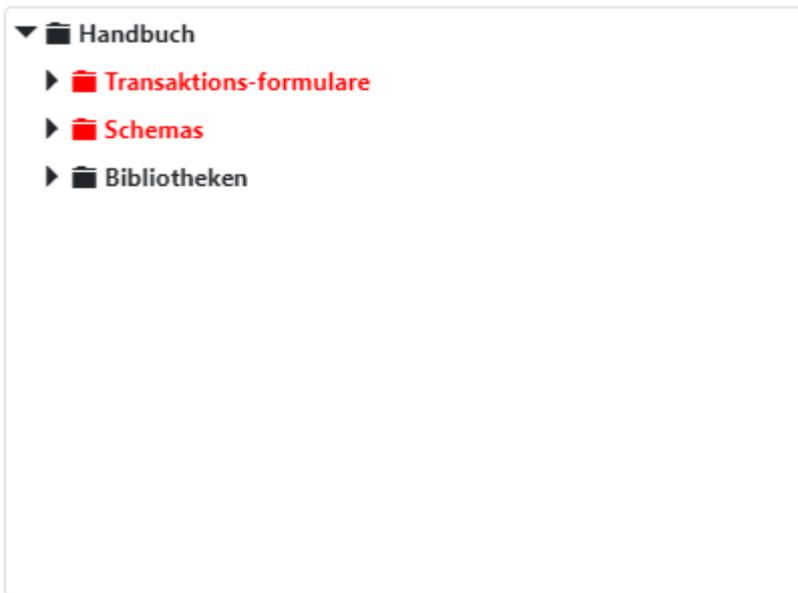


Abbildung B

Sie können jetzt z.B. alle Transaktions-Formulare anzeigen lassen, indem Sie neben "Transaktions-formulare" auf  (expandieren) klicken:

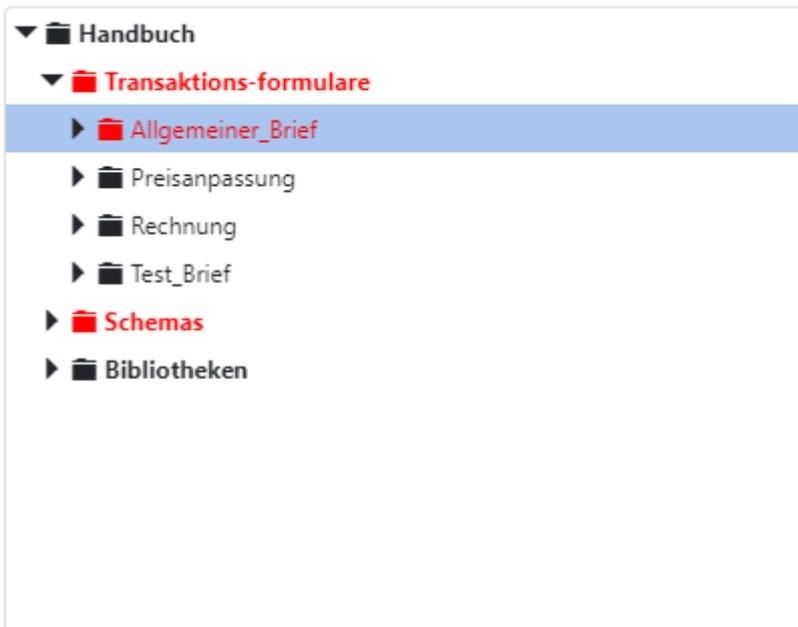


Abbildung C

Sie sehen jetzt alle Transaktions-Formulare im Arbeitsbereich. Wenn Sie nun alle Versionen des Formulars "Allgemeiner\_Brief" sehen wollen, klicken Sie neben "Allgemeiner\_Brief" auf  :

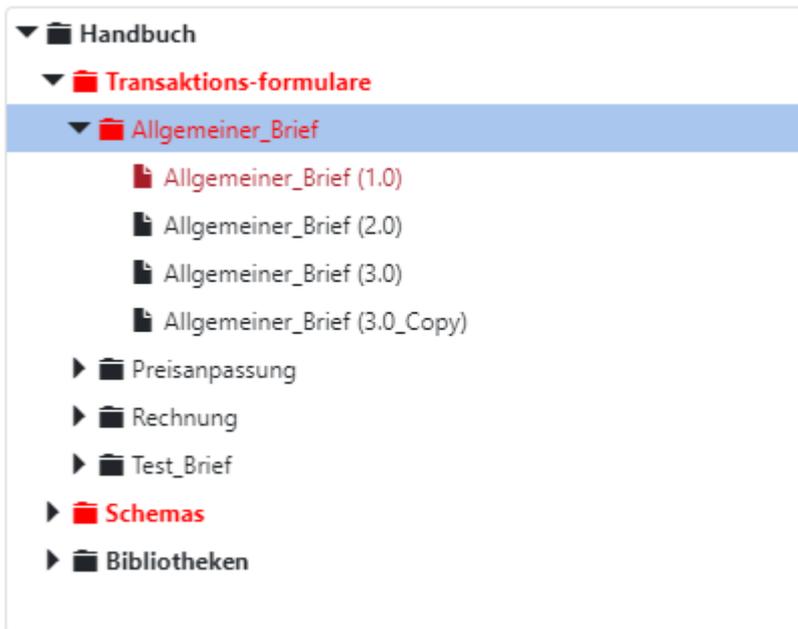


Abbildung D

Bei Schemas und Bibliotheken gehen Sie analog vor. Fett gedruckte Einträge stellen keine Objekte dar, sondern die Objekttypen ("Transaktions-formulare", "Schemas", "Bibliotheken") und können dementsprechend auch nicht ausgewählt (aber natürlich ausgeklappt) werden. Um ein Objekt auszuwählen und dessen Objektgraph anzeigen zu lassen, klicken Sie einfach auf das Objekt im Objektbaum. Die Ebenen im Objektbaum sind wie folgt:

- **Transaktions-formulare**
  - Allgemeiner\_Brief
    - Version 1 (Allgemeiner\_Brief 1.0)
    - Version 2 (Allgemeiner\_Brief 2.0)
    - ...
  - Preisanpassung
    - Version 1 (Formular 2)
    - Version 2 (Formular 2)
    - ...
  - ...
- **Schemas**
  - Briefe
    - Version 1 (Briefe 1.0)
      - Produktions-Paket 1 (PB 104156 Briefe 1.0)
      - Produktions-Paket 2 (PB 104312 Briefe 1.0)
      - ...
    - Version 2 (Briefe 2.0)
      - Produktions-Paket 1 (PB 104169 Briefe 2.0)
      - Produktions-Paket 2 (PB 104224 Briefe 2.0)
      - ...
    - ...
  - Rechnungen
    - Version 1 (Rechnung 1.0)
      - ...
    - Version 2 (Rechnung 2.0)
      - ...
    - ...
  - ...
- **Bibliotheken**
  - CI\_Design
    - Version 1 (CI\_Design 1.0)
    - Version 2 (CI\_Design 2.0)
    - ...
  - Werbeaktion
    - Version 1 (Werbeaktion 2020\_12)
    - Version 2 (Werbeaktion 2021\_01)
    - ...
  - ...



Um die Ebene unterhalb eines Objekts anzuzeigen, klicken Sie auf  . Um die Ebene wieder auszublenden, auf  .

Nachdem Sie das gewünschte Objekt ausgewählt haben, wird es im Objektgraph angezeigt:

### Objektgraph (2)

Der Objektgraph stellt das momentan im Objektbaum ausgewählte Objekt im Zusammenhang zu anderen Objekten dar, zu denen das Objekt in einer Beziehung steht. Außerdem ermöglicht es der Objektgraph, Objekte zu löschen oder gelöschte Objekte wiederherzustellen (mehr dazu später).

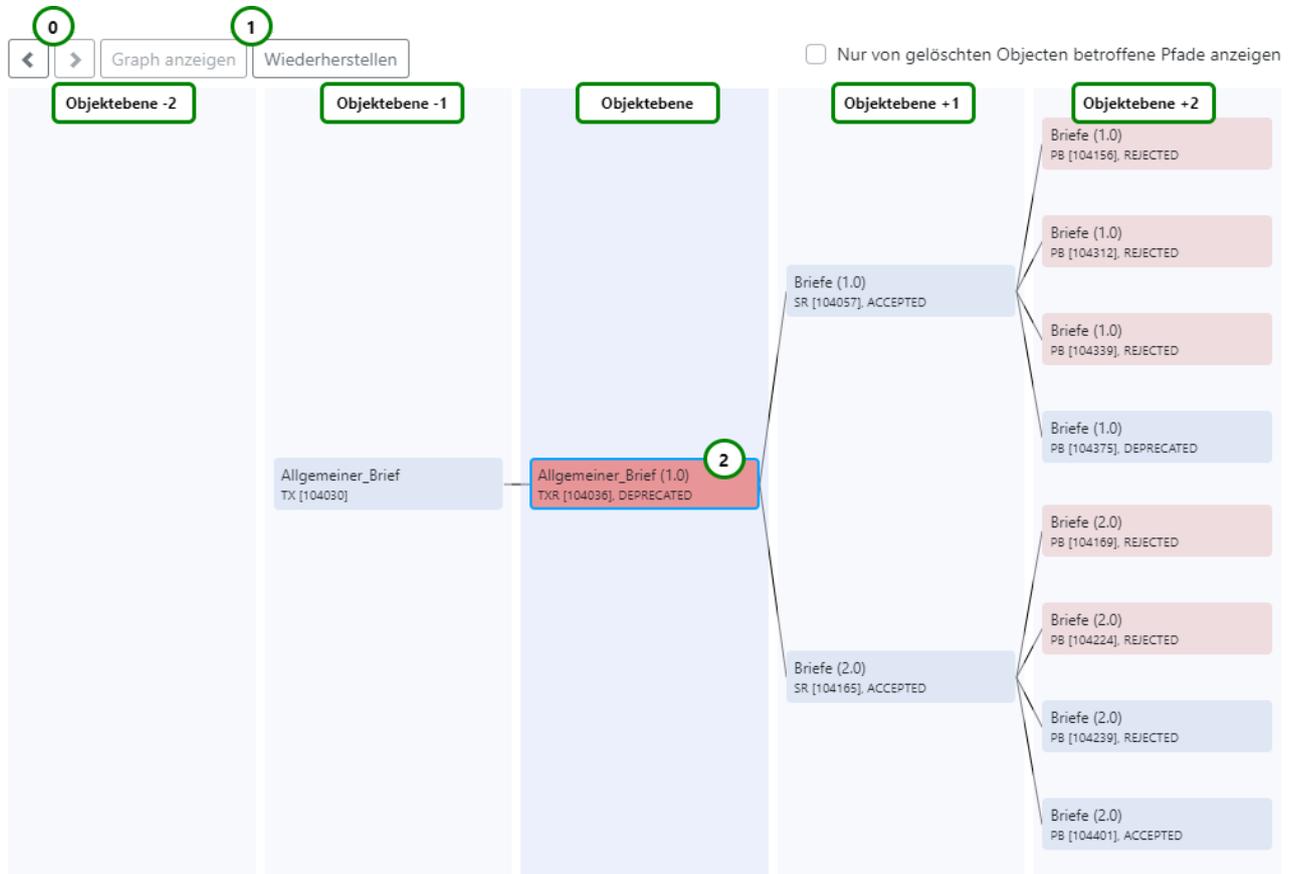


Abbildung E

Das ausgewählte Objekt wird blau umrandet dargestellt *Abb. E (2)*. Die verschiedenen Säulen stellen unterschiedliche Ebenen dar. Dabei stehen Objektversionen immer eine Säule weiter rechts als ihre Objekte, Schemas rechts von den Formularen, die sie einbinden, Produktions-Pakete rechts der Schema-Version, zur der sie gehören und Bibliotheken und Bibliotheks-Versionen links des Schemas, das sie einbindet. Zusammengefasst:

← Im Graph weiter links

Im Graph weiter rechts →

<b>Formulare</b>	<b>Formular-Versionen</b>		
	<b>Schemas</b>	<b>Schema-Versionen</b>	<b>Produktions-Pakete</b>
<b>Bibliotheken</b>	<b>Bibliotheks-Versionen</b>		

Objekte werden im Graph wie folgt dargestellt:

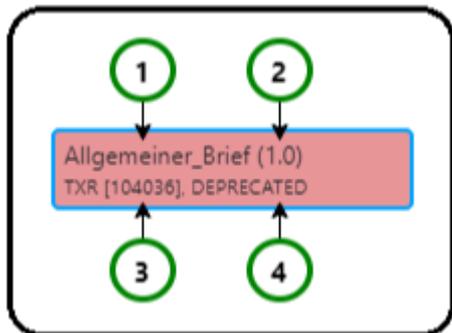


Abbildung F

Ein Objekt im Graph enthält alle nötigen Informationen über das jeweilige Objekt:

- den **Namen** des Objekts *Abb. F (1)* bzw. falls es sich um eine Version handelt, zusätzlich die **Versionsnummer** *Abb. F (2)* der Version (in *Abb. E* dargestelltes Objekt ist Allgemeiner\_Brief 1.0 von Allgemeiner\_Brief).
- den **Objekttyp** *Abb. F (3)*:

<b>TX</b>	Formular
<b>TXR</b>	Formular-Version
<b>S</b>	Schema
<b>SR</b>	Schema-Version
<b>L</b>	Bibliothek
<b>LR</b>	Bibliotheks-Version
<b>PB</b>	Produktions-Paket

- die **ID** des Objekts *Abb. F (3)* in eckigen Klammern.
- den **Status** des Objekts (z.B. **ACCEPTED**, **REJECTED**, **TEST**, etc.) *Abb. F (4)*.

Beziehungen zwischen Objekten, wie "Brief-Version X von Briefe Y" oder Brief-Version X verwendet Allgemeiner\_Brief-Version Y", werden durch Linien zwischen den beiden Objekten dargestellt.

⚠ Es werden immer nur die Objekte angezeigt, die in direkter Beziehung zum ausgewählten Objekt stehen. Deshalb hängt es vom Objekttyp ab, welche Objekttypen angezeigt werden. So wird für eine Schema-Version z.B. das zugehörige Schema und die zugehörige Formular-Version angezeigt, für eine Formular-Version allerdings nur die Schema-Versionen, die die Formular-Version benutzen, nicht aber die Schemas der Schema-Versionen. Die Anordnung der Ebenen (Säulen) in der Ansicht ist nicht fix und hängt vom Objekttyp ab. Wichtig sind die relativen Positionen der Objekte.

Für das in Abb. E gezeigte Beispiel bedeutet das:

Das im Objektbaum ausgewählte Objekt ist die Allgemeiner\_Brief-Version "1.0" (Ebene 0) des Allgemeiner Brief 1 (Ebene -1)

Es gibt 2 Briefe-Versionen, die die Allgemeiner\_Brief "1.0" einbinden (Ebene 1)

Von Briefe-Version 1.0 und Briefe Version 2.0 existieren jeweils 4 Produktions-Paket (Ebene 2).

**i** Sie können den Graph für ein Objekt auch anzeigen lassen, indem Sie auf das entsprechende Objekt im Graphen (in der Graphendarstellung für ein anderes Objekt) doppelklicken oder es per Maus auswählen und dann auf Abb.E(1) klicken, statt es explizit im Objektbaum auszuwählen. Mit Abb. E (0) können Sie zum zuvor ausgewählten Objekt zurückkehren (  ) oder, nachdem Sie zurück navigiert haben, wieder zum danach ausgewählten Objekt gehen (  ).

**i** Eventuell kann es nötig werden, den Graphen für ein Objekt oder die komplette Objektübersicht (inklusive Objektbaum) neu zu laden, beispielsweise, wenn parallel im R-S Client gearbeitet wird und sich Objekte oder Beziehungen verändert haben. Verwenden Sie Abb. E(1), um den Graphen für das ausgewählte Objekt neu zu laden (nur Änderungen, die das Objekt oder eines der Objekte, mit denen es in Beziehung steht, werden angezeigt) oder Abb. A(4), um die komplette Objektansicht zu aktualisieren (alle Änderungen werden angezeigt).

### **Löschen / Wiederherstellen von Objekten**

Es ist auch möglich, R-S Objekte zu löschen. Das Objekt wird dabei aber nicht im eigentlichen Sinne gelöscht, sondern es wird inaktiv ( **INACTIVE** ) gesetzt. Objekte, die inaktiv sind, können nicht verwendet werden (auch nicht von anderen Objekten) und werden auch im R-S Client nicht mehr angezeigt. Inaktive (gelöschte) Objekte werden im Objektgraphen besonders gekennzeichnet.

Objekte können nicht nur durch den Benutzer im R-S Client (siehe [hier](#) ), sondern auch direkt im Objektgraph gelöscht werden:

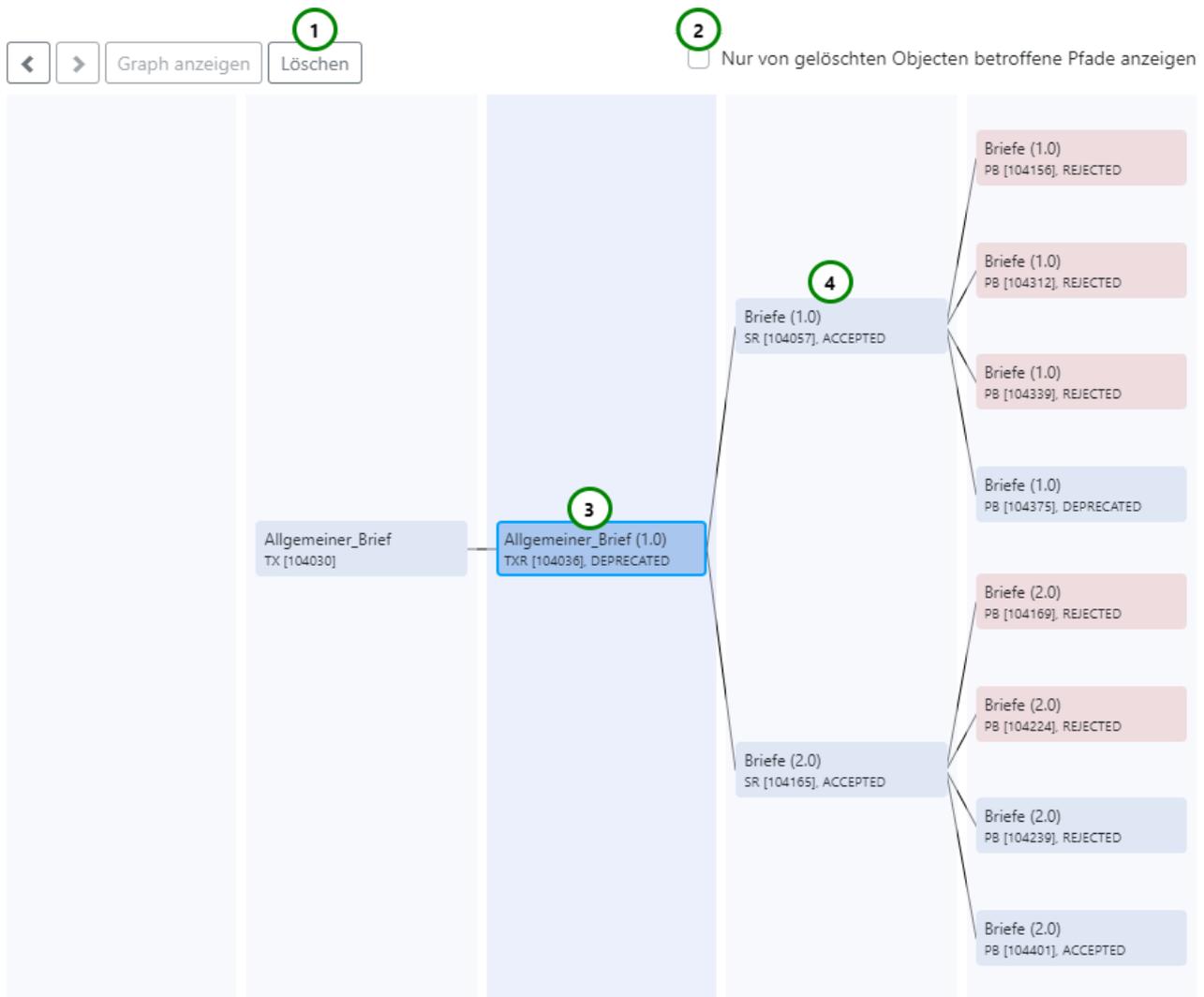


Abbildung G

In Abb. G wird der Objektgraph für Objekt Abb. G (4) dargestellt (erkennbar durch dunkelviolette Hinterlegung), das ausgewählte Objekt ist Abb. G (3) (erkennbar durch blaue Hinterlegung und blauen Rand). Löschen Sie das ausgewählte Objekt mit Abb. G (1). Das Objekt ist danach inaktiv gesetzt ("gelöscht") und deshalb im Objektgraph rot markiert:

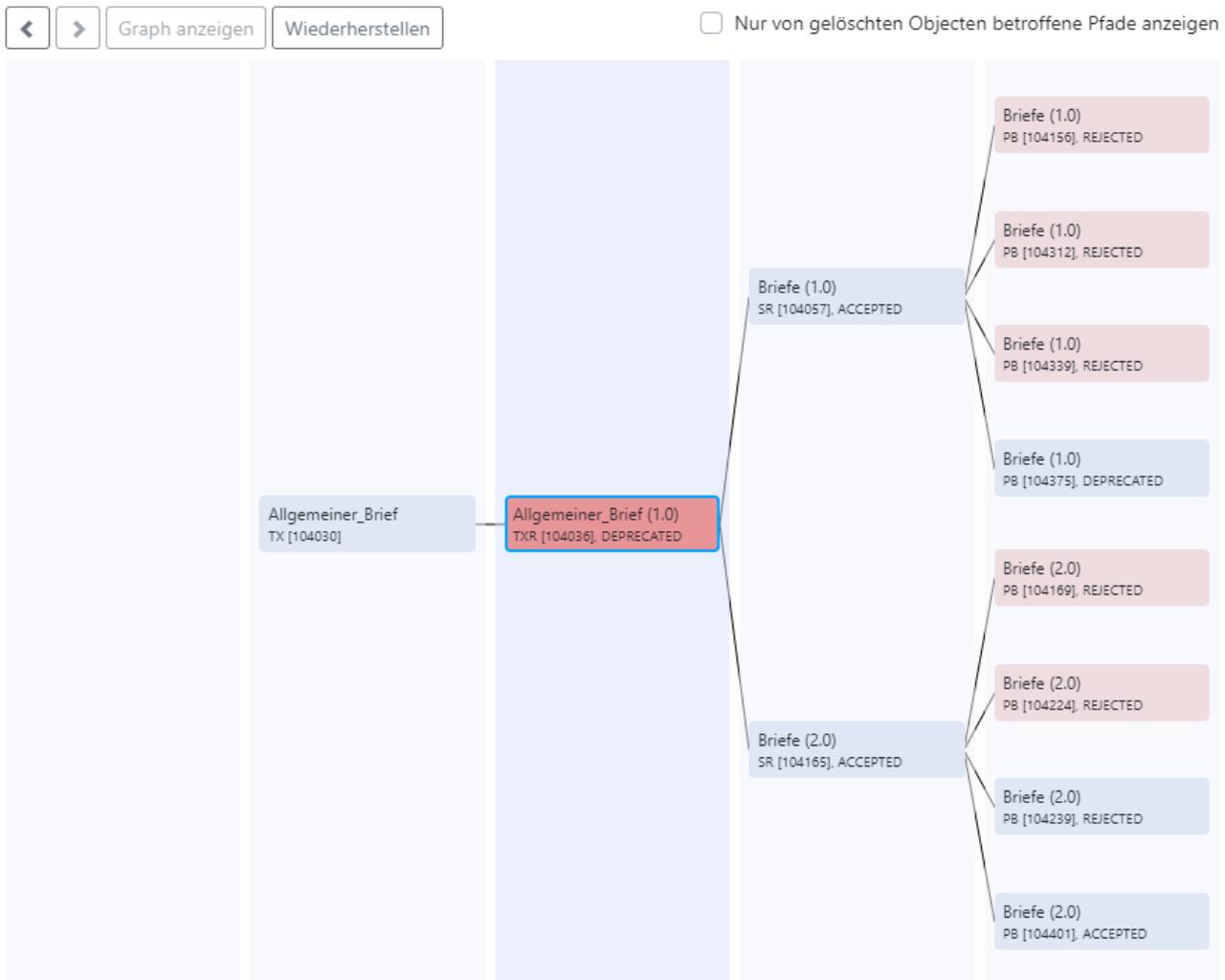


Abbildung H

Rot hinterlegte Objekte sind inaktiv gesetzt ("gelöscht") und werden im R-S Client nicht angezeigt. Inaktive Objekte werden Ihnen auch im Objektbaum durch rote Markierung (auch der übergeordneten Objekte) angezeigt:

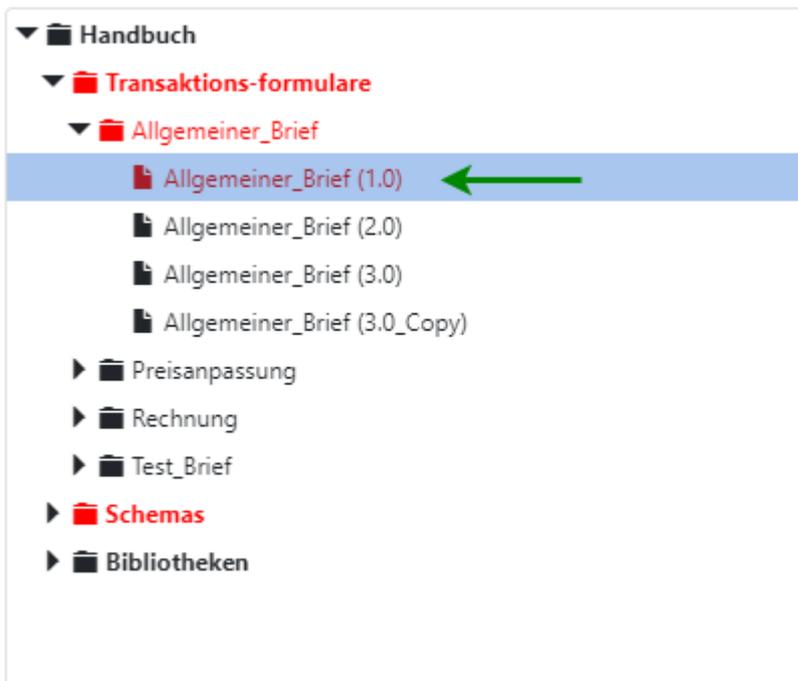


Abbildung I

Dunkelrote Einträge signalisieren, dass dieses Objekt ein Unterobjekt besitzt, das inaktiv gesetzt ist (z.B. Schema mit inaktiver Schema-Version), hellrote Einträge signalisieren, dass das Objekt selbst inaktiv ist.

Es ist nicht möglich ein Objekt zu löschen, das aktive Unterobjekte (Versionen) besitzt (eine Fehlermeldung erscheint (*Abb. J*)). Sie müssen dann zuvor alle Versionen des Objekts einzeln inaktiv setzen, bevor Sie das Objekt selbst inaktiv setzen. Das gilt selbstverständlich auch für das Löschen im *R-S Client*.

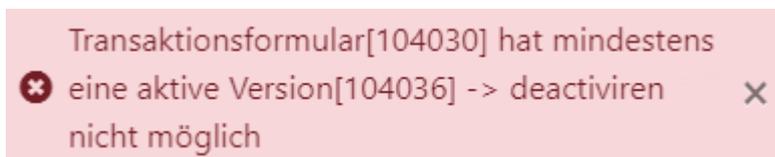


Abbildung J

⚠ Wenn ein Objekt per R-S Client gelöscht wurde wird die Änderung nicht automatisch im R-S Admin-Client angezeigt, Sie müssen zuvor die Ansicht aktualisieren (siehe *oben*).

ℹ Dass ein Objekt inaktiv gesetzt ist lässt sich auch in der Infobox nachvollziehen (siehe *unten*).

Es besteht die Möglichkeit, in einem Objektgraph nur die Objekte anzeigen zu lassen, die in einer direkten Beziehung zu einem gelöschten Objekt stehen (mit dem gelöschten Objekt durch einen Pfad/Linie verbunden sind), also durch die Löschung des Objekts direkt betroffen sind, z.B. weil eine Abhängigkeit zu dem gelöschten Objekt besteht (z.B. Eine eingebundene Bibliotheks-Version oder das Schemaobjekt für eine Schema-Version). Um nicht betroffene Objekte auszublenden, klicken Sie auf *Abb. G (2)*. im Falle von *Abb. G* :

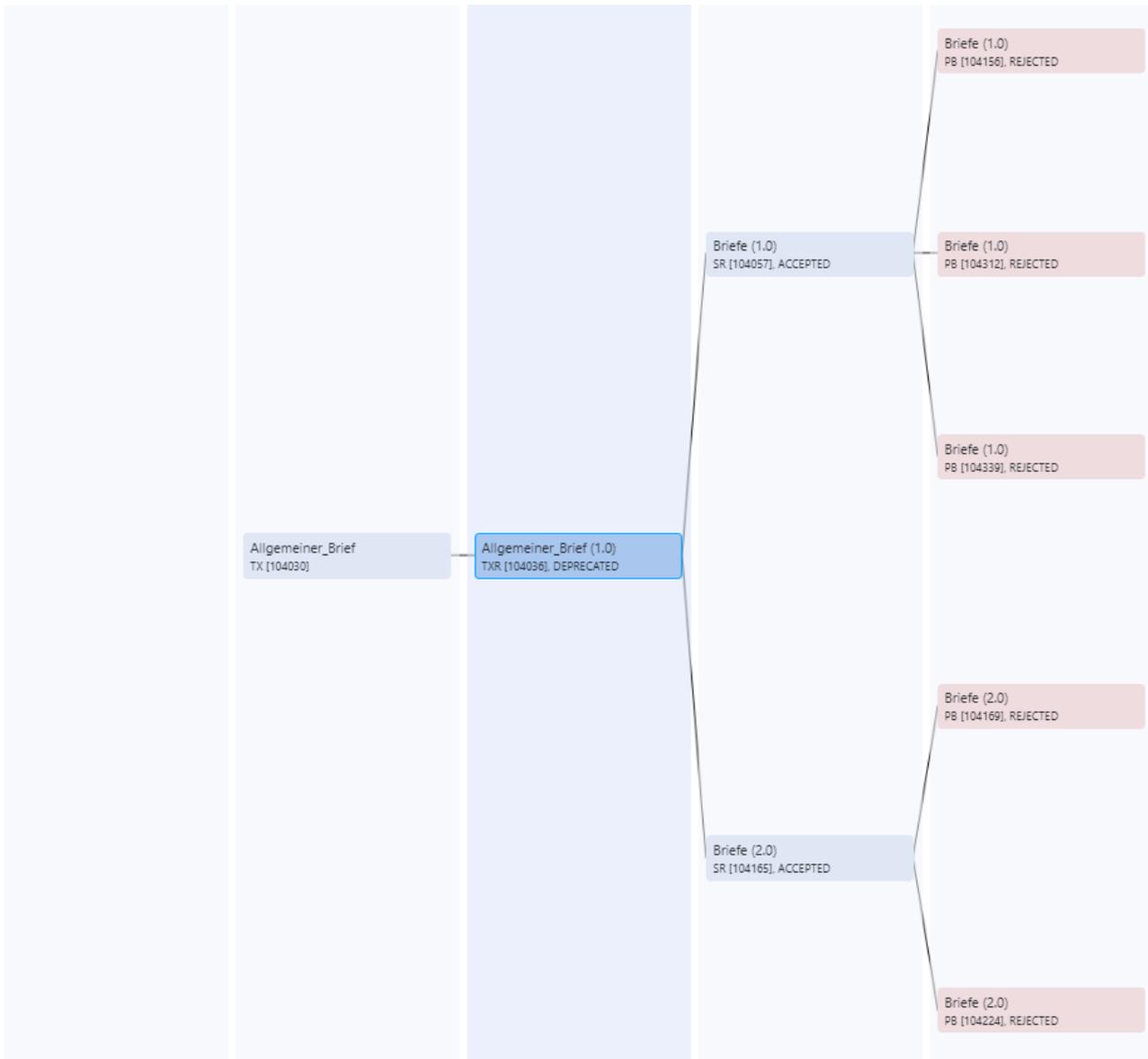


Abbildung K

⚠ Überprüfen Sie für alle Objekte, die von einer Löschung betroffen sind, ob durch die Löschung und damit durch den Wegfall des Objekts leere Referenzen in den anderen Objekten entstehen (z.B. durch das Löschen einer Bibliotheks-Version, die aber noch von einer Schema-Version verwendet wird oder das Löschen einer Formular-Version, die von einer Schema-Version verwendet wird). In diesem Fall müssen Sie das Objekt gegebenenfalls **Grafische Objektübersicht wiederherstellen**, um ungewünschtes Verhalten zu vermeiden.

**!** Wenn kein inaktives Objekt im Objektgraph existiert wird ein leerer Graph angezeigt, da kein Objekt von einer Löschung betroffen ist.

Sie können inaktiv gesetzte ("gelöschte") Objekte auch Wiederherstellen. Wenn Sie dazu ein inaktives Objekt im Objektgraph oder Objektbaum auswählen wird Ihnen statt des "Löschen"-Buttons ein "Wiederherstellen"-Button angezeigt:

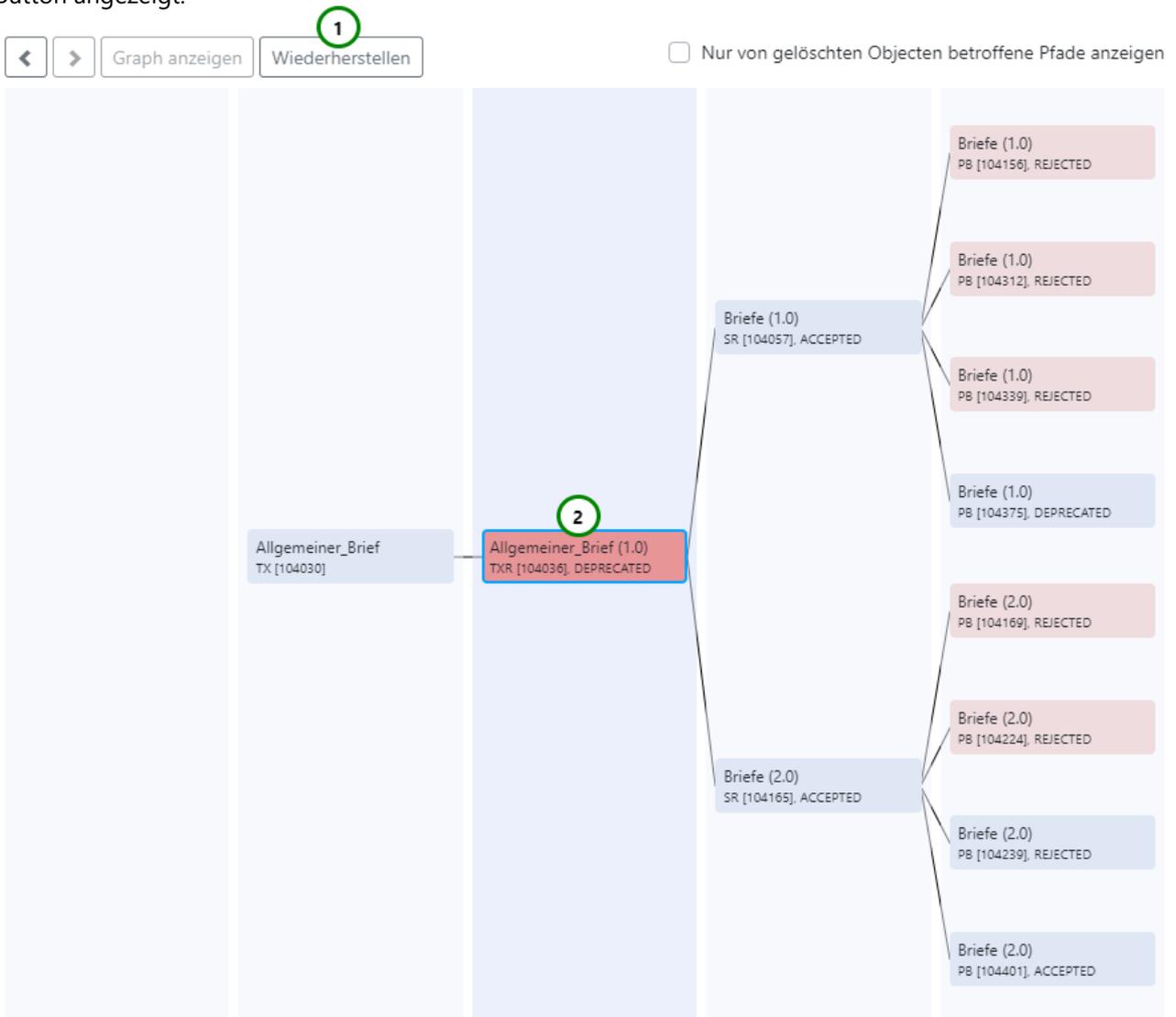


Abbildung L

Stellen Sie das inaktive Objekt Abb. L (2) durch Abb. L (1) wieder her. Das Objekt wird dadurch wieder auf aktiv gesetzt, ist nicht mehr rot hinterlegt und kann wieder ganz normal (auch von anderen Objekten) verwendet werden. Es wird jetzt auch wieder im R-S Client angezeigt.

! Inaktive Objekte werden nur eine bestimmte Zeit lang archiviert, bevor sie endgültig gelöscht werden. Danach ist eine Wiederherstellung des Objekts nicht mehr möglich.

### Informationsbox (3)

In der Informationsbox werden Ihnen die wichtigsten Informationen zu dem ausgewählten Objekt angezeigt. Ist kein Objekt im Objektbaum ausgewählt, wird die Informationsbox ausgeblendet.

TXForm Revision: ACTIVE		← 1
ID	104036	← 2
Erstellt	07.12.20 14:50:35	} 3
Bearbeitet	25.03.21 09:48:01	
Status	DEPRECATED	} 4
Deskriptor	Allgemeiner_Brief	
Version	1.0	

Abbildung M

Die Informationsbox enthält folgende Informationen:

- Ob das Objekt **aktiv oder inaktiv** ist Abb. M (1). Ein Objekt ist immer aktiv, es sei denn, es wurde gelöscht (im R-S durch den Nutzer oder direkt hier), wodurch es auf inaktiv gesetzt wird. Es kann dann nicht mehr verwendet werden und wird auch im R-S nicht mehr angezeigt. Wenn ein Objekt wiederhergestellt wird, wird es auch wieder aktiv gesetzt (also: Aktiv = Objekt existiert, ganz normale Verwendung; Inaktiv = Objekt gelöscht),
- Die **ID** des Objekts (Vgl. auch "Erweiterter Modus") Abb. M (2),
- Wann das Objekt **erstellt** und wann **zuletzt bearbeitet** wurde Abb. M (3),
- Bei Versionen: **Status**, **Deskriptor** (Bezeichnung) und **Versionsnummer** des Objekts und bei Produktions-Paketen zusätzlich das **zugehörige Schema** Abb. M (4), sonst unter Abb. M (4) nur den **Objektnamen**.

## Schriftarten

Um Schriftarten zu verwalten, wählen Sie in der Navigation Schriftarten aus. Sie gelangen dann zu folgender Übersicht:

Schriftarten

Arbeitsbereiche filtern 0

ID	Name	↓↑ Geändert
627409	ProfiForms	29.10.20 14:16:25
2193989	Profiforms_DEV	08.01.21 06:45:38
2193985	Profiforms_QA	08.01.21 06:44:51
707662	Profiforms_Training	08.01.21 06:44:42

Schriftarten für Arbeitsbereich "ProfiForms"

2
Erstellen
Ändern
Filtern Sie Schriftarten

Name	↓↑ Stile	Geändert	Beschreibung
Arial	A <sup>*</sup> B I I	08.01.21 10:13:...	
Tahoma	A <sup>*</sup> B I I	08.01.21 10:13:...	

3

2 Schriftarten

Abbildung A

Schriftarten werden in Bibliotheken benötigt, da Bibliotheken nicht an ein Transaktions-Formular gebunden sind und somit nicht, wie direkt in einer Schema-Version erstellte Ressourcen, die Schriftarten des Transaktions-Formulars erben. Da Bibliotheken nicht wissen, welche Fonts später verfügbar sein werden (Da dies von Transaktions-Formular zu Transaktions-Formular unterschiedlich sein kann.) müssen Sie diese angeben. Angelegte Schriftarten sind immer in allen Bibliotheken des Arbeitsbereichs, für den die Schriftart erstellt wurde, verfügbar.

Wählen Sie in der Arbeitsbereichsliste Abb. A (1) den Arbeitsbereich aus, für den Sie Schriftarten verwalten möchten. Ihnen werden dann in der Schriftartenübersicht (3) alle in dem ausgewählten Arbeitsbereich vorhandenen Schriftarten angezeigt.

In der Schriftartenübersicht wird Ihnen zu jeder Schriftart folgende Informationen angezeigt:

- **Name** des Fonts.
- **Erlaubte Stile** des Fonts: Nur wenn die entsprechende Option **fett** hinterlegt ist, darf dieser Font auch in der entsprechenden Ausprägung angewandt werden.
- **Beschreibung** des Fonts.
- **Änderungsdatum** des Fonts.

Verwenden Sie die entsprechende Suchleiste, um einen Arbeitsbereich oder eine Schriftart Abb. A (0) zu suchen. Dabei werden alle Spalten in die Suche mit einbezogen. Mehr zur Suchfunktion finden Sie [hier](#).

### **Schriftart erstellen oder ändern**

Wenn Sie eine neue Schriftart erstellen oder eine vorhandene ändern Abb. A (2) gelangen Sie zu folgender Übersicht:

Abbildung B

Wenn Sie eine neue Schriftart angelegt haben, sind die Elemente Abb. B (3) leer (Da die Schriftart noch keine Werte besitzt.). Andernfalls enthalten diese Elemente die aktuellen Werte der Schriftart, die Sie ändern möchten.

Wenn Sie eine Schriftart neu angelegt haben, müssen sie dieser zwingend einen Namen geben (4). Beim Ändern einer Schriftart müssen Sie den Namen nicht zwingend ändern.

Einmal erstellte Schriftarten können nicht wieder gelöscht werden!

Weitere Informationen, die Ihnen beim Ändern einer Schriftart angezeigt werden Abb. B (3):

- **ID** : Die interne ID der Schriftart.
- **Arbeitsbereich** : Der Arbeitsbereich, in dem die Schriftart verfügbar ist.
- **Erstellt** : Der Zeitpunkt, an dem die Schriftart erstellt wurde.
- **Geändert**: Der Zeitpunkt, an dem die Schriftart zuletzt geändert wurde.

Eine Schriftart kann in verschiedenen Ausprägungen verwendet werden. Sie können festlegen, welche Stile für diese Schriftart erlaubt sein sollen Abb. B (5):

- Regulär (*normal*)
- **Fett**
- *Kursiv*
- **Fett kursiv**



Sie müssen mindestens einen Stil erlauben (i.d.R. mindestens Regulär), da die Schriftart sonst nicht verwendet werden kann.

Außerdem können Sie dem Font eine Beschreibung geben Abb. B (6), die auch im R-S Client (z.B. **Schriftarten-Übersicht**) angezeigt wird.

Wenn Sie die Bearbeitung beendet haben, speichern Sie die Schriftart (1) (Neu erstellte Schriftarten werden erst dann angelegt.) oder verwerfen Sie die Änderungen (2) (Neu angelegte Schriftarten werden wieder verworfen.).

## Binärpakete

Um Binär-Pakete zu verwalten, wählen Sie in der Navigation Binär-Pakete aus. Sie gelangen dann zu folgender Übersicht:

The screenshot shows the 'Binär-Pakete' management interface. At the top, there are buttons for 'Erstellen' (1), 'Ändern' (2), 'Löschen' (3), and 'Herunterladen' (4). A search bar labeled 'Filterpakete' (0) is also present. Below these is a table with columns: Name, Ausführu..., Plattform, Erstellt, and Bearbeitet. The table lists multiple 'OMS-ReportWriter' packages with various versions and dates. One package (version 5.9.1002) is highlighted in blue, with a callout (5) pointing to it. To the right of the table is a preview pane titled 'Vorschau des Binär-Pakete' (6) showing details for the selected package, including ID (2093114), creation and modification dates, name, platform, execution version, and description.

Abbildung A

Binär-Pakete enthalten die ausführbaren Programme zur Aufbereitung der Dokumente für die Druckvorschau und zur Generierung der Druckdaten in der produktiven Umgebung. Mehr Informationen über Binär-Pakete finden Sie [hier](#).

In der Binär-Pakete-Ansicht Abb. A (5) werden Ihnen alle vorhandenen Pakete aufgelistet:

- **Name** des Programms.
- **Version** des Programms.
- **Plattform**, für die das Programm bestimmt ist.
- **Erstelldatum**.
- Wann das Binärpaket **zuletzt geändert** wurde.

Verwenden Sie die entsprechende Suchleiste Abb. A (0), um ein Binär-Paket zu suchen. Dabei werden alle Spalten in die Suche mit einbezogen. Mehr zur Suchfunktion finden Sie [hier](#).

Wenn Sie ein Binär-Paket neu erstellen Abb. A (1) müssen Sie für das Paket das entsprechende Binär-Paket (.zip-Datei) hochladen:



Abbildung B

Klicken Sie auf Abb. B (1), um ein Binär-Paket-Archiv hochzuladen. Nach erfolgreichem Abschluss des Ladevorgangs wird Ihnen die Meldung "Erfolgreich hochgeladen" angezeigt. Sie können es nun Speichern Abb. C (1) oder wieder verwerfen (2).

Kann das Archiv nicht geladen werden, z.B. weil es unvollständig oder beschädigt ist, wird Ihnen dies durch eine Fehlermeldung nach dem Laden oder beim Speichern angezeigt.



Abbildung C

Wenn Sie ein, in der Paketliste Abb. A (5) ausgewähltes, Binär-Paket ändern (2) werden Ihnen zusätzlich die Paketinformationen angezeigt:

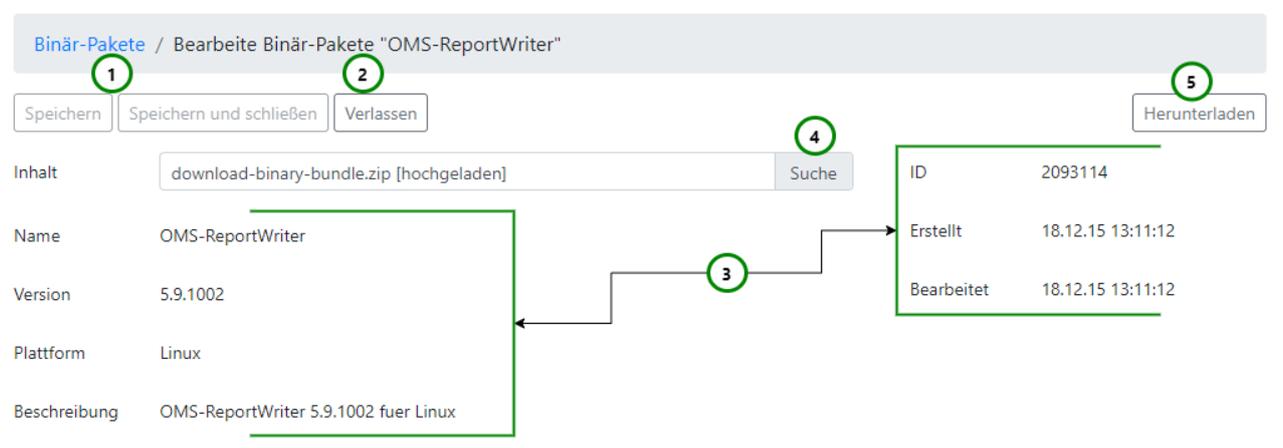


Abbildung D

Der Übersicht können Sie folgende Paketinformationen entnehmen [Abb. D \(3\)](#):

- **ID**, welche intern dem Paket zugeordnet ist.
- **Name** des Programms aus dem Binär-Paket-Archiv.
- Wann das Paket **erstellt** wurde und
- wann das Paket **zuletzt geändert** wurde.
- Die **Version** des Programms aus dem Binär-Paket.
- Die **Plattform**, für die das Binär-Paket bestimmt ist (z.B. Windows oder Linux) und
- optional eine **Beschreibung** des Archivs.

Wenn Sie dem Paket ein neues Binär-Paket-Archiv zugewiesen haben [Abb. D \(4\)](#) müssen Sie die Änderung speichern (1). Mit (2) verwerfen Sie die Änderung und das alte Paket wird nicht überschrieben. Zusätzlich können Sie das zugewiesene Archiv auch wieder auf ihren lokalen Speicher herunterladen (5) oder [Abb. A \(4\)](#).

Wenn Sie ein Binär-Paket wieder löschen möchten, wählen Sie zuvor das betreffende Paket in [Abb. A \(5\)](#) aus und klicken Sie dann auf (3).



Das Löschen von Binär-Paketen kann zu ungewünschtem Verhalten führen, wenn das gelöschte Binär-Paket zuvor im R-S als Abhängigkeit benutzt wurde. Vergewissern Sie sich deshalb vorher, dass das Paket wirklich nicht mehr gebraucht wird.

## Administrator-Client Konsole

 Dieser Artikel beschreibt die Verwendung des Administrator-Clients per Befehlszeilentool. Seit Version 2.1.1000 kann der Administrator-Client auch über eine grafische Oberfläche direkt aus dem Browser heraus gesteuert werden, siehe [Administrator-Client UI](#).

Das R-S kann über einen nicht grafischen Administrator-Client(Konsole) von der Kommandozeile des Betriebssystems in Grundfunktionen administriert werden. Solche Grundfunktionen sind u.a.:

- R-S Nutzer anlegen
- für R-S Nutzer das Passwort ändern
- Arbeitsbereiche anlegen
- Arbeitsbereichen R-S Nutzer zuordnen
- Produktions-Pakete aus dem R-S abrufen
- Verwaltung von Binäry-Paketen
- eine Liste der aktiven Nutzer abrufen
- Abruf von Version-Informationen des R-S Clients und Servers
- Verwaltung von Lock-Objekten
- Prüf-Routinen zum Testen der Lock-Algorithmen
- Abruf von Import-Paketen von Transaktions-Formularen
- Systemaufgaben wie Reorganisation des Datei-Repository

Voraussetzung für den Start des Administrator-Client(Konsole) ist:

- die Verfügbarkeit eines JAVA JRE ab der Version 1.6 auf dem Server wo der Administrator-Client(Konsole) ausgeführt wird
- eine installierte Version des Administrator-Client(Konsole)
- eine korrekt definierte Konfigurationsdatei 'rs.client.properties' des Administrator-Client(Konsole)

### Konfigurationsdatei für den Administrator-Client(Konsole)

Die Konfiguration befindet sich in der Datei 'rs.client.properties', welche im gleichen Verzeichnis abgelegt sein muss, aus welchem der Administrator-Client(Konsole) gestartet wird. Diese Konfigurationsdatei entspricht im Aufbau einer [Java-Properties-Datei](#). Folgende Einträge sind in dieser Konfigurationsdatei enthalten:

Eintrag	Wert	Beschreibung
rs.server.protocol	http https	Beschreibt das Protokoll, mit welchem auf das R-S zugegriffen werden soll
rs.server.host	xxx.xxx.xxx.xxx localhost	Die IP-Adresse oder der Host-Name des Servers auf welchem das R-S arbeitet

Eintrag	Wert	Beschreibung
rs.server.port	8080	Ist der TCP/IP-Port unter welchem das R-S mit entfernten Applikationen kommuniziert.
rs.server.user	admin	Ist der Nutzer-Name des System-Administrators des R-S (standardmäßig 'admin')
rs.server.password	admin	Ist das Passwort des System-Administrators des R-S (standardmäßig 'admin')
rs.client.workdir	runtime	Temporäres Verzeichnis 'unterhalb' des Arbeitsverzeichnisses des Administrator-Client(Konsole)

### Start des Administrator-Client(Konsole)

Die Kommandozeile des Administrator-Client(Konsole) lautet wie folgt:

```
PATH_JRE\jre\bin\java -Djava.ext.dirs=lib;PATH_JRE/jre/lib/ext -cp config;rs-client.jar de.profiforms.rs.client.admin.AdminClientStarter KOMMANDO
```

Wobei `PATH_JRE` der Pfad zur `JAVA JRE` ist und mit `KOMMANDO` die auszuführenden Befehle des Administrator-Client(Konsole) gemeint sind.

⚠ Wir bei Ihnen nach Eingabe der Kommandozeile folgender Warnhinweis ausgegeben:  
 ⚠ "log4j:WARN no appenders could be found for logger (org.springframework.core.io.support.PathMatchingResourcePatternResolver). log4j:WARN Please initialize the log4j system properly.  
 verwenden Sie statt obiger Kommandozeile:  
 ✓ `PATH_JRE\jre\bin\java -Dlog4j.configuration=file:AC_PATH/log4j.xml -Djava.ext.dirs=lib;PATH_JRE/jre/lib/ext -cp config;rs-client.jar de.profiforms.rs.client.admin.AdminClientStarter KOMMANDO`  
 wobei `AC_PATH` der Pfad zur Administrator-Konsole ist.

**i** Damit Umlaute oder Sonderzeichen bei der Eingabe über die Kommandozeile berücksichtigt werden, kann direkt hinter `..\bin\java` die verwendete Codepage angegeben werden. Unter dem Telnet/SSH-Client PUTTY für UNIX ist die Codepage z.B. wie folgt zu setzen: `-Dsun.jnu.encoding=iso-8859-1`  
`-Dfile.encoding=iso-8859-1`  
*Im PUTTY kann die Codepage auch über die Konfiguration WINDOW/Translation eingestellt werden.*

### Kommandos des Administrator-Client(Konsole)

Starten Sie den Administrator-Client(Konsole) ohne ein Kommando, so erhalten Sie eine Liste der möglichen Kommandos als Hilfe angeboten. Hier eine Auflistung der Kommandos mit Kurzbeschreibung:

Kommando	Parameter	Kommando-Beschreibung	Parameter-Beschreibung
-createUser	<user-email/ID> <password> <firstname> <lastname>	Erzeugt eines R-S Nutzer ohne Berechtigung auf Arbeitsbereiche.	<user-email/ID> Ist der Nutzernamen des R-S Nutzers im R-S <password> Ist das Passwort des R-S Nutzers <firstname> Ist der Vorname des R-S Nutzers <lastname> Ist der Familienname des R-S Nutzers
-changePassword	<user-email/ID> <oldpassword> <newpassword>	Ändert das Passwort für einen bestehenden R-S Nutzer.	<user-email/ID> Ist der Nutzernamen des R-S Nutzers im R-S <oldpassword> Ist das bisherige Passwort des R-S Nutzers <newpassword> Ist das neue Passwort des R-S Nutzers
-resetPassword	<user-email/ID> <newpassword>	Setzt das Passwort für einen bestehenden R-S Nutzer zurück.	<user-email/ID> Ist der Nutzernamen des R-S Nutzers im R-S <newpassword> Ist das neue Passwort des R-S Nutzers.
-createWorkspace	<name>	Erzeugt einen Arbeitsbereich mit dem angegebenen Namen im R-S.	<name> Names des Arbeitsbereiches

Kommando	Parameter	Kommando-Beschreibung	Parameter-Beschreibung
-assignWorkspace	<workspace-name> <user-email/ID>	Weist einen existierenden R-S Nutzer einem Arbeitsbereich zu.	<workspace-name> Name des Arbeitsbereiches, welchem der R-S Nutzer zugeordnet werden soll <user-email/ID> Ist der Nutzernamen des R-S Nutzers im R-S
-assignWorkspaceAdmin	<workspace-name> <user-email/ID>	Weist einem existierenden Arbeitsbereich einen R-S Nutzer mit administrativen Rechten zu.	<workspace-name> Name des Arbeitsbereiches, welchem der R-S Nutzer als Administrator zugeordnet werden soll <user-email/ID> Ist der Nutzernamen des R-S Nutzers im R-S
-unassignWorkspace	<workspace-name> <user-email/ID>	Entfernt einen R-S Nutzer aus dem angegebenen Arbeitsbereich.	<workspace-name> Name des Arbeitsbereiches, aus welchem der R-S Nutzer entfernt werden soll <user-email> Ist der Nutzernamen des R-S Nutzers im R-S
-renameWorkspace	<old-workspace-name> <new-workspace-name>	Benennt einen angegebenen Arbeitsbereich um.	<old-workspace-name> Name des Arbeitsbereiches welcher umbenannt werden soll <new-workspace-name> der neue Name des Arbeitsbereiches
-fetchTXData	<destination-dir> <tx form revision id>	Führt ein Download für ein Transaktions-Formular mit der angegebenen ID in das angegebene Verzeichnis durch.	<destination-dir> Verzeichnis in welches das Transaktions-Formular abgelegt werden soll <tx form revision id> ID des Transaktions-Formulars im R-S
-fetchProductionBundleID	<destination-dir> <bundle-id>	Führt ein Download für ein Produktions-Paket mit der angegebenen ID in das angegebene Verzeichnis durch.	<destination-dir> Verzeichnis in welches das Produktions-Paket abgelegt werden soll <bundle-id> ID des Produktions-Paketes im R-S

Kommando	Parameter	Kommando-Beschreibung	Parameter-Beschreibung
-fetchProductionBundl eDW	<destination-dir> <contract>	Führt ein Download für ein Produktions-Paket mit der angegebenen Verfahrensnummer aus dem docxworld-System in das angegebene Verzeichnis durch.	<destination-dir> Verzeichnis in welches das Produktions-Paket abgelegt werden soll <contract> Verfahrensnummer von docxworld
-fetchProductionBundl eLink	<destination-dir> <name>	Führt ein Download für ein Produktions-Paket mit dem angegebenen Namen aus dem OMS-/docxworld-System in das angegebene Verzeichnis durch.	<destination-dir> Verzeichnis in welches das Produktions-Paket abgelegt werden soll <contract> Name des Produktions-Paketes aus der OMS-/docxworld-Umgebung
-listUsers		Zeigt alle im System vorhandenen R-S Nutzer an.	
-listActiveUsers		Zeigt alle im System aktuell angemeldeten R-S Nutzer an.	
-listWorkspaces		Zeigt alle im System vorhandenen Arbeitsbereiche an.	
-listWorkspaceUsers	<workspace-name>	Zeigt alle einem Arbeitsbereich zugeordneten R-S Nutzer an.	<workspace-name> Name des Arbeitsbereiches, für welchen die R-S Nutzer angezeigt werden soll
-listUserWorkspaces	<user-email>	Zeigt alle einem R-S Nutzer zugeordneten Arbeitsbereiche an.	<user-email> Der Nutzernamen des R-S Nutzers, für den die Arbeitsbereiche angezeigt werden sollen.
-scheduleGC		Startet den Reorganisations-Lauf für das Datei-Repository des R-S.	
-scheduleStorageInfoU pdate		Startet die Abfrage für Informationen über den Status des Datei-Repositories und gibt diese auf Konsole aus.	
-scheduleUserLocksCol lector		—	

Kommando	Parameter	Kommando-Beschreibung	Parameter-Beschreibung
-scheduleDBBackup		—	
-createBinaryBundle	<binary-file>	Generiert ein Binär-Paket im R-S, welches zur Abarbeitung von Produktions-Paketen vorhanden sein muss.	<binary-file> Dateiname des BinaryBundles.
-fetchBinaryBundle	<destination-dir> <bundleid>	Ruft ein Binär-Paket aus dem R-S ab und legt es im angegebenen Verzeichnis ab.	<destination-dir> Verzeichnis in welches das BinaryBundle abgelegt werden soll <bundle-id> ID des BinaryBundles im R-S
-deleteBinaryBundle	<bundle-id>	Entfernt ein Binär-Paket aus dem R-S.	<bundle-id> Die ID des BinaryBundles, das aus dem R-S entfernt werden soll.
-listBinaryBundles		Zeigt alle im System vorhandenen Binär-Pakete an	
-listAllLocks		Zeigt alle im System vorhandenen Locks an.	
-removeLock	<lock-id>	Löscht einen im System vorhandenen Lock.	<lock-id> Die ID des Locks, das aus dem System entfernt werden soll.
-removeAllLocks		Löscht alle im System vorhandenen Locks. <b>Wichtiger Hinweis</b> : Damit können aktuelle Änderungen von Nutzern des R-S verloren gehen. Besser ist der Befehl '-removeLock' zu wählen, um ausgewählte Locks zu löschen.	
-testSchemaRevisionConcurrentModifications	<schema-revision-id> <threads-number> <iteration-numbers>	Führt eine Prüfroutine zum Test von konkurrierenden Zugriffen auf das angegebene Schema-Version-Objekt durch.	<schema-revision-id> ID der Schema-Version welche getestet werden soll <threads-number> Anzahl der gleichzeitig simulierten Nutzer <iteration-numbers> Anzahl der Zugriffs-Versuche pro simulierten Nutzer

Kommando	Parameter	Kommando-Beschreibung	Parameter-Beschreibung
-testTXFormRevisionConcurrentModifications	<txform-revision-id> <threads-number> <iteration-numbers>	Führt eine Prüfroutine zum Test von konkurrierenden Zugriffen auf das angegebene Transaktions-Formular-Version-Objekt durch.	<txform-revision-id> ID der Transaktions-Formular-Version welche getestet werden soll <threads-number> Anzahl der gleichzeitig simulierten Nutzer <iteration-numbers> Anzahl der Zugriffs-Versuche pro simulierten Nutzer
-clientVersion		Zeigt die Version des aktuell installierten R-S Clients an.	
-serverVersion		Zeigt die Version des aktuell installierten R-S Servers an.	
-serverInfo		Zeigt Informationen über den R-S Server an.	
-listFonts	<workspace-name>	Zeigt alle einem Arbeitsbereich zugeordneten Schriftarten an. Diese Schriftarten werden für die Verwendung in Bibliotheken benötigt.	<workspace-name> Name des Arbeitsbereiches, für welchen die Workspace-Schriftarten angezeigt werden sollen

Kommando	Parameter	Kommando-Beschreibung	Parameter-Beschreibung
-createFont	<name> <description> <work space-name> <regular(y/n)> <bold(y/n)> <italic(y/n)> <bolditalic(y/n)>	Registriert eine Schriftart für den Arbeitsbereich, welche durch Ressourcen von Bibliotheken genutzt werden kann. Hinweis: In Schema-Versionen werden diese Fonts nicht zur Auswahl angeboten.	<name> Name der Schriftart <description> Eine Kurzbeschreibung für die Schriftart. <workspace- name> Name des Arbeitsbereiches, in welchen die Workspace- Schriftarten registriert werden soll <regular(y/n)> Bestimmt, ob für die Schriftart das Attribut 'normal' angeboten werden soll <bold(y/n)> Bestimmt, ob für die Schriftart das Attribut 'fett' angeboten werden soll <italic(y/n)> Bestimmt, ob für die Schriftart das Attribut 'kursiv' angeboten werden soll <bolditalic(y/n)> Bestimmt, ob für die Schriftart das Attribut 'fett und kursiv' angeboten werden soll

## Installation

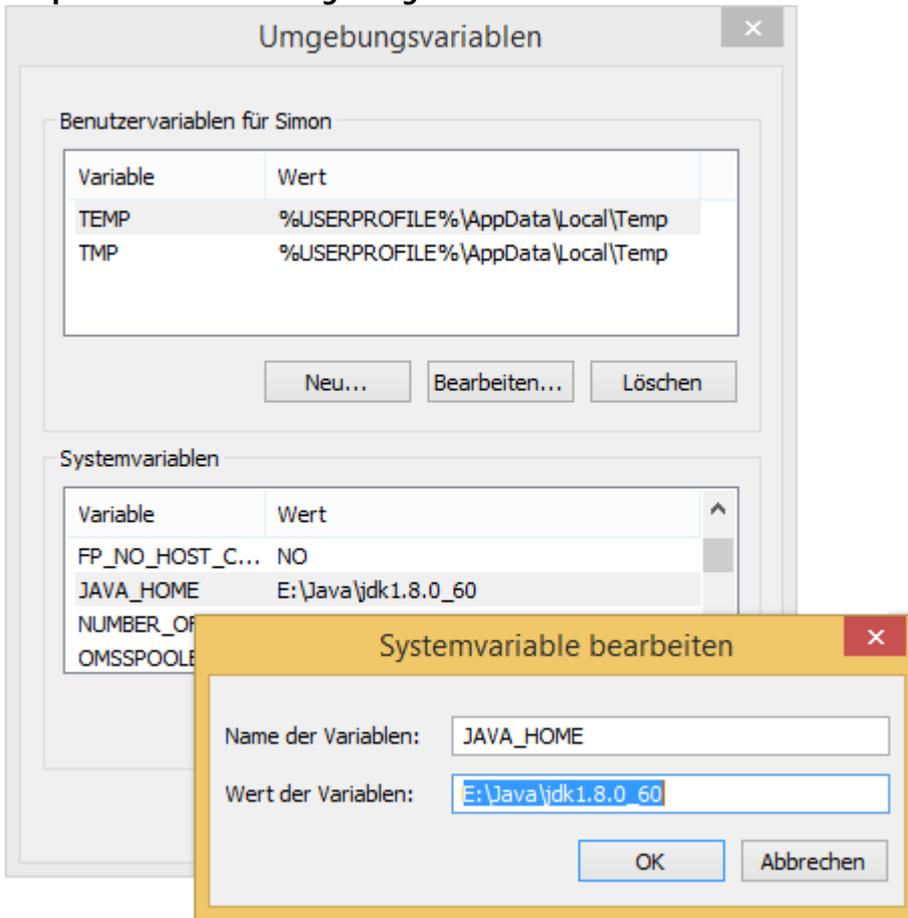
### Umgebungsvariablen Applikations-Server Apache Tomcat

Zum erfolgreichen Start des Applikations-Servers Apache Tomcat werden die Umgebungsvariablen `JAVA_HOME` und `CATALINA_HOME` benötigt. Die Variable **CATALINA\_HOME** zeigt auf das Installationsverzeichnis des Applikations-Servers Apache Tomcat. Ein Beispiel für Windows: `D:\RS\apache-tomcat-8`. Die Variable `JAVA_HOME` zeigt auf ein JDK (Java Development Kit) in der benötigten Version (Weitere Hinweise finden Sie hier: [R-S Server Systemvoraussetzungen](#)).

 Es gibt mehrere Möglichkeiten, Umgebungsvariablen im Betriebssystem oder für den Nutzer zu setzen, welcher den Applikations-Server startet. Sprechen Sie hier mit Ihrem Systemadministrator.

 Durch den Applikations-Server Apache Tomcat wird auch geprüft, ob die Umgebungsvariable **JRE\_HOME** existiert. Ist diese vorhanden (auf manchen LINUX/UNIX-Servern), so wird diese vorrangig **VOR** der Umgebungsvariable **JAVA\_HOME** genutzt. Existiert die Umgebungsvariable `JRE_HOME`, so sollte diese Umgebungsvariable gelöscht oder zumindest geprüft werden, ob diese Umgebungsvariable auf eine `JAVA`-Installation zeigt, welche kompatibel mit der Version des Applikations-Servers Apache Tomcat ist.  
**Tipp:** Lässt sich aus organisatorischen Gründen die Umgebungsvariable **JRE\_HOME** nicht löschen und wird damit auch keine gültige `JAVA`-Version referenziert, so kann das Start-Script (`catalina.bat`|`catalina.sh`) den Bedürfnissen angepasst werden.

**Beispiel für Windows-Umgebungsvariablen:**



## **System-Voraussetzungen**

Um das Redaktions-System ordnungsgemäß betreiben zu können, muss Ihr System gewisse Mindestanforderungen erfüllen. Je nachdem, ob auf dem System der R-S Server oder ein R-S Client läuft, werden unterschiedliche Systemanforderungen gestellt.

*Systemvoraussetzungen für den R-S Client*

*Systemvoraussetzungen für den R-S Server*

## R-S Client

Um mit dem R-S Client arbeiten zu können, sind folgende Voraussetzungen notwendig:

### Technische Voraussetzungen

- Arbeitsplatz-PC mit aktueller Hardware-Ausstattung (mindestens 512 MB freiem Hauptspeicher).
- Bildschirm-Auflösung von mindestens 1280 x 720 Bildpunkten.
- Maximale Displayskalierung (Windows) von 125 %.
- WEB-Browser (getestete Versionen finden Sie im unteren Bereich dieses Artikels).
- im WEB-Browser müssen Cookies erlaubt sein.

### Logistische Voraussetzungen

- Sie müssen HTTP/HTTPS-Zugriff auf den R-S Server haben.  
(kontaktieren Sie bitte Ihren System-Administrator)
- Sie müssen über ein gültigen Nutzer mit einem entsprechenden Passwort verfügen.  
(kontaktieren Sie bitte Ihren System-Administrator)
- Sie müssen über die Berechtigung für den Zugriff auf mindestens einen Arbeitsbereich verfügen.  
(kontaktieren Sie bitte Ihren System-Administrator)

Derzeit getestete/freigegebene WEB-Browser (es wurden immer nur die Haupt-Versionen getestet):

- Mozilla Firefox ab 83.0
- Google Chrome ab 87.0
- Microsoft Edge ab 87.0

 Die Nutzung anderer als der hier aufgeführten WEB-Browser ist möglich. Bitte beachten Sie, dass keinerlei Support für nicht unterstützte WEB-Browser erbracht wird.

## R-S Server

Um den R-S Server ordnungsgemäß zu betreiben, sind folgende Voraussetzungen notwendig:

### Technische Voraussetzungen

- Unterstützte Betriebssysteme. Getestete und freigegebene Systeme sind:
  - Windows Server 2003 und 2008
  - Linux (SuseLinux SLES 10)
- Unterstützte Applikations-Server sind:
  - Apache Tomcat 8.X

### Logistische Voraussetzungen

- JAVA JDK 1.8 muss auf dem System verfügbar sein und die Umgebungsvariable JAVA\_HOME muss auf das JAVA JDK-Installationsverzeichnis zeigen.
- Eine der unter dem Abschnitt **Datenbanken** genannte SQL-Datenbank muss vom Kunden installiert/ eingerichtet sein.

 Das Redaktions-System verwendet in der Version 2.3 nicht mehr Java 6, sondern Java 8, weshalb das JDK 1.8 installiert sein muss.

 Die Nutzung anderer als der hier aufgeführten Betriebssysteme ist möglich, wenn diese Java in der entsprechenden Version unterstützen. Bitte beachten Sie, dass keinerlei Support für nicht unterstützte Betriebssysteme erbracht wird.

## Datenbanken

Das R-S benötigt zur Speicherung seiner Informationen eine Datenbank. Zum Einsatz kommen damit eine Reihe von relationalen SQL-Datenbanken wie folgt:

- **Derby (Apache)**  
+ unterstützte/getestete Versionen 10.6.X
- **MySQL (Oracle)**  
+ unterstützte/getestete Versionen 5.0.X, 5.6.X
- MSSQL-Server (Microsoft)  
+ unterstützte/getestete Versionen 2008

 Bei der Installation/Auslieferung des R-S ist die Derby-Datenbank vorinstalliert und kann für Test-/Evaluationszwecke genutzt werden. Diese Derby-Datenbank wird mit dem Applikations-Container (Apache Tomcat) zusammen gestartet und gestoppt und benötigt damit keinen administrativen Eingriff. Für den produktiven Betrieb wird die Derby-Datenbank jedoch nur eingeschränkt empfohlen und supportet.

Das R-S setzt einen der o.g. SQL-Datenbank-Server voraus, welcher in produktiven Kundenumgebungen in der Verantwortung durch den Kunden installiert/konfiguriert und betrieben (incl. Backup) wird. Ein Datenbank-Server sollte in produktiven Umgebungen des R-S immer dediziert, im besten Fall auf einem anderen Server-System als auf dem R-S Server betrieben werden.

## Datenbank - Derby

### **Datenbank-Server (Derby-DB)**

Nutzen Sie die Derby-SQL-Datenbank als Datenbank-Server (Auslieferungszustand des R-S), so finden Sie eine vorinstallierte Datenbank nach einer fehlerfreien Konfiguration vor, welche Sie sofort nutzen können. Haben Sie die Anforderung eine Derby-Datenbank neu zu erstellen, so können Sie dies entweder über SQL-Client-Software von Fremdanbietern erledigen oder mit dem Kommando-Zeilen-Client "ij" des Derby-Datenbank-Servers.

**i** Zum erfolgreichen Start des Kommando-Zeilen-Client "ij" des Derby-Datenbank-Servers wird eine Umgebungsvariable DERBY\_HOME benötigt. Diese Variable zeigt auf auf das Verzeichnis, in welchem der Kommando-Zeilen-Client "ij" installiert ist. Standardmäßig befindet sich dieses Verzeichnis unterhalb des Installation-Verzeichnisses des Applikations-Servers Apache Tomcat an folgender Stelle: "[R-S]\bin\rs\_helper\derby\_admin". Es gibt mehrere Möglichkeiten, Umgebungsvariablen im Betriebssystem oder für den Nutzer zu setzen, welcher den Applikations-Server startet. Sprechen Sie hier mit Ihrem Systemadministrator.

Starten Sie den Applikations-Server und danach den Kommando-Zeilen-Client "ij". Wenn Sie den Kommando-Zeilen-Prompt des "ij" angezeigt bekommen, führen Sie folgende Schritte aus:

#### **Schritt 1: Anlegen der Datenbank**

```
ij> connect 'jdbc:derby://localhost:XXXX/rs;create=true' user 'rs' password 'rs';
```

*Hinweis: Für 'localhost' können Sie die IP-Adresse oder Host-Namen des Derby-SQL-Datenbank-Servers angeben und für 'XXXX' geben Sie den TCP/IP-Port des Derby-SQL-Datenbank-Servers (Standard: 1527) an.*

#### **Schritt 2: Verbinden mit einer bestehenden Datenbank (Datenbank neu initialisieren)**

```
ij> connect 'jdbc:derby://localhost:XXXX/rs;create=false' user 'rs' password 'rs';
```

*Hinweis: Für 'localhost' können Sie die IP-Adresse oder Host-Namen des Derby-SQL-Datenbank-Servers angeben und für 'XXXX' geben Sie den TCP/IP-Port des Derby-SQL-Datenbank-Servers (Standard: 1527) an. Auch die Parameter 'user' / 'password' können bei einer bestehenden Derby-SQL-Datenbank geändert worden sein.*

#### **Schritt 3: Erzeugen der R-S Datenbank-Struktur**

```
ij> run './sql/rs.derby.complete.X_Y.sql';
```

Siehe: [http://db.apache.org/derby/manuals/#docs\\_10.0](http://db.apache.org/derby/manuals/#docs_10.0)

## Datenbank - MySQL

Für den produktiven Einsatz ist derby nur eingeschränkt geeignet. Deshalb wird empfohlen, hier MySQL (oder Alternativ: MSSQL) einzusetzen. Dieser Artikel beschreibt die Einrichtung der EOMS-Datenbank auf einem MySQL-Server.

### 1. Bezug von MySQL

Falls auf Ihrem System noch kein MySQL installiert ist, ist der 1. Schritt die Installation und Einrichtung eines MySQL-Servers. MySQL gibt es von verschiedenen Anbietern, die verbreitetste und empfohlene MySQL-Distribution ist MySQL von Oracle. Prinzipiell sind Sie bei der Wahl der Distribution aber frei.

Die MySQL Community Edition von Oracle können Sie [hier](#) beziehen. Achten Sie darauf, das richtige Paket für Ihr Betriebssystem und Ihre Systemarchitektur auszuwählen. Installieren Sie Server und Client, alle anderen Erweiterungen werden nicht benötigt.

 Merken Sie sich unbedingt den Benutzernamen und das Passwort, das Sie bei der Installation angeben.

### 2. Start des MySQL-Servers und des Kommandozeilen-Clients

Starten Sie den Datenbank-Server von MySQL, indem Sie im Verzeichnis {MYSQL\_SERVER}/bin eine Eingabeaufforderung öffnen und dort folgenden Befehl ausführen:

```
> mysqld
```

Starten Sie dann den Kommandozeilen-Client, indem Sie im Verzeichnis {MYSQL\_SERVER}/bin eine neue Eingabeaufforderung öffnen und dort folgenden Befehl ausführen:

```
> mysql --host <host> --port <port> -u <user> -p
> <password>
```

 Dabei ist <host> die IP-Adresse oder der Hostname des MySQL-Datenbank-Servers (z.B. 192.168.128.1, wenn der MySQL Server auf dem selben System läuft (Host = localhost) kann dieser Parameter weggelassen werden) und <port> der TCP/IP-Port des MySQL-Datenbank-Servers (Standard: 3306, wenn der Standardport verwendet wird kann dieser Parameter weggelassen werden), <user> der Benutzername, den Sie bei der Installation angegeben haben und <password> das dazugehörige Passwort. Achten Sie darauf, die Befehle getrennt einzugeben.

### 3. Anlegen der Datenbank

Führen Sie folgenden Befehl aus, um eine neue EOMS-Datenbank anzulegen:

```
mysql > CREATE DATABASE IF NOT EXISTS rs;
```

Um nun die neu angelegte / neu initialisierte Datenbank für das RS zu konfigurieren, führen Sie das mitgelieferte SQL-Skript aus:

```
mysql > use eoms;
mysql > source 'PATH_TO_SCRIPT/sql/rs.mysql.X_Y.sql';
```

*Wobei rs.mysql.X\_Y.sql das beim RS mitgelieferte Datenbankskript ist und X\_Y die RS-Versionsnummer. Wenn bei Ihnen kein Skript mitgeliefert wurde, können Sie das aktuelle Skript für mysql [hier](#) herunterladen.*

Damit das Core eine Verbindung zur Datenbank herstellen kann, müssen Sie in der rs.server.properties die Datenbankverbindung entsprechend konfigurieren. Mehr dazu finden Sie [hier](#).



Es wird dringend empfohlen, InnoDB als Datenbanktreiber zu verwenden. Mehr Informationen dazu finden Sie [hier](#).

## Systemsicherheit

Ist der Applikations-Server, die SQL-Datenbank und R-S korrekt installiert, sollten noch einige Aspekte der Systemsicherheit erörtert werden.

### **Apache Tomcat Applikations-Server Version 6.X:**

Nach der Installation des Apache Tomcat Server ist zu empfehlen, folgende Verzeichnisse/Dateien aus dem Installationsverzeichnis des Tomcat Applikations-Servers zu löschen:

- /webapps/examples
- /webapps/docs
- /webapps/manager
- /webapps/host-manager
- /conf/catalina/localhost (die Dateien 'host-manager.xml' und 'manager.xml')
- /work/Catalina/localhost (wird dynamisch neu vom Tomcat Applikations-Server erstellt)
- /conf/logging.properties

### **Apache Tomcat Applikations-Server Version 7.X:**

Nach der Installation des Apache Tomcat Server ist zu empfehlen, folgende Verzeichnisse/Dateien aus dem Installationsverzeichnis des Tomcat Applikations-Servers zu löschen:

- /webapps/ --> hier alle enthaltenen Verzeichnisse mit Unterverzeichnissen wie 'docs', 'examples', 'host-manager', 'manager', 'ROOT' ...
- /conf/logging.properties

 Sollten Sie die Manager-Applikation des Tomcat Applikations-Servers nutzen wollen, so sichern Sie diese nach den Vorgaben aus der Administrator-Dokumentation des Tomcat Applikations-Servers.

### **Apache Derby Datenbank**

Nach der Standard-Installation sind keine System-Nutzer auf der Derby-SQL-Datenbank aktiviert, somit kann sich jeder User ohne Angabe von Nutzer-ID und Passwort auf die Derby Datenbank verbinden. Ist dies nicht gewollt, so gibt es zwei Möglichkeiten, den Zugang zu erschweren oder zu autorisieren.

#### ***Bindung des Host-Prozesses der Derby Datenbank an die IP-Adresse LOCALHOST und einen anderen Port als im Standard***

Im Auslieferungszustand des R-S ist die Derby Datenbank an die lokale IP-Adresse (LOCALHOST) des Servers installiert, auf welchem das R-S installiert ist. Dies ist bezüglich der Systemsicherheit die effektivste Methode um den direkten Zugriff auf die Derby Datenbank von entfernten Computern zu unterbinden. Des weiteren ist zu empfehlen, den Standard-TCP/IP-Port der Datenbank auf einen anderen Wert als 1527 (Standard-Port der Derby Datenbank) zu setzen.

 Den im Auslieferungszustand genutzten TCP/IP-Port der Derby Datenbank entnehmen Sie aus Sicherheitsgründen der Administrator-Dokumentation.

### **Nutzung der Nutzer-Autorisierung der Derby Datenbank**

Bevor der Derby Datenbank-Server über authentifizierte Nutzer genutzt werden kann, müssen diese auf der Derby Datenbank angelegt werden. Nutzen Sie dazu den von Apache mitgelieferten Command-Line-Client 'ij' oder grafische SQL-Clients, welche in der Lage sind, sich auf die Derby SQL-Datenbank zu verbinden.

#### Befehl/Beispiel zur Anlage eines neuen System-Nutzers auf der Derby Datenbank

```
CALL SYSCS_UTIL.SYSCS_SET_DATABASE_PROPERTY('derby.user.XXX', 'YYYY')
```

Wobei 'XXX' der Name des Nutzers und 'YYY' das Passwort des Nutzers in Klartext ist.

#### Befehl/Beispiel zum Setzen der notwendigen Rechte für den User 'XXX'

```
CALL SYSCS_UTIL.SYSCS_SET_DATABASE_PROPERTY('derby.database.fullAccessUsers', 'XXX')
```

Mit der Datei 'derby.properties' im Datenbankverzeichnis der Derby Datenbank wird nun festgelegt, dass der Derby Datenbank-Server mit einer User-Authentifizierung arbeiten soll:

```
derby.connection.requireAuthentication=TRUE
derby.authentication.provider=BUILTIN
derby.user.XXX=YYYY
```

Wobei 'XXX' der Name des Nutzers und 'YYY' das Passwort des Nutzers in Klartext ist.

Es ist seitens des Betriebssystems darauf zu achten, dass die Konfigurations-Datei des Derby Servers nicht von unberechtigten Usern gelesen werden kann. Lesen Sie die Online-Dokumentation des Derby Datenbank-Servers, um weitere 'Authentication-Provider' wie LDAP oder Betriebssystem zu nutzen.

**i** Änderungen bezüglich der IP-Adresse/Port der Derby Datenbank, sowie des standardmäßigen System-Nutzers müssen in den entsprechend korrespondierenden Konfigurations-Dateien des R-S nachgepflegt werden.

**i** Grundsätzlich sind die Vorgaben der Produkt-Hersteller bezüglich Systemsicherheit oder die Sicherheitsanforderungen in den Kundenumgebungen umzusetzen.

## Verwendete Software

Nachfolgend wird aufgelistet, welche externe Software das R-S verwendet / benötigt:

- Apache Tomcat Version 8.X.
- Apache Derby Version 10.6.X (*optional MS SQL 8.X und MySQL 5.X*).
- Java JDK 1.8.X

## System-Berechtigungen

Die System-Berechtigungen werden durch den Administrator-Client des R-S gesetzt. Diese Berechtigungen sind systemweit im R-S gültig und haben keinen Bezug auf den Arbeitsbereich. Diese Art von Berechtigung ist möglich, um das R-S selbst zu administrieren und Systemaufgaben wahrzunehmen. Dazu zählen u.a.:

- Anlegen von Arbeitsbereichen
- Anlegen von Nutzern und zuordnen von Arbeitsbereichen
- Administration von Arbeitsbereich-Administratoren
- Zuweisung von Nutzer-Rechten
- etc...

## Arbeitsbereich-Administrator

Nachdem im R-S ein Arbeitsbereich angelegt worden ist, wird durch den System-Administrator diesem Arbeitsbereich ein Arbeitsbereich-Administrator zugeordnet (im R-S Admin Client, siehe [hier](#)). Der Arbeitsbereich-Administrator bekommt bei der Zuweisung zum Arbeitsbereich den Zugriffs-Pfad `/*` mit der Berechtigung **"Grant"** zugeordnet. Damit hat der Arbeitsbereich-Administrator alle verfügbaren Rechte im gesamten Arbeitsbereich. Außerdem hat er die Möglichkeit, anderen Nutzern und Nutzer-Gruppen Rechte zu vergeben.

 Sollte es Ihnen passieren, dass Sie sich bei der Arbeit in den Berechtigungen selbst ausschließen, so können Sie sich vom System-Administrator über die **Administrator-Client Konsole** jederzeit diese Berechtigung wieder erteilen lassen.

## Nutzer, Nutzer-Gruppen und Rechte

Als Arbeitsbereichs-Administrator haben Sie die Befugnis, Nutzer-Gruppen zu erstellen, Nutzer zu verwalten und ihnen Rechte zu vergeben. Wechseln Sie dazu in den Navigationspunkt  Nutzerverwaltung Abb. A (1).

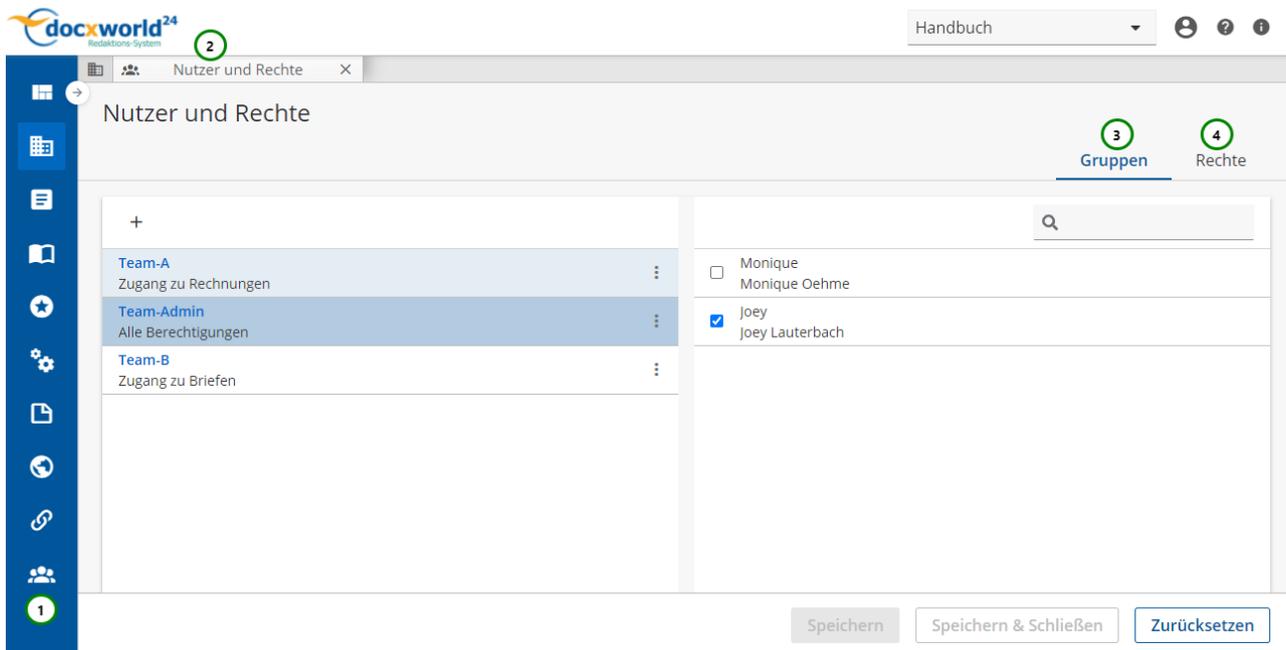


Abbildung A

Die Nutzerverwaltung ist als ein Objekt zu betrachten. Nachdem Sie den Menüpunkt geöffnet haben, öffnet sich ein neuer Tab Abb. A (2).

Das Objekt teilt sich in 2 Reiter auf :

- Gruppen Abb. A (3) und
- Rechte (4).



- Sie können alle Änderungen jederzeit speichern oder verwerfen.
- Sie können diese Ansicht nur aufrufen, wenn Sie Administrator des Arbeitsbereichs sind. Dieses Recht wird im R-S Admin Client **vergeben**. Wenden Sie sich dazu gegebenenfalls an Ihren Systemadministrator.
- Sie können im R-S keine Nutzer erstellen. Um Nutzer zu erstellen wird der Admin-Client verwendet. Im R-S können Sie Nutzer nur für den aktuellen Arbeitsbereich verwalten und nur Rechte für diesen Arbeitsbereich vergeben. Weitere Informationen finden Sie **hier** und **hier**.

## Nutzer verwalten

Als Administrator haben Sie im Admin-Client die Möglichkeit, Nutzer zu Verwalten und zu Bearbeiten. Öffnen Sie dazu den Admin-Client (*Die Benutzung des Admin-Client wird [hier](#) beschrieben*).

 Momentan wird der Admin-Client noch nicht zusammen mit dem R-S an Endkunden ausgeliefert.

Folgende Befehle stehen Ihnen zur Nutzerverwaltung zur Verfügung:

### **Einen Nutzer anlegen**

Um einen neuen Nutzer im R-S anzulegen, benutzen Sie folgenden Befehl:

```
adminclient -createUser <user-email/ID> <password> <firstname> <lastname>
```

wobei **user-email/ID** der Nutzername ist, **password** das Passwort, **firstname** der Vorname und **lastname** der Nachname des Nutzers.

Bsp: adminclient -createUser MaxMuster qwertz123 Max Mustermann

Ein neu angelegter Nutzer ist noch keinem Arbeitsbereich zugewiesen. Sie müssen deshalb nach dem Anlegen dem Nutzers mit -assignWorkspace einen Arbeitsbereich zuweisen. Beachten Sie dazu den Artikel über [Arbeitsbereiche](#).

 Nachdem ein Nutzer angelegt wurde, wird dringend empfohlen, dass der Nutzer das Passwort zu einem nur ihm bekannten Passwort ändert!

### **Passwort eines Nutzers ändern**

Ein Nutzer kann sein Passwort direkt in seinem Profil ändern. Das Passwort kann aber auch per Admin-Client geändert werden:

```
adminclient -changePassword <user-email/ID> <old_password> <new_password>
```

wobei **user-email/ID** der Nutzername ist, **old\_password** das alte Passwort und **new\_password** das neue Passwort.

Bsp: adminclient -changePassword MaxMuster qwertz123 asdfgh456

Das Passwort kann auch zurückgesetzt werden, wobei das alte Passwort nicht erforderlich ist:

```
adminclient -resetPassword <user-email/ID> <new_password>
```

wobei **user-email/ID** der Nutzername ist und **new\_password** das neue Passwort.

Bsp: adminclient -resetPassword MaxMuster asdfgh456

### **Nutzer auflisten**

Sie haben 3 Möglichkeiten, eine Liste bestimmter Nutzer anzeigen zu lassen:

1. Alle Nutzer im R-S anzeigen lassen:

*Befehl erwartet keine Argumente.*

```
adminclient -listUsers
```

*Befehl erwartet keine Argumente.*

2. Aktive (Momentan angemeldete Nutzer) anzeigen lassen:

```
adminclient -listActiveUsers
```

*Befehl erwartet keine Argumente.*

3. Nutzer eines bestimmten Arbeitsbereichs anzeigen lassen:

```
adminclient -listWorkspaceUsers <workspace_name>
```

wobei **workspace\_name** der Name des Arbeitsbereichs ist.

Bsp: adminclient -listWorkspaceUsers MeinWorkspace

## Rechte

Im Reiter Rechte Abb. A (1) legen Sie die Berechtigungen für Nutzer und Gruppen fest.

The screenshot displays the 'Nutzer und Rechte' (Users and Rights) management interface. At the top, there is a search bar for 'Alle Nutzer und Gruppen' and a list of users: Timo Timo Stahl, Monique Monique Oehme, and Joey Joey Lauterbach. A context menu is open over the third user entry, showing options: 'Eigenschaften', 'Löschen', 'Aufwärts', and 'Abwärts'. The interface also includes a 'Speichern' button, a 'Speichern & Schließen' button, and a 'Zurücksetzen' button.

Abbildung A

Mit Rechten legen Sie fest, zu welchen Aktionen Nutzer autorisiert sind und welche Objekte sie sehen dürfen. Jeder Nutzer hat ihm zugewiesene Rechte. Sie können Nutzern Rechte entweder geben (*grant*) oder verwehren (*deny*). Für folgende allgemeine Aktionen können Sie Rechte erteilen:

Recht	Beschreibung
create	Bestimmt, ob Nutzer Objekte erstellen dürfen.
update	Bestimmt v.a., ob Nutzer Objekte ändern dürfen.
delete	Bestimmt, ob Nutzer Objekte löschen dürfen.
read	Bestimmt, ob Nutzer Objekte sehen und einsehen dürfen.
test	Bestimmt u.a., ob Nutzer Objekte in den TEST -Status setzen dürfen.
production	Bestimmt u.a., ob Nutzer Objekte in den PRODUCTION -Status setzen dürfen.

Neben diesen allgemeinen Aktionen können Sie mit Hilfe eines Zugriffs-Pfads Rechte auch auf einzelne Objekte anwenden und detailliert und präzise angeben, welche Rechte ein Nutzer genau hat. Lesen Sie dazu die folgenden Abschnitte.

### Rechte erstellen und ändern

Um Rechte zu erstellen, klicken Sie bitte auf **+** [Abb. A \(2\)](#).

Um bestehende Rechte zu ändern, wählen Sie das **:** Kontextmenü (3) des Rechts aus, das Sie ändern möchten und klicken Sie anschließend auf **≡** Eigenschaften (4). In beiden Fällen gelangen Sie zu folgender Seite:

The image shows two side-by-side screenshots of a software interface for creating and editing access paths. Both windows have a title bar with a close button (X) and a title: 'Zugriffs-Pfad erstellen' (left) and 'Zugriffs-Pfad ändern' (right). Each window contains the following fields:

- Pfad \***: A text input field containing the value `/*`.
- Typ \***: A dropdown menu with the value `GRANT`.
- Alle Aktionen \***: A checkbox that is currently unchecked.
- Aktionen \***: A list of actions with checkboxes:
  - READ
  - CREATE
  - UPDATE
  - DELETE
  - TEST
  - PRODUCTION
 In the 'Zugriffs-Pfad ändern' window, all these checkboxes are checked.

At the bottom of each window are two buttons: 'Abbrechen' and 'Ok'. In the left window, there are four green circles with numbers 1, 2, 3, and 4. Circle 1 points to the 'Pfad' field, circle 2 to the 'Typ' dropdown, circle 3 to the 'Alle Aktionen' checkbox, and circle 4 to the 'Aktionen' list.

Abbildung B

**Sie können:**

- i • Einen **Zugriffs-Pfad** zuteilen *Abb. B (1)* oder ihn ändern.
- Einen **Zugriffs-Typ** festlegen (2) oder ändern.
- **Dem Nutzer oder der Gruppe alle Rechte geben** (3) ..
- **einzelne Aktionen zuordnen**, die Sie selbst auswählen können (4).

Mit Hilfe des Zugriffs-Pfads können Sie genau bestimmen, welche Aktionen der Nutzer für welche R-S Objekte ausführen darf. In das Eingabefeld geben Sie den Pfad immer in der gleichen Weise ein. Der Zugriffs-Pfad spezifiziert, auf welche Objekte sich die Regel bezieht. Eine komplette Liste finden Sie [hier](#). Wenn Sie die Regel auf alle Objekte oder auf alle Aktionen eines Objekts anwenden wollen, nutzen Sie " \* ". Der Operator \* ist für alle möglichen Angaben wahr (*siehe dazu auch Regel*) . Sie können die Objekt-Bezeichnungen auch einschränken, in dem Sie z.B. "/tx/Fly\*" eingeben. Jetzt werden nur alle Objekt-Bezeichnungen angezeigt, die mit "Fly" beginnen und beliebig enden.

Es gibt zwei Arten von Zugriffs-Typen: `grant` und `deny` . Die Standardeinstellung ist `grant` . Wenn Sie für die in einem Zugriffs-Pfad definierte Regel den Zugriffs-Typ `grant` angeben, bedeutet das, dass der Nutzer für die in der Regel definierten Aktionen autorisiert ist (Rechte werden gegeben). Wenn Sie für die in einem Zugriffs-Pfad definierte Regel den Zugriffs-Typ `deny` angeben, bedeutet das, dass der Nutzer für die in der Regel definierten Aktionen nicht autorisiert ist (Rechte werden verwehrt).

Bestätigen Sie die Eingabe mit Ok, um die Regel anzulegen bzw. zu ändern.

**Nutzer und Gruppen einem Recht zuordnen**

Damit dieses Recht auf Benutzer und Gruppen angewandt werden kann, müssen Sie die entsprechenden Nutzer und Gruppen diesem Recht zuordnen.

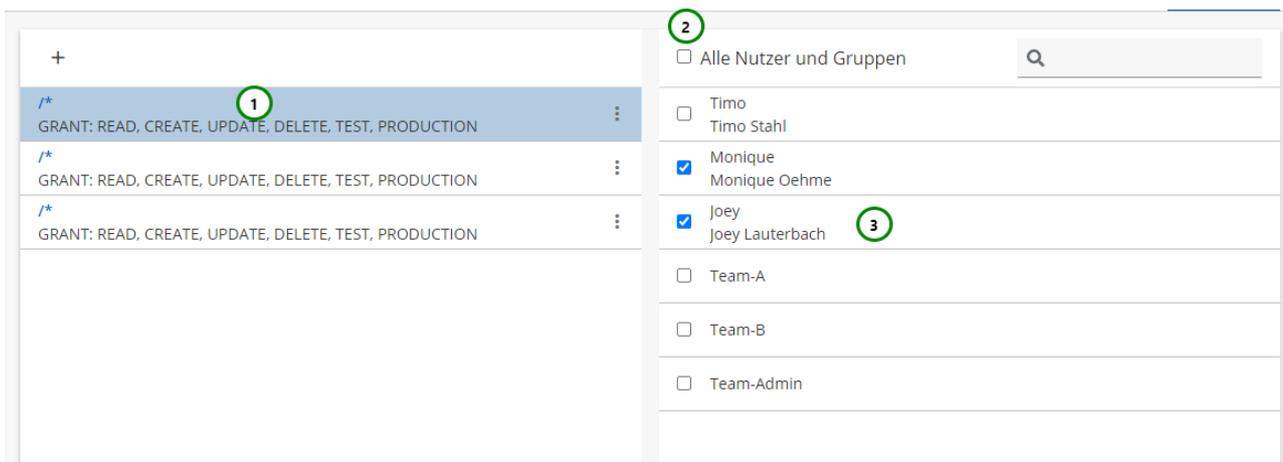


Abbildung C

Wählen Sie zunächst die eben erstelle Berechtigung aus *Abb. C (1)*. Sie sehen nun auf der Rechten Seite alle Benutzer und Gruppen, die in diesem Arbeitsbereich existieren. Setzen Sie einen Haken bei den Benutzern, auf denen dieses Recht angewandt werden soll.

### **Rechte verschieben**

Klicken Sie auf das  Kontextmenü *Abb. A (3)* und anschließend auf  Aufwärts oder  Abwärts (6), um das Recht zu verschieben. Sie bestimmen damit deren Priorität.

Die Priorität der in der Berechtigungsliste enthaltenen Rechte nimmt von oben nach unten zu. Das bedeutet dass ein Recht, das in der Berechtigungsliste unter einem anderen Recht steht immer Vorrang hat und das obere Recht überschreibt. Es ist deshalb auch möglich, die Interpretations- Reihenfolge der enthaltenen Rechte zu beeinflussen, indem Sie Rechte nach oben oder unten verschieben.

Beispiel: (in beiden Rechten sind die gleichen Benutzer hinterlegt)

/txr/*		
DENY: ALL		
/txr/*		
GRANT: ALL		

Durch Zeile (1) verbieten Sie den Zugang zu den Transaktions-Formular-Versionen. Dies wird allerdings durch die darunterliegende Zeile (2) wieder aufgehoben.

### **Rechte löschen**

Klicken Sie auf das  Kontextmenü *Abb. A (3)* und anschließend auf  löschen (5), um das Recht zu löschen.

 Zum Speichern der Änderungen, klicken Sie auf Speichern!

## Berechtigungen

### Berechtigungen im Arbeitsbereich

Im Arbeitsbereich können Nutzer Berechtigungen über Nutzergruppen zugeordnet bekommen oder der Nutzer erhält die Berechtigungen direkt als R-S Nutzer in diesem Arbeitsbereich zugeordnet. Auf folgende Objekte des R-S können Berechtigungen vergeben werden:

R-S Objekt	Objekt-URL	Recht/Aktion 1	Recht/Aktion 2	Recht/Aktion 3	Recht/Aktion 4	Recht/Aktion 5	Recht/Aktion 6
Transaktions-Formular	/tx/*	<b>create</b>	<b>read</b>	<b>update</b>			
Transaktions-Formular-Version	/txr/*	<b>create</b>	<b>read</b>	<b>update</b> Workflow-Status: - set to TEST - set to REJECTED - set to EDIT		<b>test</b> Workflow-Status: - set to ACCEPTED - set to REJECTED	<b>production</b> Workflow-Status: - set to DEPRECATED - set to ACCEPTED
Schema	/s/*	<b>create</b>	<b>read</b>	<b>update</b>			
Schema-Version	/sr/*	<b>create</b>	<b>read</b>	<b>update</b> Workflow-Status: - set to TEST - set to REJECTED - set to EDIT			<b>production</b> Workflow-Status: - set to DEPRECATED - set to ACCEPTED
Produktions-Paket	/pb/*		<b>read</b>	<b>update</b>		<b>test</b> Workflow-Status: - set to ACCEPTED - set to REJECTED - set to TEST	<b>production</b> Workflow-Status: - set to DEPRECATED - set to ACCEPTED - set to ERROR - set to PRODUCTION
Bibliothek	/l/*	<b>create</b>	<b>read</b>	<b>update</b>			

R-S Objekt	Objekt-URL	Recht/Aktion 1	Recht/Aktion 2	Recht/Aktion 3	Recht/Aktion 4	Recht/Aktion 5	Recht/Aktion 6
Bibliotheks-Version	/lr/*	<b>create</b>	<b>read</b>	<b>update</b> Workflow-Status: - set to TEST - set to REJECTED - set to EDIT		<b>test</b> Workflow-Status: - set to ACCEPTED - set to REJECTED	<b>production</b> Workflow-Status: - set to DEPRECATED - set to ACCEPTED
Docxworld-Link	/dw/*	<b>create</b>	<b>read</b>	<b>update</b>	<b>delete</b>		
Produktions-Paket Verknüpfung	/pblink/*	<b>create</b>	<b>read</b>	<b>update</b>	<b>delete</b>		
Material	/m/*	<b>create</b>	<b>read</b>	<b>update</b>	<b>delete</b>		
fonts	/f/*		<b>read</b>				
Arbeitsbereich (z.B. Nutzerverwaltung)	/ws		<b>read</b>	<b>update</b>			
globales Administrator-Recht (umfasst alle R-S Objekte)	/*	<b>create</b>	<b>read</b>	<b>update</b>	<b>delete</b>	<b>test</b>	<b>production</b>



- Bei dem **R-S Objekt** Material wird nicht zwischen physikalisch und logisch unterschieden. Es wird also nur ein "m" geschrieben.
- Beim Arbeitsbereich wird nur "/ws" geschrieben, da in einem Arbeitsbereich nicht mehrere Nutzerverwaltungen existieren und diese damit nicht explizit angegeben werden müssen.
- Bereichs-Administratoren besitzen alle Rechte.

## Gruppen

Im Reiter Gruppen können Sie Gruppen anlegen, um diesen später Rechte zuweisen zu können. Das hat den Vorteil, dass Sie nicht für jeden Nutzer einzeln die Rechte vergeben müssen, sondern einer ganzen Gruppe an Nutzern die gleichen Rechte vergeben können.

Bei den Gruppen stehen Ihnen folgende Aktionen zur Verfügung um Nutzer-Gruppen zu erstellen **+** Abb. A (1), zu bearbeiten  (3) oder zu löschen  (4):

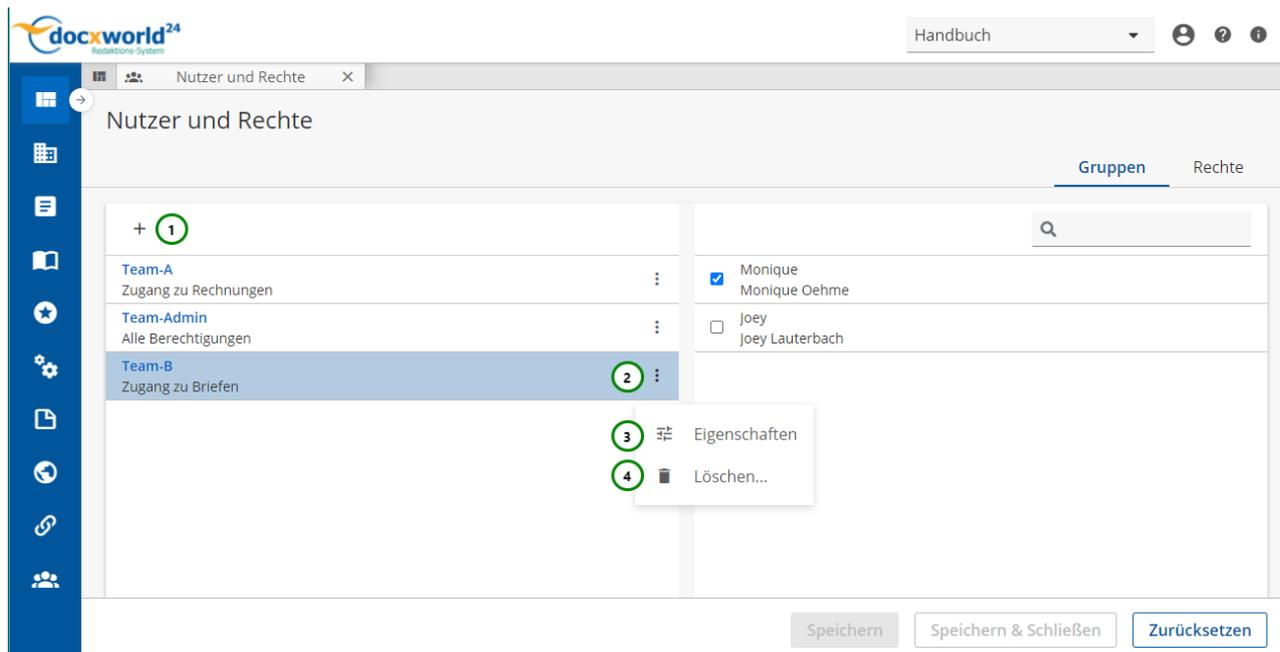


Abbildung A

Sie können mehrere Nutzer in einer Nutzer-Gruppe zusammenfassen und dieser Nutzer-Gruppe dann im Zugriffs-Pfad **ansprechen**. Die dort definierten Rechte gelten dann für alle Nutzer in dieser Gruppe.

### Gruppe erstellen oder ändern

Um eine Nutzer-Gruppe zu erstellen, klicken Sie bitte auf **+** Abb. A (1).

Um eine Nutzer-Gruppe zu ändern, wählen Sie das  Kontextmenü (2) der Gruppe aus, welche Sie ändern möchten und klicken Sie anschließend auf  Eigenschaften (3).

In beiden Fällen gelangen Sie zu folgender Ansicht:

Abbildung B

Tragen Sie nun den Namen und die Beschreibung (optional) der Gruppe ein und bestätigen Sie mit Ok. Die Gruppe ist nun angelegt bzw. geändert.

### **Nutzer zuordnen**

Wählen Sie die Gruppe, der Sie Nutzer zuordnen möchten aus [Abb. C \(1\)](#).

Abbildung C

Nun sehen Sie auf der Rechten Seite [Abb. C \(2\)](#) die Nutzer, die diesen Arbeitsbereich zugeordnet sind. Geben Sie nun jedem Nutzer, der dieser Gruppe zugeordnet werden soll einen Haken.

Ein Benutzer kann auch gleichzeitig mehreren Benutzergruppen zugeordnet werden

### **Gruppe löschen**

Wählen Sie zum Löschen einer Gruppe, das Kontextmenü [Abb. A \(2\)](#) der Gruppe aus, die Sie ändern möchten und klicken Sie anschließend auf Löschen (4).

Um Alle Änderungen zu speichern, klicken Sie auf Speichern.

## System-Nutzer

Wird ein R-S neu installiert, so werden folgende Nutzer vom System automatisch angelegt:

- admin
- production

### **Nutzer 'admin'**

Der Nutzer 'admin' ist System-Nutzer und der Administrator des R-S und keinem Arbeitsbereich zugeordnet. Benötigt wird dieser Nutzer für die Arbeiten mit dem **Administrator-Client**. Der Nutzer 'admin' besitzt nach der Installation des R-S das Passwort 'admin'.

### **Nutzer 'production'**

Der Nutzer 'production' ist System-Nutzer und keinem Arbeitsbereich zugeordnet. Benötigt wird dieser Nutzer für den externen Abruf von Produktions-Paketen. Der Nutzer 'production' besitzt nach der Installation des R-S das Passwort 'production'.



Es wird dringend empfohlen nach der Installation des R-S das Passwort der Nutzer 'admin' und 'production' zu ändern und diese Änderungen in den Konfigurations-Dateien des R-S nachzuarbeiten.

## Systembetrieb

Dieses Kapitel beinhaltet weiterführende Informationen zum Betrieb des R-S und dazugehöriger Komponenten. Dazu gehören Konfiguration, Administration und Sicherung des R-S. Folgende Themen werden ausgeführt:

- Die Erstellung von **Backups**.
- **Binär-Pakete**, die die zur Dokumentenverarbeitung nötigen Programme enthalten.
- **HTTP-Status-Codes** durch Client- oder Server-Fehler im WEB-Browser.
- Wie Sie das R-S mit Hilfe eines zuvor gemachten **Backup's wiederherstellen**.
- Die **R-S Server Konfigurationsdatei**, die wichtige R-S Einstellungen enthält.

## Backup

Ein wirklich konsistentes/synchrones und vollständiges Backup für das R-S erhalten Sie indem Sie wie folgt vorgehen:

1. Beenden Sie das Redaktions-System und sichern Sie das komplette Verzeichnis des R-S.
2. Beenden Sie das Datenbank-System und sichern Sie das komplette Verzeichnis des Datenbank-Systems.
3. Starten Sie das Datenbank-System neu.
4. Starten Sie das R-S neu.

 Bei allen anderen nachfolgend beschriebenen Backup-Möglichkeiten garantiert profiforms nicht für die Synchronität/Wiederherstellbarkeit von Datenbank und File-Storage.

## Binär-Pakete

Binär-Pakete sind die ausführbaren Programme zur Aufbereitung der Druckvorschau im R-S oder zur Aufbereitung der Dokumente in den produktiven Umgebungen wie Docxworld oder OMS/EOMS. Welche Vorteile werden durch Binär-Pakete u.a. erlangt:

- Binär-Pakete werden durch geschulte Systemadministratoren des R-S administriert, d.h. nicht die Endanwender müssen sich um die Verfügbarkeit/Korrektheit der Versionen der Output-Management-Produkte kümmern.
- Die Import-Pakete für Transaktions-Formulare müssen nicht mehr die Output-Management-Produkte (Binär-Dateien) enthalten.
- Der Speicher des R-S wird wesentlich entlastet, da die Import-Pakete sich stark verkleinern.
- Es wird sichergestellt, dass die Output-Management-Produkte für die Plattformen verfügbar sind, auf welchen Docxworld und OMS/EOMS die Dokumente in der Produktion aufbereiten.

In den Import-Paketen für die Transaktions-Formulare wird lediglich für die unterschiedlichen Plattformen der Name und die gewünschte Version des Output-Management-Produktes (ReportWriter, Logistics ...) angegeben. Auf der Basis dieser Angaben stellt das R-S die für die Druckvorschau oder die Dokument-Aufbereitung notwendigen Binär-Pakete zur Verfügung. Ein Binär-Paket wird pro gewünschter Version des Output-Management-Produktes erstellt. Der Abruf eines Binär-Paketes erfolgt über den Namen des Output-Management-Produktes und über die angeforderte Version.



Es muss pro Version des Output-Management-Produktes (z.B. ReportWriter oder Logistics) und pro Plattform (z.B. Windows, Linux) ein eigenes Binär-Paket erstellt werden!

### Weiterführende Informationen zu Binär-Paketen:

- [Aufbau des ZIP-Binär-Paketes](#)
- [Aufbau der Binär-Paket Beschreibungs-Datei](#)

## Aufbau des ZIP-Binär-Paketes

Generell sei vorausgeschickt, dass das ZIP-Binär-Paket zwingend nur das Unterverzeichnis **etc** zur Strukturierung der Dateien benötigt, da darin die Paket-Beschreibungsdatei enthalten ist. Nutzer der Output-Management-Suite (**Spooler**) und des DeveloperClient haben sich in der Praxis jedoch so organisiert, dass die notwendigen Programme (Binäre-Dateien) thematisch in Unterverzeichnissen abgelegt werden. Im Folgenden sehen Sie einen typischen Verzeichnis-Aufbau eines ZIP-Binär-Paketes als Beispiel:

Verzeichnisname	Inhalt des Verzeichnisses
etc	Das Verzeichnis enthält die Paket-Beschreibungs-Datei <b>rs_binary_configuration.xml</b> des Binär-Paketes, welche für den erfolgreichen Import eines Binär-Paketes in das R-S unbedingt notwendig ist.  ⚠️ Detaillierte Informationen zum Aufbau der Paket-Beschreibungs-Datei finden Sie hier: <a href="#">Aufbau der Binär-Paket Beschreibungs-Datei</a>
bin	Das Verzeichnis enthält alle binären Dateien (z.B. reportw.exe oder logistics.exe) oder Skripte, welche für die Druckvorschau im R-S und zur Aufbereitung der Dokumente in docxworld- oder OMS/EOMS-Umgebungen genutzt werden.



### Bitte beachten Sie:

- dass die Verzeichnisse mit dem Namen rs und rw im Standard des R-S verwendet werden und als Verzeichnis-Namen im Import-Paket nicht benutzt werden dürfen.
- dass Dateien und Verzeichnisse innerhalb des ZIP-Import-Containers keinerlei Umlaute und Sonderzeichen enthalten dürfen (Optimal: a-z,A-Z,-).

### Achten Sie bei der Erstellung der ZIP-Datei des Binär-Paketes auf folgende Hinweise:

- vermeiden Sie bei der Vergabe der Verzeichnisnamen Leerzeichen
- die genannten Verzeichnisnamen beim ZIP-Vorgang nicht wiederum in einem Verzeichnis enthalten sind
- das ZIP-Programm die ZIP-Spezifikation erfüllt (siehe WIKIPEDIA: <http://www.iana.org/assignments/media-types/application/zip>)

## Aufbau der Binär-Paket Beschreibungs-Datei

Wird ein neues Binär-Paket (**ZIP-Binär-Paket**) über **Administrator-Client (Console)** in das R-S importiert, so wird im Unterverzeichnis **etc** nach einer XML-Datei mit dem Namen **rs\_binary\_configuration.xml** gesucht. Diese XML-Datei beschreibt den Inhalt und den Aufbau des ZIP-Binär-Paketes genauer und ermöglicht eine inhaltliche Prüfung.

In der Paket-Beschreibungs-Datei sind u.a. folgende Informationen abgelegt:

- Name des Output-Management-Produktes/Scriptes (nachfolgend Programm genannt), als Referenz aus dem Transaktions-Formulars.
- Genaue Programm-Version.
- Die Plattform, für die das Programm gültig ist.
- Eine Information, ob das Binär-Paket mit dem Produktions-Paket bei der Verarbeitung gemischt (MERGED) werden kann oder ob mit einer Referenz des Binär-Paketes (SHARED) gearbeitet werden muss.
- Eine Kurz-Beschreibung zum Programm.
- Die plattformspezifische Aufrufzeile des Programmes.
- Die Kommandozeilen-Parameter beim Aufruf des Programmes.

Die Paket-Beschreibungs-Datei **rs\_binary\_configuration.xml** wird mit einem geeigneten XML-Editor durch den Administrator des R-S manuell erstellt.

 Ab der Version 1.7.1000 wird nur noch der Import von Binär-Paketen mit der API-Version 0.0.02 unterstützt.

### Die in der Paket-Beschreibungs-Datei verwendeten reservierten Variablen bedeuten:

- `${inputdata}` - Verzeichnis und Dateiname der Dokument-Datei, welche aufbereitet werden soll.
- `${bin}` - Verzeichnis, in welchem das Binär-Paket bei der Dokument-Aufbereitung abgelegt wird.
- `${prod}` - Verzeichnis, in welchem das Produktions-Paket bei der Dokument-Aufbereitung abgelegt wird.
- `${home}` - temporäres Verzeichnis, in welchem die Ergebnisse der Dokument-Aufbereitung abgelegt werden.
- `${params}` - ergänzende Kommandozeilen-Parameter aus dem Transaktions-Formular ('rs\_package\_configuration.xml' --> <cmdLine> **Aufbau der Paket-Beschreibungs-Datei**).

Beispiel einer Paket-Beschreibungs-Datei in der API-Version 0.0.02 (ab Version 1.7.1000) für das Output-Management-Programm ReportWriter:

```
<?pte APIVersion="0.0.02"?>
<binary>
  <name>ReportWriter</name>
  <version>5.7.1000</version>
  <platform>Windows</platform>
  <supported-runtime-environments>
    <environment>MERGED</environment>
    <environment>SHARED</environment>
  </supported-runtime-environments>
  <description>ReportWriter 5.7.1000 fuer Windows</description>
  <cmd>${bin}/bin/reportw</cmd>
  <args>${inputdata} ${prod}/etc/reportw.tci -awp ${home} -aip ${prod}/etc -afp $
{prod}/forms -afp ${prod}/rw -alp ${prod}/logos -all process.log -vol -rsp XML $
{params} -sno XXXXXX</args>
</binary>
```

Beispiel einer Paket-Beschreibungs-Datei in der API-Version 0.0.01 (bis Version 1.7.1000) für das Output-Management-Programm ReportWriter:

```
<?pte APIVersion="0.0.01"?>
<binary>
  <name>ReportWriter</name>
  <version>5.7.1000</version>
  <platform>Windows</platform>
  <description>ReportWriter 5.7.1000 fuer Windows</description>
  <cmd>${home}/bin/reportw</cmd>
  <args>${inputdata} ${home}/etc/reportw.tci -awp ${home} -aip ${home}/etc -afp $
{home}/forms -afp ${home}/rw -alp ${home}/logos -all process.log -vol -rsp XML $
{params} -sno XXXXXX</args>
</binary>
```

 Bei fehlerhafter XML-Syntax oder fehlerhaft definierten Objekt-Referenzen bricht der Import des Paketes im R-S mit Fehlermeldungen ab.

Download einer Muster-Paket-Beschreibungs-Datei: [rs\\_binary\\_configuration.xml](#)

Die Verwaltung der Binär-Pakete wie der Import oder die Auflistung verfügbarer Binär-Pakete, findet über den **Administrator-Client (Console)** statt.

## Programme in Binär-Paketen

Generell können beliebige Programme/Scripts in Binär-Paketen eingebunden werden, welche dann durch das R-S bei der Druckvorschau und durch docxworld oder weitere Systeme zur Generierung von Druckdaten genutzt werden.

In diesem Teil der Online-Hilfe wird auf ausgewählte Produkte der Firma profiforms verwiesen, welche üblicherweise in Binär-Paketen genutzt werden.

 Dieser Teil der Online-Hilfe ersetzt NICHT die Dokumentation und Hilfesystem der entsprechenden Produkte sondern ergänzt Informationen, welche im Zusammenhang mit der Nutzung als Binär-Pakete des R-S notwendig sind.  
Diese Informationen erheben keinen Anspruch auf Vollständigkeit !

## ReportWriter

Der ReportWriter ist eine Software aus dem Hause profiforms und fügt Rohdaten, vordefinierte Formular-Layouts und fertige PDF's ganz flexibel zu kompletten Dokumenten in Ihrem Corporate Design zusammen.

Weitere Informationen und Datenblätter sowie Informationen zur Arbeitsweise finden Sie hier: [profiforms](#).

### Return-Codes des ReportWriters

Durch fehlerhafte Konfigurationen im Transaktions-Formular, dem Binär-Paket oder auch im R-S kann es zu Fehlern bei der Ausführung des ReportWriters kommen. Dieser Fehlercode wird als Ergebnis der Druckvorschau im R-S angezeigt und die Ursache des Fehlers wird zur Analyse in den Log-Files zu diesem Prozess angeboten. Um die Nutzerfreundlichkeit zu verbessern, finden Sie hier einen Auszug aus der Liste der Return-Codes des ReportWriters mit einer kurzen Fehlerbeschreibung:

Return-Code	Beschreibung
0	Die Arbeit des ReportWriters wurde ohne Fehler beendet.
3 4	<p>Eine vom ReportWriter benötigte Konfigurations-/Datendatei kann nicht geöffnet werden.</p> <p><u>Ursache:</u> Die entsprechende Konfigurations-/Datendatei ist nicht vorhanden, der ReportWriter besitzt nicht die notwendigen Zugriffsrechte oder die benötigte Datei ist nicht im erwarteten Format (INI, TCI ...).</p> <p><u>Lösung:</u> Prüfen Sie ob die entsprechende Datei verfügbar, zugreifbar ist und im richtigen Format vorliegt.</p>
24	<p>Der für die Druckvorschau verwendete Datenstrom kann durch den ReportWriter nicht erkannt werden.</p> <p><u>Ursache:</u> In der Konfigurationsdatei des ReportWriters ist die Dokument-Erkennung fehlerhaft oder der verwendete Datenstrom ist nicht im erwarteten Format/Inhalt für das verwendete Transaktions-Formular.</p> <p><u>Lösung:</u> Prüfen Sie den Datenstrom für die Druckvorschau und ob dieser in dem erwarteten/definierten Format vorliegt, oder überarbeiten Sie die Dokument-Erkennung in der Konfigurationsdatei des ReportWriter.</p>
134 1203	<p>Bei der Abarbeitung des Formulars ist es zu einem rekursivem Aufruf von Objekten (Loop) gekommen, die Verarbeitung wurde abgebrochen.</p> <p><u>Lösung:</u> Prüfen Sie die Ressourcen im R-S bezüglich rekursivem Aufruf. Hinweis: Das R-S erkennt selbständig nur einstufige Rekursionen.</p>

Return-Code	Beschreibung
690	<p>Fehler in der Serien-Nummer.</p> <p><u>Ursache:</u> Die Seriennummer ist zeitlich begrenzt und abgelaufen oder nicht für die Plattform des R-S gültig.</p> <p><u>Lösung:</u> Erzeugen Sie ein Binär-Paket mit einer gültigen Produkt-Seriennummer.</p>

## Funktion - GetTranspromoResource

### Funktionsbeschreibung

Die GetTranspromoResource-Funktion des ReportWriters sucht in seiner Laufzeitumgebung nach Definitionen aus dem R-S und fügt den Inhalt der in der Funktion angegebenen Ressource aus dem R-S an der Stelle ein an der die Funktion GetTranspromoResource aufgerufen worden ist.

Bei der Selektion der angegebenen Ressource aus dem R-S werden alle dynamischen Inhalte der Ressource aufgelöst. Ist bei dem Funktionsaufruf zum Beispiel der Name einer Regel angegeben, so wird die Regel ausgeführt und die entsprechende Ergebniszeile an den ReportWriter übergeben. Folgende Ressource-Typen sind für diese Funktion vorgesehen:

- Variablen
- Regeln
- RichTexte

### Syntax

GetTranspromoResource(PN1)

### Parameter (1)

STRING PN1: Name der Resource im R-S

### Rückgabewert

Inhalt der Resource aus dem R-S (Inhalt einer Variable, Regel, ...)

### Beispiel (selbstumbrechender, dynamischer Text)

```
Position LETTER_TEXT {
  Calc {
    letter_text_field=GetTranspromoResource("R_Letter")
  }
  Subform {
    LineWrap 1
    WrapTable LETTER
    name letter_text
  }
}
```

### **Selbstumbrechende Felder mittels RichTexten**

Mit GetTranspromoResource-Funktion können im ReportWriter u.a. Variablen definiert oder durch dynamische Positionen sogar komplette selbst umbrechende Texte (RichText) genutzt werden.

 Damit Fonhöhen, die Höhe von Leerzeilen von RichTexten aus dem R-S berücksichtigt werden, müssen die korrespondierenden Teilformulare im Adobe XDP-Formular mit der Layout-Eigenschaft 'Automatisch anpassen' und das korrespondierende Textfeld mit der Layout-Eigenschaft 'Passend erweitern' definiert werden.

#### **Für RichText-Inhalte**

Durch selbst umbrechende Felder wird der Inhalt des RichTextes in einzelne Zeilen aufgeteilt. Erstreckt sich eine RichText-Formatierung wie zum Beispiel Fett/Unterstrichen etc. über mehrere Zeilen so, kann es vorkommen, dass diese RichText-Formatierungen zwar für die erste Zeile, aber nicht für die Folgezeilen übernommen wird.

Die ist leider eine Technologie-Einschränkung von Adobe, welche profiforms versucht im ReportWriter zu umgehen. Für einzelne RichText-Formatierungen ist es möglich diese Formatierungen auf Folgezeilen auszudehnen, jedoch nicht für alle. Bitte prüfen Sie vor Verwendung dieser Funktion ob die gewünschten Formatierungen übernommen werden können (Abhängigkeit zur verwendeten Version des ReportWriters).

**Ergebnis:** Dynamischer RichText aus dem R-S

 Funktioniert nicht auf DocDef-Ebene.

## HTTP-Status-Codes durch Client- oder Server-Fehler im WEB-Browser

Unabhängig von möglichen Fehlern in der R-S Applikation kann es zu unterschiedlichen Fehlern im Applikations-Server des R-S kommen. Diese Fehler sind z.B. durch Abbrüche in der Netzwerk-Kommunikation zwischen WEB-Browser oder bei Eingabe einer fehlerhaften Adresse (URL) des R-S möglich. Fehler, welche am häufigsten oder auch üblicherweise durch Fehleingabe von Anwendern entstehen, werden nach Möglichkeit durch den Applikations-Server des R-S oder durch das R-S selbst behandelt. Eine ausführliche Beschreibung von HTTP-Statuscodes ist unter [HTTP-Statuscodes](#) zu finden.

Einige wesentliche Fehler werden im Folgenden kurz aufgelistet:

Fehler-Nummer	Fehler-Bezeichnung	Fehler-Beschreibung
401	Unauthorised	Die Aktion/Anfrage, welche an das R-S gestellt worden ist, kann nicht ohne Authentifizierung durchgeführt werden. Kontaktieren Sie Ihren Systemadministrator.
403	Forbidden	Die Aktion/Anfrage, welche an das R-S gestellt worden ist, kann mangels Berechtigung nicht ausgeführt werden. Mögliche Ursachen sind u.a., dass Anfragen mit dem HTTPS-Protokoll erwartet werden (https://...), aber vom Client/WEB-Browser via HTTP-Protokoll (http://...) gestellt worden sind. Problemlösung: Prüfen Sie die eingegebene Adresse im WEB-Browser, ob Sie das HTTP- oder HTTPS-Protokoll nutzen. Sollten Sie sich nicht sicher sein, so kontaktieren Sie Ihren Systemadministrator.

Fehler-Nummer	Fehler-Bezeichnung	Fehler-Beschreibung
404	Not found	<p>Die angeforderte Ressource wurde nicht gefunden. Dieser Statuscode kann ebenfalls verwendet werden, um eine Anfrage ohne näheren Grund abzuweisen. Links, welche auf solche Fehlerseiten verweisen, werden auch als 'Tote Links' bezeichnet. Ursache ist die Eingabe einer Adresse im WEB-Browser, welche im ersten Teil die gültige Adresse des R-S Applikations-Servers enthält, im zweiten Teil aber eine Adresse, welche im R-S nicht existiert.</p> <div data-bbox="355 701 1433 844" style="border: 1px solid black; padding: 5px;"> <p><b>Beispiel</b></p> <pre>http://localhost:8080/rs/hier_ist_der_Fehler;</pre> </div> <p>In diesem Beispiel ist die Adresse bis 'http://localhost:8080/rs' korrekt, nur der darauffolgende Teil der WEB-Adresse 'hier_ist_der_Fehler' ist fehlerhaft, solch eine WEB-Seite existiert nicht im R-S.</p> <p><u>Problemlösung:</u> Bitte prüfen Sie, ob die eingegebene WEB-Adresse des R-S korrekt ist und korrigieren Sie diese.</p> <p> Durch System-/Wartungsarbeiten kann das R-S u.a. ebenfalls nicht zur Verfügung stehen, in diesem Fall erhalten Sie ggf. ebenfalls den Fehlercode 404.</p>

## Restore

Ein vollständiges Restore erhalten Sie, indem Sie ein **Vollständiges Backup** wie folgt wiederherstellen:

1. Beenden Sie das Redaktions-System und löschen Sie das komplette Verzeichnis des R-S
2. Kopieren Sie das komplette R-S Verzeichnis aus Ihrem Backup in das im Schritt 1 gelöschte R-S Verzeichnis
3. Beenden Sie das Datenbank-System und löschen Sie das komplette Verzeichnis des Datenbank-Systems
4. Kopieren Sie das komplette Datenbank-System aus Ihrem Backup in das im Schritt 1 gelöschte Datenbank Verzeichnis
5. Starten Sie das Datenbank-System neu
6. Starten Sie das R-S neu

### **Zurücksetzen eines R-S auf den Initial-Zustand**

#### **Zurücksetzen durch Neuinstallation**

Die favorisierte Methode für das Zurücksetzen eines R-S auf den Initial-Zustand ist die Neu-Installation des R-S. Gehen Sie dabei wie folgt vor:

1. Beenden Sie das Redaktions-System und löschen Sie das komplette Verzeichnis des R-S
2. Beenden Sie das Datenbank-System und löschen Sie das komplette Verzeichnis des Datenbank-Systems

Die Beschreibung zur Installation finden Sie hier: [Installation](#)

#### **Zurücksetzen durch Neukonfiguration**

Wurde ein R-S auf einem Server-System erfolgreich installiert, konfiguriert und betrieben so kann es unter Umständen notwendig sein, dass der File-Storage und die Datenbank des R-S auf einen initialen Zustand zurückgesetzt werden soll. Damit erspart sich ein Systemadministrator den Aufwand für eine Neuinstallation, wichtige Konfigurationen des R-S Servers bleiben jedoch erhalten.

Arbeitet das R-S in einem ausschließlich dafür betriebenen Applikations-Container (hier Apache TOMCAT), gehen Sie für das Zurücksetzen des R-S nach folgender Checkliste vor:

#### Schritt 1 - Vorarbeiten

- vergewissern Sie sich das keine Nutzer mehr auf diesem R-S aktiv sind (Administrator-Client)
- öffnen Sie im Verzeichnis [RS\_INSTALL]/conf die Konfigurations-Datei des R-S Servers mit einem Editor und notieren Sie sich den Verzeichnis-Pfad der Eigenschaft 'rs.storage.workdir' als Verzeichnis für den File-Storage  
Hinweis: Im R-S ist das Verzeichnis [RS\_INSTALL]/runtime das Standard-Verzeichnis für den File-Storage, wenn die Eigenschaft 'rs.storage.workdir' nicht gesetzt ist.  
prüfen Sie im Vorfeld ob externe Systeme (docxworld, OMS/EOMS) auf das R-S zugreifen; ist dies der Fall, so planen Sie gleichzeitig das Upgrade der Schnittstellen der externen Systeme

#### Schritt 2 - Initialisierung/Update der Datenbank-Struktur

- starten Sie den von Ihnen gewählten SQL-Datenbank-Client, verbinden Sie sich auf die Datenbank des R-S
- führen Sie aus der Beschreibung der **Datenbank-Installation** die Schritte
  - + Verbinden mi einer bestehenden Datenbank
  - + Erzeugen der R-S Datenbank-Struktur aus

### Schritt 3 - Löschen der nicht mehr benötigten Daten

- beenden Sie den R-S Applikation
- löschen Sie alle Dateien/Verzeichnisse/Unterverzeichnisse unterhalb des im Schritt 1 notierten Verzeichnisses für den File-Storage

### Schritt 4 - Start der R-S Applikation

- starten Sie den System-Dienst des R-S (TOMCAT wird gestartet)

### Schritt 5 - Nacharbeiten

- prüfen Sie wenige Minuten nach dem Start des R-S im Verzeichnis [RS\_INSTALL]/logs ob die Datei rs.info.log existiert und eine Größe von mehr als 0 kb hat.  
(wenn Sie die Datei öffnen, sollte darin die Zeichenkette "started version X.Y...." zu finden sein; anderenfalls prüfen Sie die ausgegeben Fehler oder kontaktieren Sie unseren [Support](#))
- sorgen Sie dafür, dass alle Nutzer des Systems vom Abschluss des Upgrade informiert sind

## R-S Server Konfigurationsdatei

Über die Konfigurationsdatei 'rs.server.properties' können wesentliche Eigenschaften des R-S konfiguriert werden.

### Konfigurationsdatei für den R-S Server

Die Konfiguration befindet sich in der Datei **rs.server.properties**, welche im Verzeichnis [RS\_INSTALL]/conf zu finden ist. Für die meisten der definierten Eigenschaften gibt es meist programmtechnisch vorgegebene, sinnvolle Standardwerte. Aus diesem Grund sollten die Eigenschaften nur über die Konfigurationsdatei definiert werden, wenn die Vorgabe-Werte des Programms bewusst überschrieben werden sollen. Dies ist zum Beispiel für die Definition von Backup-Zyklen der R-S Datenbank sinnvoll.

 Der Aufbau der Konfigurationsdatei orientiert sich am **Nucleus properties file format**.

**Folgende Einträge sind in dieser Konfigurationsdatei beispielsweise möglich (nur Werte, welche sinnvollerweise durch Administratoren geändert werden sollten):**

Eintrag	Beschreibung	Mögliche Werte	Standardwert
<b>Allgemein</b>			
rs.auth.password.min-length	Die Mindestlänge für Passwörter.	Numerisch, >= 0	0
rs.auth.email-validation	Gibt an, ob geprüft werden soll, ob eingegebene E-Mail-Adressen gültige Form haben.	false   true	false
rs.auth.token-expire-time	Gibt an, wie lange Sicherheitstoken gültig bleiben.	Numerisch, >= 0, in ms	300000
rs.auth.expired-token-collector.start	Gibt an, nach wie vielen ms der Collector nach Start des RS das erste mal startet	Numerisch, >= 0, in ms	100000
rs.auth.expired-token-collector.repeat	Gibt an, nach wie vielen ms der Collector seine Aufgabe wiederholt	Numerisch, >= 0, in ms	100000
<b>rs.server.name</b>	Name des Redaktionssystems	Beliebig	

rs.server.serial-number	Seriennummer des Redaktions-Systems	Beliebig	
rs.server.help-url	Url zu Dokumentation	URL	
<b>Maintenance</b>			
rs.server.server-maintenance.enable-backup	Gibt an, ob periodische Backups aktiviert werden sollen (empfohlen!).	false   true	true
rs.server.server-maintenance.schedule-backup	<p>Parameter A: Anzahl der Sekunden [0-59]            Parameter B: Anzahl der Minuten [0-59]            Parameter G: Das Jahr der Ausführung [leer JJJJ-JJJJ]            Platzhalter:            '*' - steht für alle Werte die in dem Feld erlaubt sind (steht der Wert '*' im Parameter D, so wird das Backup jeden Tag im Monat ausgeführt)            '?' - als nicht definierten Wert</p> <p>Weitere Informationen unter: <a href="http://quartz-scheduler.org/documentation/quartz-1.x/tutorials/crontrigger">http://quartz-scheduler.org/documentation/quartz-1.x/tutorials/crontrigger</a></p>	<p>&lt;A&gt; &lt;B&gt; &lt;C&gt;            &lt;D&gt; &lt;E&gt; &lt;F&gt;            &lt;G&gt;</p>	0 0 * * * ?
rs.server.db-backup-dir	Ist das Verzeichnis, in welchem die zyklisch erstellten Datenbank-Backups für das R-S abgelegt werden.	Pfad	../data/rs/db-backup
<b>Datenbank</b>			
rs.server.dao	Gibt den Typ des genutzten Datenbank-Systems für das R-S an.	mysql   mssql   derby	derby
rs.core.db.driver	Beschreibt den JDBC-Treiber für das verwendete Datenbank-System. Die JDBC-Treiber werden in Form von JAR-Dateien vom Datenbank-Hersteller zur Verfügung gestellt und in das Verzeichnis [RS_INSTALL]/lib kopiert.	org.apache.derby.jdbc.ClientDriver   com.microsoft.sqlserver.jdbc.SQLServerDriver   com.mysql.jdbc.Driver	org.apache.derby.jdbc.ClientDriver
rs.core.db.url	Die Verbindungs-URL, unter der der angegebene Datenbankserver erreichbar ist.	Beliebig, typische Form: <TREIBER>:<DB>://<IP>:<PORT>/<DBNAME>;<PARAMETER...>	jdbc:derby://localhost:8888/rs;create=false

rs.core.db.user	Ist der Nutzernamen, mit dem sich das R-S an der Datenbank anmeldet.	Beliebig	rs
rs.core.db.password	Ist das Passwort für den Nutzer, mit dem sich das R-S an der Datenbank anmeldet.	Beliebig	rs
rs.core.db.schema	Ist der Name des verwendeten Datenbankschemas auf dem DB-System.	Beliebig	rs
rs.core.db.enable-structure-check	Gibt an, ob bei Start die Struktur der Datenbank geprüft werden soll.	false   true	true
rs.server.db.connect-retries	Gibt die maximale Anzahl fehlgeschlagener Verbindungsaufbauversuche an, bevor das System keine weiteren Versuche unternimmt.	Numerisch, >= 0	10
rs.server.db.connect-timeout	Gibt an, wie lange (in ms) gewartet werden soll, bevor nach einem fehlgeschlagenen Verbindungsaufbauversuch ein neuer Versuch unternommen wird.	Numerisch, >= 0	1000
rs.server.db.query-retries	Gibt die maximale Anzahl fehlgeschlagener Query Versuchen an, bevor das System keine weiteren Versuche unternimmt. (z.B. bei Docker-Umgebungen)	Numerisch, >= 0	2
rs.server.db.query-timeout	Gibt an, wie lange (in ms) gewartet werden soll, bevor nach einem fehlgeschlagenen Query ein neuer Versuch unternommen wird.	Numerisch, >= 0	100
<b>Job</b>			
rs.job.terminated-live-time	Ist die Anzahl der Millisekunden (der Standardwert bedeutet 24 Stunden) nach welchen terminierte Jobs	Numerisch, >= 0, in ms	86400000
rs.job.terminated-job-collector.start	Ist die Anzahl der Millisekunden nach Systemstart (der Standardwert bedeutet 6 Stunden), nach welchen die Löschroutine für terminierte Jobs gestartet wird.  Wird dieser Wert zu klein gewählt (z.B. 1 Sekunde), so kann es zu Verzögerungen beim Systemstart des R-S kommen.	Numerisch, >= 0, in ms	21600000
rs.job.terminated-job-collector.repeat	Ist die Anzahl der Millisekunden für das Intervall (der Standardwert bedeutet 6 Stunden), nach welchen sich die Löschroutine für terminierte Jobs wiederholt (seit der letzten Löschroutine).  Wird dieser Wert zu klein gewählt (zum Beispiel 1 Sekunde), kann es zu einer hohen Systembelastung des R-S durch Löschanforderungen kommen.	Numerisch, >= 0, in ms	21600000

rs.job.expire d-job- collector.star t	Gibt an, nach wie vielen ms der Collector nach Start des RS das erste mal startet	Numerisch, >= 0, in ms	60000
rs.job.expire d-job- collector.rep eat	Gibt an, nach wie vielen ms der Collector seine Aufgabe wiederholt	Numerisch, >= 0, in ms	60000
<b>Workspace</b>			
rs.workspace .acl.auto- repair- system-users	R-S 2.4 erlaubt keine Zuweisung von Arbeitsbereichen an Systemnutzer. Frühere Zuweisungen, die vor einem Upgrade auf Version 2.4 durchgeführt wurden, können daher Probleme bereiten. rs.workspace.acl.auto-repair-system-users gibt an, ob Systemnutzer (ADMIN / PRODUCTION) automatisch aus ihren zugewiesenen Arbeitsbereichen entfernt werden, wenn dies das Redaktions-System am Start hindert.	true   false	false
rs.workspace .txform.inacti ve-expire- time	Gibt an nach wie vielen ms ein gelöscht/inaktives Tx-Form ohne aktive Abhängigkeiten endgültig aus dem System gelöscht wird.	Numerisch, >= 0, in ms	86400000
rs.workspace .txform.inacti ve- collector.star t	Gibt die Verzögerung in ms nach Systemstart für das Starten der Löschroutine für Transaktions-Formulare an.  Die Löschroutine prüft für Objekte mit Status DELETED, ob diese noch Abhängigkeiten besitzen. Ist dies nicht der Fall, wird das Objekt durch die Löschroutine endgültig gelöscht.  Wird dieser Wert zu klein gewählt (z.B. 1 Sekunde), so kann es zu Verzögerungen beim Systemstart des R-S kommen.	Numerisch, >= 0, in ms	21600000
rs.workspace .txform.inacti ve- collector.rep eat	Gibt die Intervalllänge in ms für die Ausführung der Löschroutine von Transaktions-Formularen an. Die Löschroutine wird periodisch nach Ablauf der Intervalllänge gestartet.  Die Löschroutine prüft für Objekte mit Status DELETED, ob diese noch Abhängigkeiten besitzen. Ist dies nicht der Fall, wird das Objekt durch die Löschroutine endgültig gelöscht.  Wird dieser Wert zu klein gewählt (z.B. 1 Minute), kann es zu einer hohen Systembelastung des R-S durch Löschanforderungen kommen.	Numerisch, >= 0, in ms	21600000
rs.workspace .txform- revision.inacti ve-expire- time	Gibt an nach wie vielen ms eine gelöschte/inaktive Tx-Form-Version ohne aktive Abhängigkeiten endgültig aus dem System gelöscht wird.	Numerisch, >= 0, in ms	86400000

<p>rs.workspace .txform- revision.inact ive- collector.star t</p>	<p>Gibt die Verzögerung in ms nach Systemstart für das Starten der Löschroutine für Transaktions-Formular-Versionen an.</p> <p>Die Löschroutine prüft für Objekte mit Status DELETED, ob diese noch Abhängigkeiten besitzen. Ist dies nicht der Fall, wird das Objekt durch die Löschroutine endgültig gelöscht.</p> <p>Wird dieser Wert zu klein gewählt (z.B. 1 Sekunde), so kann es zu Verzögerungen beim Systemstart des R-S kommen.</p>	<p>Numerisch, &gt;= 0, in ms</p>	<p>21600000</p>
<p>rs.workspace .txform- revision.inact ive- collector.rep eat</p>	<p>Gibt die Intervalllänge in ms für die Ausführung der Löschroutine von Transaktions-Formular-Versionen an. Die Löschroutine wird periodisch nach Ablauf der Intervalllänge gestartet.</p> <p>Die Löschroutine prüft für Objekte mit Status DELETED, ob diese noch Abhängigkeiten besitzen. Ist dies nicht der Fall, wird das Objekt durch die Löschroutine endgültig gelöscht.</p> <p>Wird dieser Wert zu klein gewählt (z.B. 1 Minute), kann es zu einer hohen Systembelastung des R-S durch Löschanforderungen kommen.</p>	<p>Numerisch, &gt;= 0, in ms</p>	<p>21600000</p>
<p>rs.workspace .schema.inac tive-expire- time</p>	<p>Gibt an nach wie vielen ms ein gelöschttes/inaktives Schema ohne aktive Abhängigkeiten entgültig aus dem System gelöscht wird.</p>	<p>Numerisch, &gt;= 0, in ms</p>	<p>86400000</p>
<p>rs.workspace .schema.inac tive- collector.star t</p>	<p>Gibt die Verzögerung in ms nach Systemstart für das Starten der Löschroutine für Schemas an.</p> <p>Die Löschroutine prüft für Objekte mit Status DELETED, ob diese noch Abhängigkeiten besitzen. Ist dies nicht der Fall, wird das Objekt durch die Löschroutine endgültig gelöscht.</p> <p>Wird dieser Wert zu klein gewählt (z.B. 1 Sekunde), so kann es zu Verzögerungen beim Systemstart des R-S kommen.</p>	<p>Numerisch, &gt;= 0, in ms</p>	<p>21600000</p>
<p>rs.workspace .schema.inac tive- collector.rep eat</p>	<p>Gibt die Intervalllänge in ms für die Ausführung der Löschroutine von Schemas an. Die Löschroutine wird periodisch nach Ablauf der Intervalllänge gestartet.</p> <p>Die Löschroutine prüft für Objekte mit Status DELETED, ob diese noch Abhängigkeiten besitzen. Ist dies nicht der Fall, wird das Objekt durch die Löschroutine endgültig gelöscht.</p> <p>Wird dieser Wert zu klein gewählt (z.B. 1 Minute), kann es zu einer hohen Systembelastung des R-S durch Löschanforderungen kommen.</p>	<p>Numerisch, &gt;= 0, in ms</p>	<p>21600000</p>

rs.workspace .schema- revision.inact ive-expire- time	Gibt an nach wie vielen ms eine gelöschte/inaktive Schema-Version ohne aktive Abhängigkeiten endgültig aus dem System gelöscht wird.	Numerisch, >= 0, in ms	86400000
rs.workspace .schema- revision.inact ive- collector.star t	Gibt die Verzögerung in ms nach Systemstart für das Starten der Löschroutine für Schema-Versionen an.  Die Löschroutine prüft für Objekte mit Status DELETED, ob diese noch Abhängigkeiten besitzen. Ist dies nicht der Fall, wird das Objekt durch die Löschroutine endgültig gelöscht.  Wird dieser Wert zu klein gewählt (z.B. 1 Sekunde), so kann es zu Verzögerungen beim Systemstart des R-S kommen.	Numerisch, >= 0, in ms	21600000
rs.workspace .schema- revision.inact ive- collector.rep eat	Gibt die Intervalllänge in ms für die Ausführung der Löschroutine von Schema-Versionen an. Die Löschroutine wird periodisch nach Ablauf der Intervalllänge gestartet.  Die Löschroutine prüft für Objekte mit Status DELETED, ob diese noch Abhängigkeiten besitzen. Ist dies nicht der Fall, wird das Objekt durch die Löschroutine endgültig gelöscht.  Wird dieser Wert zu klein gewählt (z.B. 1 Minute), kann es zu einer hohen Systembelastung des R-S durch Löschanforderungen kommen.	Numerisch, >= 0, in ms	21600000
rs.workspace .library.inact ive-expire- time	Gibt an nach wie vielen ms eine gelöschte/inaktive Bibliothek ohne aktive Abhängigkeiten endgültig aus dem System gelöscht wird.	Numerisch, >= 0, in ms	86400000
rs.workspace .library.inact ive- collector.star t	Gibt die Verzögerung in ms nach Systemstart für das Starten der Löschroutine für Bibliotheken an.  Die Löschroutine prüft für Objekte mit Status DELETED, ob diese noch Abhängigkeiten besitzen. Ist dies nicht der Fall, wird das Objekt durch die Löschroutine endgültig gelöscht.  Wird dieser Wert zu klein gewählt (z.B. 1 Sekunde), so kann es zu Verzögerungen beim Systemstart des R-S kommen.	Numerisch, >= 0, in ms	21600000

<p>rs.workspace.library.inactive-collector.repeat</p>	<p>Gibt die Intervalllänge in ms für die Ausführung der Löschroutine von Bibliotheken an. Die Löschroutine wird periodisch nach Ablauf der Intervalllänge gestartet.</p> <p>Die Löschroutine prüft für Objekte mit Status DELETED, ob diese noch Abhängigkeiten besitzen. Ist dies nicht der Fall, wird das Objekt durch die Löschroutine endgültig gelöscht.</p> <p>Wird dieser Wert zu klein gewählt (z.B. 1 Minute), kann es zu einer hohen Systembelastung des R-S durch Löschanforderungen kommen.</p>	<p>Numerisch, &gt;= 0, in ms</p>	<p>21600000</p>
<p>rs.workspace.library-revision.inactive-expire-time</p>	<p>Gibt an nach wie vielen ms eine gelöschte/inaktive Bibliothek-Version ohne aktive Abhängigkeiten endgültig aus dem System gelöscht wird.</p>	<p>Numerisch, &gt;= 0, in ms</p>	<p>86400000</p>
<p>rs.workspace.library-revision.inactive-collector.start</p>	<p>Gibt die Verzögerung in ms nach Systemstart für das Starten der Löschroutine für Bibliotheks-Versionen an.</p> <p>Die Löschroutine prüft für Objekte mit Status DELETED, ob diese noch Abhängigkeiten besitzen. Ist dies nicht der Fall, wird das Objekt durch die Löschroutine endgültig gelöscht.</p> <p>Wird dieser Wert zu klein gewählt (z.B. 1 Sekunde), so kann es zu Verzögerungen beim Systemstart des R-S kommen.</p>	<p>Numerisch, &gt;= 0, in ms</p>	<p>21600000</p>
<p>rs.workspace.library-revision.inactive-collector.repeat</p>	<p>Gibt die Intervalllänge in ms für die Ausführung der Löschroutine von Bibliotheks-Versionen an. Die Löschroutine wird periodisch nach Ablauf der Intervalllänge gestartet.</p> <p>Die Löschroutine prüft für Objekte mit Status DELETED, ob diese noch Abhängigkeiten besitzen. Ist dies nicht der Fall, wird das Objekt durch die Löschroutine endgültig gelöscht.</p> <p>Wird dieser Wert zu klein gewählt (z.B. 1 Minute), kann es zu einer hohen Systembelastung des R-S durch Löschanforderungen kommen.</p>	<p>Numerisch, &gt;= 0, in ms</p>	<p>21600000</p>
<p>rs.workspace.schema-revision-preview.expire-time</p>	<p>Ist die Anzahl der Millisekunden (der Standardwert bedeutet 24 Stunden) nach welchen die Ergebnisse einer Druckvorschau aus dem Storage des R-S gelöscht werden kann.</p> <p>Dieser Wert sollte nur angepasst werden, wenn durch eine Vielzahl von Nutzern eine hohe Menge von Druckvorschauen angefordert werden.</p>	<p>Numerisch, &gt;= 0, in ms</p>	<p>86400000</p>

rs.workspace .schema- revision- preview.expir ed- collector.star t	<p>Gibt die Verzögerung in ms nach Systemstart für das Starten der Löschroutine für Vorschauen (von Schema-Versionen) an.</p> <p>Die Löschroutine prüft für Objekte mit Status DELETED, ob diese noch Abhängigkeiten besitzen. Ist dies nicht der Fall, wird das Objekt durch die Löschroutine endgültig gelöscht.</p> <p>Wird dieser Wert zu klein gewählt (z.B. 1 Sekunde), so kann es zu Verzögerungen beim Systemstart des R-S kommen.</p>	Numerisch, >= 0, in ms	21600000
rs.workspace .schema- revision- preview.expir ed- collector.rep eat	<p>Gibt die Intervalllänge in ms für die Ausführung der Löschroutine von Vorschauen (von Schema-Versionen) an. Die Löschroutine wird periodisch nach Ablauf der Intervalllänge gestartet.</p> <p>Die Löschroutine prüft für Objekte mit Status DELETED, ob diese noch Abhängigkeiten besitzen. Ist dies nicht der Fall, wird das Objekt durch die Löschroutine endgültig gelöscht.</p> <p>Wird dieser Wert zu klein gewählt (z.B. 1 Minute), kann es zu einer hohen Systembelastung des R-S durch Löschanforderungen kommen.</p>	Numerisch, >= 0, in ms	21600000
rs.workspace .production- bundle.dispa tcher.start	<p>Ist die Anzahl der Millisekunden (der Standardwert bedeutet 15 Minuten) nach welchem nach Systemstart des R-S geprüft wird, ob ein Produktions-Paket produktiv gesetzt werden muss.</p> <p>Wird dieser Wert zu klein gewählt (z.B. 30 Sekunde), kann es zu einer hohen Systembelastung des R-S durch Prüfanforderungen kommen.</p>	Numerisch, >= 0, in ms	900000
rs.workspace .lock- collector.star t		Numerisch, >= 0, in ms	30000
rs.workspace .lock- collector.rep eat		Numerisch, >= 0, in ms	30000
rs.workspace .lock- collector.sch edule.repeat		Numerisch, >= 0, in ms	36000000
rs.workspace .lock- collector.sch edule.start		Numerisch, >= 0, in ms	5000

rs.workspace.locks.hide-owners		Boolean	false
<b>Upload</b>			
rs.workspace.upload-check.enabled	Gibt an, ob bei Dateiuploads Format und Größe limitiert sein sollen.	false   true	true
rs.workspace.upload-check.allowed-file-extensions	Komma-getrennte Liste erlaubter Dateiformate für Dateiuploads (nur falls rs.workspace.upload-check.enabled = true).	bmp   tiff   jpeg   jpg   pdf   png   gif   xdp	bmp,tiff,jpeg,jpg,pdf,png,gif
rs.workspace.upload-check.allowed-file-size	Maximale Dateigröße (nur falls rs.workspace.upload-enabled = true). Dieser Parameter wird nur beim Upload von Binären Ressourcen in Schema-Versionen oder zum Beispiel in Bibliotheken angewendet. Der Upload von Binär-Paketen oder zum Beispiel von Schema-Versionen welche importiert werden - sind davon nicht betroffen.	Numerisch, >= 0, in Bytes oder -1 für keine Beschränkung	-1
rs.workspace.upload-check.content-check	(nur im Falle von command-Line und upload-check aktiv)Gibt an welche Methode für den Upload-Check verwendet werden soll. Abhängig davon sind nachfolgende Parameter.	command-line   bouncer	command-line
rs.workspace.upload-check.command-line	Ist das Programm, welches gestartet wird um Dateien, die in das R-S geladen werden vor der Speicherung zu prüfen. Diese Prüfung kann z.B. ein Prüfung auf Viren oder eine inhaltliche Prüfung sein. Beendet das angegebene Programm/Script mit einem Rückgabewert ungleich 0, so wird die Speicherung im R-S mit Fehler abgebrochen, der Nutzer erhält einen Hinweis.	Kommandozeilenübergabe	antivir.bat{{checkFile}}
rs.workspace.upload-check.bouncer.dir	(nur im Falle von bouncer und upload-check aktiv) Verzeichnis des Bouncers	Pfad	runtime/ bouncer
rs.workspace.upload-check.bouncer.config-path	(nur im Falle von bouncer und upload-check aktiv) Konfigurationsdatei des Bouncers	Pfad	runtime/ bouncer/ rs.bouncer. unix.config

Vorschau			
rs.workspace .schema- revision- preview.work er.keep- previewdir	<p>Pro Druckvorschau des R-S wird einem Verzeichnis mit einem eindeutigen Namen unterhalb des definierten Preview-Verzeichnisses angelegt. Ist die Druckvorschau erfolgreich beendet wird dieses Verzeichnis in Abhängigkeit dieser Option gelöscht oder beibehalten.</p> <p>Wird die Druckvorschau mit einem fehlerhaftem Returncode beendet, bleibt das Verzeichnis für die Druckvorschau unabhängig von dieser Option, erhalten.</p> <p>Bleiben Verzeichnisse für die einzelnen Druckvorschauen erhalten, muss seitens des System-Managements für ein zeitgestütztes Löschen dieser Dateien gesorgt werden.</p>	false   true	false
Temporäre Dateien und Verzeichnisse			
rs.workspace .upload- check.worker .dir	Verzeichnis, in welchem hochgeladene Dateien für das R-S geprüft werden.	Pfad	../data/rs/ tmp/upload
rs.workspace .txform- revision.work er.dir	Verzeichnis, in welchem Transaktions-Formulare und Input-Pakete temporär abgelegt werden.	Pfad	../data/rs/ tmp/txform
rs.workspace .production- bundle.work er.dir	Verzeichnis, in welchem Produktions-Pakete temporär abgelegt werden.	Pfad	../data/rs/ tmp/bundle
<b>rs.workspace</b> <b>e.schema.io</b> .workdir	Verzeichnis, in welchem Schema-Versionen temporär abgelegt werden.	Pfad	../data/rs/ tmp/io
<b>rs.workspace</b> <b>e.library.io</b> .workdir	Verzeichnis, in welchem Bibliotheks-Versionen temporär abgelegt werden.	Pfad	../data/rs/ tmp/io
<b>rs.workspace</b> <b>e.resources.i</b> <b>o</b> .workdir	Verzeichnis, in welchem Ressourcen temporär abgelegt werden.	Pfad	../data/rs/ tmp/io
rs.bin.binary- bundle.fetch er.dir	Verzeichnis, in welchem Binär-Pakete temporär abgelegt werden.	Pfad	../data/rs/ tmp/binary
<b>rs.bin.binary-</b> <b>bundle.io</b> .workdir	Verzeichnis, in welchem Binär-Pakete temporär abgelegt werden.		

rs.workspace .schema- revision- preview.work er.dir	Ist das Verzeichnis, in welchem die Druckvorschau aus dem R-S aufbereitet wird.	Pfad	../data/rs/ tmp/ preview
Verzeichnisk onfiguration			
rs.storage.w orkdir	Ist das Verzeichnis in welchem der File-Storage (Dateien des R-S) aufgebaut wird.	Pfad	../data/rs/ storage
rs.storage.up load- check.enable d	Gibt an, ob beim Upload in den Storage eine Überprüfung stattfindet.	false   true	true
<b>Papier-Verarbeitungs-Optionen (Finsihing)</b>			
rs.workspace .finishing- mapping	<p>Ist eine Liste der Finishing-Optionen, welche als Seiten-/ Blattoptionen für logische Beileger zur Verfügung stehen. Hinweis: Die Liste der Finishing-Optionen, muss mit den technischen Möglichkeiten der verarbeitenden Systeme (z.B. Druck abgestimmt sein).</p> <p>Die Liste wird von zwei geschweiften Klammern {} umgeben. Einzelne Listenelemente werden durch Komma getrennt. Ein Listenelement ist definiert durch einen ersten Teil zur Beschreibung und in einem zweiten Teil mit einer Nummer zwischen 1-99. Der erster Teil (Beschreibung) ist in doppelten Anführungszeichen "" einzuschließen. Erster und zweiter Teil sind durch einen Doppelpunkt : und Leerzeichen von einander getrennt .</p> <p>Beispiele:</p> <pre>{ "22 - Zahlschein-Perforation DP65" : 22, "1 - Doppelte Lochung DP65" : 1 }</pre> <pre>{ "Ösung" : 23 }</pre>	siehe Beschreibung	(leer)

Änderungen an der Konfigurationsdatei werden erst nach einem Neustart des R-S Servers wirksam!

Kommentare starten in der ersten Position einer Zeile mit dem Zeichen '#'.

## Anleitungen

### ***R-S und ReportWriter***

In diesem Teil der Dokumentation finden Sie ausgewählte Themen, wenn als System zur Dokument-Generierung das Produkt ReportWriter eingesetzt wird.

Dies bedeutet das bei der Dokument-Generierung Teile der Dokument-Logik oder inhaltliche Teile des Dokumentes über das R-S eingebracht werden.

- ***Logische Beileger des R-S über SupplementWorkitem des ReportWriter***
- ***Optionale Beileger und Porto***
  - ***TCI des ReportWriters***
  - ***Konfigurationen des Transaktions-Formulars***
  - ***Konfiguration im R-S***
  - ***Beispiel - optionale Beileger***
  - ***Algorithmen***

## Logische Beileger des R-S über SupplementWorkitem des ReportWriters

Über das R-S können logische und physikalische Beileger definiert werden. Physikalische Beileger haben fest definierte Regeln, denn es sind vorgedruckte Materialien, welche über die Kuvertierstraße dem Dokument beigesteuert werden. Aus diesem Grund benötigt der ReportWriter für die postalische Aufbereitung nur die Material-Informationen des physikalischen Beilegers wie zum Beispiel das Gewicht oder die Dimension. Andere Angaben sind für die Verwendung von physikalischen Beilegern im ReportWriter nicht notwendig.

Anders verhält sich dies jedoch für die logischen Beileger

.....

An dem Rest des Artikels wird noch gearbeitet!!!

In den Eigenschaften für die Erweiterungen kann für die entsprechenden Ressourcen von logischen/ physikalischen Beilegern angegeben werden, ob diese im Produktionsprozess des Dokuments obligatorisch oder optional berücksichtigt werden sollen.

### Ein Beispiel zum Verständnis:

Ein Rechnungs-Dokument besteht aus einem Blatt Papier. Für dieses Rechnungs-Dokument sind nun zwei logische und ein physikalischer Beileger definiert:

- 'Logischer Beileger 1' - ist **optional/simplex** und stellt im R-S ein Blatt mit kundenspezifischer Werbung dar
- 'Logischer Beileger 2' - ist **optional/simplex** und stellt im R-S ein Blatt mit Standard-Produktwerbung dar
- 'Logischer Beileger 3' - ist **obligatorisch/simplex** und stellt im R-S ein Blatt mit AGB's dar

Sind in der Konfiguration des ReportWriters und in der Konfiguration des Transaktions-Formulars **keine** Informationen zur Verarbeitung von optionalen Beilegern definiert, generiert der ReportWriter ein postalisches Dokument mit folgendem Aufbau:

- Briefumschlag
- ein Blatt der Rechnung
- 'Logischer Beileger 1'
- 'Logischer Beileger 2'
- 'Logischer Beileger 3'

Legt man als Versender die Deutsche Post AG fest und geht man von Standard-Papiergewichten der Dokument-Bestandteile aus, so ergibt sich damit ein Brief, welcher nicht mehr mit dem Porto (aktuell 70 Cent; im Jahr 2016) eines Standard-Briefes versendet werden kann. Grund dafür: Der Inhalt des Briefes ist schwerer als 20 Gramm.

Durch die optionalen Werbungsbestandteile des Briefes ist dieser zu schwer geworden und muss nun mit dem Porto eines Kompakt-Briefes versendet werden (aktuell 85 Cent; im Jahr 2016). Also 15 Cent höhere Kosten. Für manche Anwender ist der Wert der Werbungszusteuern höher als die höheren Porto-Kosten (in diesem Fall sollten die logischen/physikalischen Beileger im R-S auf 'obligatorisch' gesetzt werden). Die weitaus meiste Zahl der Anwender möchte aber Werbung nur dann zusteuern, wenn dadurch kein höheres Porto verursacht wird - wie dies im R-S und im ReportWriter eingerichtet wird, beschreiben die folgenden Ausführungen.

Die Konfiguration erfolgt in folgenden Stufen:

1. Arbeiten in der ***TCI des ReportWriters***
2. ***Konfiguration des Transaktion-Formulars***
3. ***Konfiguration im R-S***

## Optionale Beileger und Porto

Über das R-S können logische und physikalische Beileger definiert werden. In den Eigenschaften für die Erweiterungen kann für die entsprechenden Ressourcen von logischen/physikalischen Beilegern angegeben werden, ob diese im Produktionsprozess des Dokuments obligatorisch oder optional berücksichtigt werden sollen.

### Ein Beispiel zum Verständnis:

Ein Rechnungs-Dokument besteht aus einem Blatt Papier. Für dieses Rechnungs-Dokument sind nun zwei logische und ein physikalischer Beileger definiert:

- 'Logischer Beileger 1' - ist **optional/simplex** und stellt im R-S ein Blatt mit kundenspezifischer Werbung dar
- 'Logischer Beileger 2' - ist **optional/simplex** und stellt im R-S ein Blatt mit Standard-Produktwerbung dar
- 'Logischer Beileger 3' - ist **obligatorisch/simplex** und stellt im R-S ein Blatt mit AGB's dar

Sind in der Konfiguration des ReportWriters und in der Konfiguration des Transaktions-Formulars **keine** Informationen zur Verarbeitung von optionalen Beilegern definiert, generiert der ReportWriter ein postalisches Dokument mit folgendem Aufbau:

- Briefumschlag
- ein Blatt der Rechnung
- 'Logischer Beileger 1'
- 'Logischer Beileger 2'
- 'Logischer Beileger 3'

Legt man als Versender die Deutsche Post AG fest und geht man von Standard-Papiergewichten der Dokument-Bestandteile aus, so ergibt sich damit ein Brief, welcher nicht mehr mit dem Porto (aktuell 70 Cent; im Jahr 2016) eines Standard-Briefes versendet werden kann. Grund dafür: Der Inhalt des Briefes ist schwerer als 20 Gramm.

Durch die optionalen Werbungsbestandteile des Briefes ist dieser zu schwer geworden und muss nun mit dem Porto eines Kompakt-Briefes versendet werden (aktuell 85 Cent; im Jahr 2016). Also 15 Cent höhere Kosten. Für manche Anwender ist der Wert der Werbungszusteuern höher als die höheren Porto-Kosten (in diesem Fall sollten die logischen/physikalischen Beileger im R-S auf 'obligatorisch' gesetzt werden). Die weitaus meiste Zahl der Anwender möchte aber Werbung nur dann zusteuern, wenn dadurch kein höheres Porto verursacht wird - wie dies im R-S und im ReportWriter eingerichtet wird, beschreiben die folgenden Ausführungen.

Die Konfiguration erfolgt in folgenden Stufen:

1. Arbeiten in der **TCl des ReportWriters**
2. **Konfiguration des Transaktion-Formulars**
3. **Konfiguration im R-S**

## TCI des ReportWriters

### **Grundlagen**

Anwender möchten gern die definierte Business-Logik für physikalische und logische Beileger schon im R-S testen und nicht erst die Ergebnisse der produktiven postalischen Verarbeitung begutachten. Der ReportWriter unterstützt diese Vorgehensweise, in dem der ReportWriter die Zusammenstellung von Dokumenten und deren Beilegern simuliert. Bei dieser Simulation wird keine Post-/Portoberechnung ausgeführt oder gar die Datamatrix-Codes für die DV-Freimachung erzeugt.

Findet der ReportWriter beim Start der Verarbeitung im definierten 'etc'-Verzeichnis eine Datei 'psp.ini', so wird eine Simulation der postalischen Dokument-Erzeugung angestoßen. Ob diese Simulation der Sendungsverarbeitung durchgeführt wird oder nicht, ist jedoch nicht nur vom Vorhandensein der 'psp.ini' abhängig. Der ReportWriter erwartet in der TCI weitere Parameter an den DOCDEF- und WORKITEM-Objekten in der TCI des ReportWriters. Fehlen dort wichtige Einträge so wird keine Simulation der Sendungsverarbeitung durchgeführt. Welche Parameter der TCI für eine Simulation notwendig sind, ist in den nächsten Abschnitten beschrieben.

### **Einschalten der zentralen Postsimulation auf Dokument-Ebene im DOCDEF**

Über das TCI-Objekt DOCDEF wird nun für die Dokumente welche mit Post-/Portooptionen generiert werden sollen der zentrale Druck aktiviert. Dies geschieht über das Attribut PRINTER im TCI-Objekt DOCDEF. Steht das Attribut PRINTER auf dem Wert 'CENTRAL', so wird die Berücksichtigung von Post- und Portooptionen bei der Dokumentenaufbereitung des entsprechenden Dokumentes grundsätzlich aktiviert. Ob es dann zu entsprechenden postalischen Bewertungen kommt, hängt dann jedoch im Wesentlichen von den Einstellungen des Dokumentes in den TCI-Objekten WORKITEM für dieses Dokument ab.

```
DocDef PSP_Example {
  Printer          CENTRAL
  WorkItems        PSP_Example
  SupplementWorkItem LogicalSupplements
}
```

### **Konfiguration der zentralen Postsimulation auf Dokument-Teil-Ebene im WORKITEM**

Über das TCI-Objekt WORKITEM wird nun für die einzelnen Dokument-Teile eine detaillierte Konfiguration vorgenommen.

Zuerst muss das Attribut 'CENTRAL' mit dem Wert '1' versehen werden um die Berücksichtigung von Post- und Portooptionen bei der Aufbereitung des Dokument-Teiles grundsätzlich zu aktivieren und es muss ebenfalls ein Attribut 'FIRSTPAGE' mit dem Namen der ersten Seite für das Design des Dokument-Teils definiert sein.

Für den führenden Dokument-Teil (z.B. bei Rechnungen mit logischen Beilegern aus dem R-S ist dies die Rechnung selbst), wird nun über das TCI-Objekt ENVELOPESORTSYSTEM die fachlichen Einstellungen für die Verarbeitung des Dokumentes mit Berücksichtigung der Post-/Portooptionen vorgenommen.

Im Attribut 'TYPE' ist der Typ der Kuvertieranlage zu hinterlegen, über welche das Dokument verarbeitet wird. Da zum Zeitpunkt der Dokument-Erzeugung im docxworld24-Umfeld noch gar nicht festgelegt ist über welche Kuvertieranlage das Dokument verarbeitet wird, ist der Wert 'NONE' zu konfigurieren.

Im Attribut 'CHANNEL' ist der Kanal der Kuvertieranlage zu hinterlegen, über welchen das Dokument verarbeitet wird. Da zum Zeitpunkt der Dokument-Erzeugung im docxworld24-Umfeld noch gar nicht festgelegt ist über welche Kuvertieranlage das Dokument verarbeitet wird, ist der Wert '1' zu konfigurieren.

Im Attribut 'PSP' ist der Postal-Service-Provider, also der postalische Versender des Dokumentes zu hinterlegen. Im Falle der Aufbereitung mit docxworld24, ist der Wert 'docxworld'. Dieser Wert korrespondiert mit dem Namen des POSTALSERVICEPROVIDER in der Konfigurationsdatei 'psp.ini' des ReportWriter.

 In der Konfigurationsdatei 'psp.ini' enthält postalische Konfigurationen für freie Postdienstleister (nicht die DP AG). Aufbau und Inhalt der Konfigurationsdatei 'psp.ini' ist in der **Online-Dokumentation des ReportWriters** im Bereich 'Freie Postdienstleister Konfiguration' zu finden.

 Die Konfigurationsdatei 'psp.ini' enthält postalische Informationen zu Sendungsgewichten, Sendungspreisen und dem Porto von Sendungen. In docxworld24 wird diese Konfigurationsdatei genutzt um die Sendungstypen, Gewichte und Preise der DP AG im docxworldPrepare-Prozess nutzen zu können. Ändern sich die Produkte, Gewichte oder Preise der DP AG, sind diese Informationen über die Konfiguration des Transaktions-Formulars vom **docxworld-Kunden** nachzupflegen und zu den entsprechenden Änderungsterminen der DP AG zu aktivieren.

Im Attribut 'CLASS' ist die Sendungsklasse anzugeben über welche ein Dokument versendet werden soll. Der Wert des Attributes 'CLASS' korrespondiert mit den möglichen Werten aus dem Attribut 'CLASS' der Konfigurationsdatei 'psp.ini'. Sollen Dokumente von Standard, über Kompakt bis zum MaxiPlus-Bief versendet werden wird in das Attribut 'CLASS' der Wert BRIEF' eingesetzt.

Im Attribut 'ZIPCODE' ist die Postleitzahl aus der Adresse jeden Dokumentes anzugeben. Typischerweise steht die Postleitzahl als Feld im Datenstrom zur Verfügung und wird mit der '@'-Annotation als Wert für das Attribut 'ZIPCODE' übergeben.

 Über die Funktion 'GetZIPFromAddress' bietet der ReportWriter auch die Möglichkeit die PLZ aus einem kompletten Adressfeld zu extrahieren.

Im Attribut 'COUNTRYCODE' ist der Ländercode aus der Adresse jeden Dokumentes anzugeben. Typischerweise steht der Ländercode als Feld im Datenstrom zur Verfügung und wird mit der '@'-Annotation als Wert für das Attribut 'COUNTRYCODE' übergeben.

 Der ReportWriter verfügt über vielfältige Möglichkeiten Standardwerte für nicht definierte Ländercodes anzugeben, bzw. unterschiedliche Formatierungen des Ländercodes zu akzeptieren. In der **Online-Dokumentation des ReportWriters** gibt es dazu weiterführende Informationen.

Ist die Postverarbeitung im führenden Teil des Dokumentes vollständig definiert, so erfolgt noch die Definition der Post-/Portoverarbeitung für nachrangige Dokument-Teile (z.B. Zahlscheine) oder auch für Workitems vom Typ 'SUPPLEMENTWORKITEM'. In diesen nachgeordneten Dokument-Teilen müssen nicht mehr alle postalischen Informationen wie PLZ, Ländercode etc. angegeben werden, diese sind schon am führenden Dokument-Teil definiert.

Für den nachgeordneten Dokument-Teil, wird nun über das TCI-Objekt ENVELOPESORTSYSTEM die fachlichen Einstellungen für die Verarbeitung des Dokumentes mit Berücksichtigung der Post-/Portoptionen vorgenommen.

Im Attribut 'TYPE' ist der Typ der Kuvertieranlage zu hinterlegen, über welche das Dokument verarbeitet wird. Da zum Zeitpunkt der Dokument-Erzeugung im docxworld24-Umfeld noch gar nicht festgelegt ist über welche Kuvertieranlage das Dokument verarbeitet wird, ist der Wert 'NONE' zu konfigurieren.

Im Attribut 'CHANNEL' ist der Kanal der Kuvertieranlage zu hinterlegen, über welchen das Dokument verarbeitet wird. Da zum Zeitpunkt der Dokument-Erzeugung im docxworld24-Umfeld noch gar nicht festgelegt ist über welche Kuvertieranlage das Dokument verarbeitet wird, ist der Wert '1' zu konfigurieren.

## Konfigurationen Transaktions-Formulars

Eine weitere Stelle zur Konfiguration der Art und Weise ob und wie obligate oder optional logische/physikalische Beileger bei der postalischen Dokument-Aufbereitung durch den ReportWriter behandelt werden - ist die Konfigurationsdatei 'rs\_package\_configuration.xml' des Transaktions-Formulars. In dieser Konfigurationsdatei sind folgende Aufgabenstellungen zu lösen:

1. Festlegen ob logische/physikalische Beileger im R-S für dieses Transaktionsformular genutzt werden können oder nicht
2. Festlegen ob optionale logische/physikalische Beileger einen postalischen Porto-Sprung für Dokumente bewirken darf

### Nutzung von logischen/physikalischen Beilegern

Im XML-Tag (Abschnitt) SUPPLEMENT befinden sich der XML-Tags LOGICALSUPPLEMENT.

Für das Unter-Tags LOGICALSUPPLEMENT kann das Unter-Tag ALLOWED mit den Werten 'true' oder 'false' angegeben werden. Der Wert 'false' bedeutet, das bei der Dokument-Bildung durch den ReportWriter für dieses Transaktions-Formular keine logischen Beileger berücksichtigt werden. Der Wert 'true' bedeutet das für dieses Transaktions-Formular logische Beileger (so definiert) berücksichtigt werden.

Für das Unter-Tag PHYSICALSUPPLEMENT kann das Unter-Tag ALLOWED mit den Werten 'true' oder 'false' angegeben werden. Der Wert 'false' bedeutet, das bei der Dokument-Bildung durch den ReportWriter für dieses Transaktions-Formular keine physikalischen Beileger berücksichtigt werden. Der Wert 'true' bedeutet das für dieses Transaktions-Formular physikalische Beileger (so definiert) berücksichtigt werden.

 Ist die Verwendung von logischen/physikalischen Beilegern in der Konfigurations-Datei des Transaktions-Formulars verboten, so kann es trotzdem möglich sein im R-S entsprechende Ressourcen zu definieren. Diese Einstellung wird erst durch den ReportWriter gelesen und berücksichtigt.

### Portosprünge durch optionale Beileger steuern

Im XML-Tag (Abschnitt) SHIPMENT befinden sich der XML-Tags POSTAGE.

Für das Unter-Tag POSTAGE kann das Unter-Tag OPTIONALSUPPLEMENTCANEXCEEDPOSTAGE mit den Werten 'true' oder 'false' angegeben werden. Der Wert 'false' bedeutet, das bei der Dokument-Bildung durch den ReportWriter optionale logischen/physikalische Beileger KEINEN Portosprung erzeugen und überzählige optionale logische/physikalische Beileger entfernt werden. Der Wert 'true' bedeutet das optionale logische/physikalische Beileger einen Porto-Sprung erzeugen dürfen.

## Konfiguration im R-S

### Konfiguration in der Version des Transaktions-Formulars

Ist die Konfiguration des Transaktions-Formulars korrekt eingerichtet, sieht die Information zur Beilegersteuerung im R-S für die Version des Transaktions-Formulars wie folgt aus:

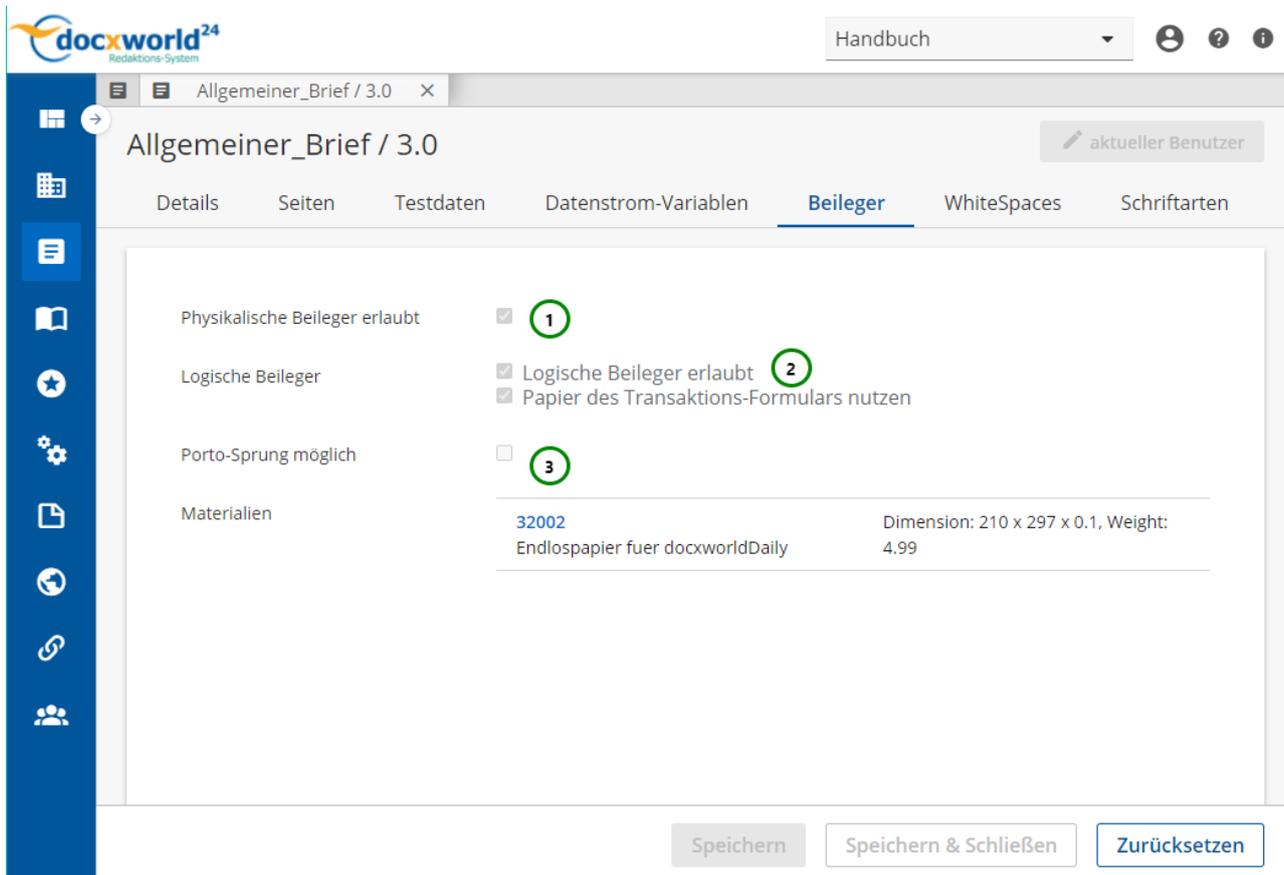


Abbildung A

Abb. A (1): Zeigt an, das die Verwendung von physikalischen Beilegern erlaubt ist

Abb. A (2): Zeigt an, das die Verwendung von logischen Beilegern erlaubt ist

Abb. A (3): Es soll verhindert werden das optionale Beileger einen Porto-Sprung verursachen

### Konfiguration in der Schema-Version

In der Schema-Version des Formulars sind nun drei logische Beileger definiert:

- LOG1 ist der logische Beileger 1 - als optional definiert
- LOG2 ist der logische Beileger 2 - als optional definiert
- LOG3 ist der logische Beileger 3 - als obligatorisch definiert

The screenshot shows the 'docxworld<sup>24</sup> Redaktions-System' interface. The main window title is 'Briefe / 3.0'. The 'Erweiterungen' (Extensions) tab is selected. On the left, a tree view shows the following structure:

- > DUMMY
- > ErsteSeite
- > Folgeseite
- > Physikalische Beileger
- ▼ Logische Beileger
  - LOG1
  - LOG2
  - LOG3
- > WhiteSpaces

The configuration for 'LOG1' is shown on the right:

- Name Erweiterungspunkt \*: LOG1
- Optionaler Beileger:
- Bedingung: true
- Ressource: R\_Beileger1 (with a 'x' icon and a 'Selektieren' button)

At the bottom, there are three buttons: 'Speichern', 'Speichern & Schließen', and 'Zurücksetzen'.

### Ergebnis aus der Vorschau zur Schema-Version

Im Testdatenstrom für die Druckvorschau sind zwei Dokumente enthalten, welche nun nach den Beilagen-Regeln aus dem Transaktions-Formular, den logischen Beilegern aus der Schema-Version und den Einstellungen nach Post-/Portobedingen aus der Konfiguration des ReportWriters aufbereitet werden. Das Ergebnis ist eine PDF-Datei mit folgendem Inhalt:

#### Dokument1

- Blatt 1 - Dokument-Seite aus dem Transaktion-Formular
- LOG3 - der obligate logische Beileger aus der Schema-Version
- LOG1 - der erste optionale logische Beileger aus der Tabelle der logischen Beileger aus der Schema-Version

==> LOG2 wird weggelassen, weil dieser einen Porto-Sprung von Standard-Pist auf Kompakt-Post erzwingen hätte --> dies ist aber nicht erlaubt

## Dokument2

- Blatt 1 - Dokument-Seite aus dem Transaktion-Formular
- LOG3 - der obligate logische Beileger aus der Schema-Version
- LOG1 - der erste optionale logische Beileger aus der Tabelle der logischen Beileger aus der Schema-Version

==> LOG2 wird weggelassen, weil dieser einen Porto-Sprung von Standard-Pist auf Kompakt-Post erzwungen hätte --> dies ist aber nicht erlaubt

[test\\_C1\\_PQPSP0\\_Q3.pdf](#)

## Beispiel - optionale Beileger

Die in dem Beitrag [Optionale Beileger und Porto](#) verwendeten Konfigurationen für den ReportWriter oder das Transaktionsformular sind in einem typischen Projekt für den profiforms Developer-Client zusammen gefasst. Dieses Projekt kann als ZIP-Datei hier herunter geladen werden:

Datei	Geändert
 <a href="#">PSP_Example_Doku.zip</a> Version ReportWriter 5.9	Dez. 08, 2015 by <b>Redakteur1</b>

 Um diese Konfigurationen komplett im Developer-Client ausprobieren zu können, muss noch eine Version des ReportWriters in das Verzeichnis 'bin' in dieses Projekt eingespielt werden.

 Wie im Haftungsausschluss beschrieben, gelten diese Beispiele zur Orientierung und als Muster zum Aufbau der gewünschten Logik. Diese Beispiele müssen auf die eigenen gewünschten Bedingungen angepasst werden, eine Prüfung der Funktionsfähigkeit des Beispiels ist nur mit dem ReportWriter 5.9.X und dem R-S 2.3.\* geprüft und sichergestellt.

## Algorithmen

In diesem Abschnitt werden ausgewählte Algorithmen des ReportWriters bei der Berechnung von Beilegern dargestellt. Diese Themen werden ständig nach praktischen Gesichtspunkten ergänzt.

### Gruppierung nach logischen und physikalischen Beilegern, sowie nach optionalen und nicht optionalen Beilegern

Grundsätzlich trennt der ReportWriter vor der Berechnung des Dokumentes in folgende Gruppen:

- optionale Beileger
  - Logische Beileger
  - Physikalische Beileger
- nicht optionale Beileger
  - Logische Beileger
  - Physikalische Beileger

Nicht optionale Beileger werden immer produziert. Optionale Beileger unterliegen in der Berechnung immer den postalischen Gesichtspunkten, wenn im Transaktions-Formular die Eigenschaft 'Optionale Beileger können einen Porto-Sprung bedingen' gesetzt ist.

### Eliminierung von gleichen optionalen logischen Beilegern

**Ausgangslage:** Es wird ein logischer Beileger 'RES\_BEILEGER' als Ressource in einer Schema-Version erzeugt. Dieser logische Beileger wird nun in den Erweiterungen zum Schema in zwei Erweiterungen für logische Beileger genutzt: 'LOG\_BEILEGER\_1' und 'LOG\_BEILEGER\_2' (zwei Erweiterungen zeigen auf den selben logischen Beileger als Ressource. Beide Erweiterung für die logischen Beileger sind als 'Optional' markiert. Es wäre für beide Beileger um Dokument noch 'Platz' (nicht wegen Portogrenzen entfernt).

**Ergebnis:** Der ReportWriter 'erkennt' bei der Kalkulation der logischen Beileger zum Dokument, das beide Beileger auf den selben Inhalt verweisen und entfernt eine Erweiterung zum logischen Beileger aus der Liste der logischen Beileger. Es wird also nur eine logische Beilage zum Dokument ausgegeben.

 Wichtig ist hier, das beide Erweiterungen die Eigenschaft 'Optional' für o.g. Beispiel haben müssen. Ist einer der Erweiterungen für Beileger als 'Optional' markiert und der andere als 'Nicht Optional', so werden in dem hier beschriebenen Fall BEIDE logische Beileger in das Dokument aufgenommen. Zwar zeigen beider Erweiterung auf den gleichen logischen Beileger als Ressource, aber die beiden Erweiterungen sind jeweils in einer anderen Beileger-Gruppe (siehe Abschnitt 'Gruppierung...').

# Glossar

[A](#), [B](#), [C](#), [D](#), [E](#), [F](#), [G](#), [H](#), [I](#), [J](#), [K](#), [L](#), [M](#), [N](#), [O](#), [P](#), [Q](#), [R](#), [S](#), [T](#), [U](#), [V](#), [W](#), [X](#), [Y](#), [Z](#)

---

## A

### Abmelden

- In der Top-Navigation können Sie sich aus dem Arbeitsbereich abmelden.

### Abwärts

- Listeneinträge lassen sich auf- und abwärts verschieben (wichtig, um die Hierarchie von Objekten festzulegen).

### Accepted-Status

- Zustand eines R-S Objekts. Bedeutet, dass das Objekt vom Benutzer angenommen wurde und nicht mehr bearbeitet werden kann.  
Nur im ACCEPTED-Status sind Objekte produktiv nutzbar.

### Admin

- Konfiguriert das System, siehe [R-S-Administrator](#) und [Systemadministrator](#).

### Admin-Rechte

- Als Administrator haben Sie Zugriff auf die Systemverwaltung des R-S und dessen Systemeinstellungen.

### Administrator-Client

- Kommandozeilen-Applikation, mit der Grundfunktionen des R-S administriert werden (siehe: [Administrator-Dokumentation](#)).

### Administrator-Client UI

- Grafische Oberfläche für den Administrator-Client, verfügbar seit Version 2.1.1000 des R-S. (siehe [R-S Admin-Client Dokumentation](#)).

### Administrator-Dokumentation

- Teil der Online-Dokumentation, der für [R-S](#)- und [Systemadministratoren](#) konzipiert ist und die Administrierung des R-S behandelt (siehe auch: [Nutzer-Dokumentation](#)).

### Aktionen

- Vom Benutzer durchgeführte Interaktion mit dem R-S oder vom R-S durchgeführte Aufgabe.

### Aktualisierung

- In dem Navigations-Pfad können Sie die aktuelle R-S Seite aktualisieren. Dadurch werden alle Änderungen auf den neuesten Stand gebracht. Verwenden Sie zur Aktualisierung der R-S Seite nicht den Aktualisierungs-Button Ihres Browsers, da das gesamte R-S sonst neu geladen wird.

### Anmelden

- Beim Start des R-S müssen Sie sich mit ihrem Benutzernamen und ihrem Kennwort anmelden und, falls Sie mehreren **Arbeitsbereichen** zugewiesen sind, einen Arbeitsbereich auswählen.

### Anzeigemodus

- Wenn Sie sich im Anzeigemodus befinden, können Sie die Eigenschaften eines R-S Objekts nicht verändern. R-S Objekte, die den **TEST**- oder **ACCEPTED**-Status haben, können nur im Anzeigemodus geöffnet werden.

### Arbeitsbereich

- Im Arbeitsbereich können Sie auf R-S Objekte zugreifen und mit ihnen arbeiten. Jeder Arbeitsbereich besitzt seine eigenen R-S Objekte und ist unabhängig von anderen Arbeitsbereichen.

### Arbeitsbereich wechseln

- In der Top-Navigation können Sie den Arbeitsbereich wechseln, vorausgesetzt, Sie sind anderen Arbeitsbereichen zugewiesen.

### Aufwärts

- Listeneinträge lassen sich auf- und abwärts verschieben (wichtig, um die Hierarchie von Objekten festzulegen).

### nach oben

---

## B

### Backup

- Eine komplette Sicherung von Daten. Es wird empfohlen, in regelmäßigen Abständen ein Backup durchzuführen (siehe: **Backup**).

### Bearbeitungsmodus

- Im Bearbeitungsmodus können Sie die Eigenschaften eines R-S Objekts verändern. R-S Objekte lassen sich nur dann im Bearbeitungsmodus öffnen, wenn sie den Status **Edit** haben.

### Beileger

- siehe *physikalischer Beileger* und *logischer Beileger*.

### Benutzertyp

- Es gibt 3 Arten von Benutzern: *Endbenutzer*, *R-S-Administratoren* und *System-Administratoren*.

### Berechtigungen

- Ein Nutzer hat nur Zugriff auf die Bereiche und Aktionen des R-S, für die er die notwendigen Berechtigungen besitzt.

### Bibliothek

- Eine Bibliothek ist eine Sammlung von Ressourcen, die Sie in Schema-Versionen einbinden können.

### Binär-Paket

- Binär-Pakete werden über den R-S Administrator-Client auf den R-S Server geladen. Ein Binär-Paket beinhaltet die notwendigen Programme, Skript und Konfigurationen die nötig sind, um aus Produktions-Paketen fertige Dokumente zu generieren. Binär-Pakete sind obligatorisch für die Dokument-Produktion. Sie werden für die Produktion zusammen mit den Produktions-Paketen aus dem R-S exportiert.

### Binäre Ressource (Ressource)

- Kann eine externe Datei speichern (z.B. Bilddateien), die dann als R-S Ressource genutzt werden kann.

### Boolean

- *Datentyp*: Kann 2 Werte annehmen: wahr oder falsch (1 oder 0, true oder false).

*nach oben*

---

## C

### CRM-System

- Steht für *Customer-Relationship-Management-System*. System zu Verwaltung von Kundenbeziehungen und Kundendaten.

*nach oben*

---

## D

### Date

- **Datentyp**: Enthält ein Datum.

### Datenstromvariable

- Vom **Transaktions-Formular** bereitgestellte Datenfelder, die Formulare spezifisch Informationen aus Datensystemen enthalten. Im R-S werden diese, aus Datensystemen migrierten Datenfelder als Datenstromvariable nutzbar. Der Wert der Datenstromvariable wird später z.B. durch dem Spooler mit dem passenden Wert aus dem Datensystem aufgefüllt.

### Datentyp

- siehe **Datentypen**.

### Debug-Modus

- Bezeichnung für den **Erweiterten Modus** bis Version 2.0.5000.

### Deprecated-Status

- Zustand eines R-S Objekts. Bedeutet, dass das Objekt angenommen wurde (ACCEPTED-Status), nun aber veraltet ist und nicht mehr benutzt werden sollte.

### Design-Zyklus

- Abschnitt des R-S Workflows, in dem ein R-S Objekt bearbeitet wird (EDIT-Status, etc.)

### Detail-Bereich

- Enthält die Übersichtstabellen, die die R-S Objekte enthalten. Im R-S wird mit diesen Objekten gearbeitet.

### Dimension

- Sie geben dem Material eine Größe: Breite, Höhe, Stärke.

### Docxworld

- Service zur Datenverarbeitung von profiforms, siehe **hier**.

### Docxworld-Link

- Verlinkungen auf ein Produktions-Paket, die eine Schnittstelle für externe Programme bereitstellen, die durch die Verlinkung Produktions-Pakete im R-S finden und im weiteren Produktions-Prozess nutzen können. Kann nur genutzt werden, wenn docxworld für den Druckprozess benutzt wird.

Ansonsten müssen zur Bereitstellung der Pakete **Produktions-Paket Verknüpfungen** benutzt werden.

*nach oben*

---

## E

### **Edit-Status**

- Status eines R-S Objekts. Bedeutet, dass sich das Objekt im Bearbeitungszustand befindet.

### **Eingabe-Feld-Validierung**

- Bei einer Benutzereingabe wird überprüft, ob der eingegebene Wert mit dem erwarteten Datentyp konform ist.

### **Eingabe-Hilfe**

- Zeigt in Eingabe-Feldern, die EVAL-Anweisungen erlauben, einen Eingabe-Hilfe Dialog an (<CTRL><SHIFT><SPACE>).

### **Endbenutzer**

- Bedient das R-S, hat aber keine Admin-Rechte.

### **Erforderliche Ressource**

- Von einer Bibliothek erwartete Ressource, die beim Einbinden der Bibliothek in ein Schema vorhanden sein muss.

### **Error-Status**

- Status eines R-S Objekts. Bedeutet, dass bei dem Objekt ein Fehler aufgetreten ist.

### **Erstellungsmodus**

- Wenn Sie ein R-S Objekt neu erstellen befinden Sie sich automatisch im Erstellungsmodus. Unter Umständen können Sie im Erstellungsmodus bestimmte Attribute noch nicht einsehen oder ändern.

### **Erweiterter Modus**

- Im Erweiterten Modus werden Ihnen die internen ID's der R-S Objekte direkt im R-S Client / R-S Admin-Client angezeigt (ehemals **Debug-Modus**).

### **Erweiterungs-Punkt**

- Vordefinierter Bereich im Formular, in den per Erweiterung Ressourcen eingefügt werden können.

## Erweiterungen

- Durch Erweiterungen können Ressourcen und Materialien in ein Schema eingebunden und in das Dokument eingefügt werden.

## EVAL-Anweisung

- Evaluierungs-Befehl. Ermöglicht dynamische Einbindung von Ressourcen, dynamische Berechnungen und die Verwendung vorgefertigter Funktionen. Eine EVAL-Anweisung hat die Grundform `#{BEFEHL}`.

## Export

- Durch Exportieren von Objekten werden diese so gespeichert, dass Sie sie jederzeit wieder importieren und mit Ihnen weiterarbeiten können.

[nach oben](#)

---

# F

## Falz-Typ

- Gibt an, ob der Beileger schon gefalzt vorliegt und wenn nicht, ob er gefalzt werden darf und wenn ja, in welcher Weise.

## Formular-Version

- siehe [Transaktions-Formular-Version](#).

## Formularname

- Name eines Formulars, der in der Konfiguration des Formular-Datenstroms festgelegt ist (`<name>FORMULARNAME</name>`). Über diesen Formularnamen werden z.B. Produktions-Paket Verknüpfungen definiert.

## Funktion

- Führt Befehle aus und gibt einen Wert zurück. Funktionen können innerhalb von **EVAL-Anweisungen** angegeben werden. Es gibt eine Reihe vorgefertigter Funktionen, die Sie benutzen können (siehe [Funktionen in EVAL-Anweisungen](#)).

[nach oben](#)

---

## G

### Gewicht

- Das Gewicht eines Beilegers muss im Material angegeben werden. Die Eingabe erfolgt in Gramm (g).

*nach oben*

---

## H

### Hilfe

- Die umfassendste Hilfestellung bietet die **Online-Dokumentation**. Außerdem stehen Ihnen noch die **Eingabe-Hilfe** und die **kontextbezogene Hilfe** zur Verfügung.

### Hinweis

- Ist ein Symbol in der Online-Dokumentation (siehe **Verwendete Symbole**).

### Historie

- Zeigt Änderungen in chronologischer Reihenfolge an.

*nach oben*

---

## I

### ID

- Ein einem Objekt eindeutig zugewiesener Identifikator. Das R-S arbeitet intern mit ID's, um Objekte anzusprechen. Per **Erweitertem Modus** können Sie sich die internen ID's anzeigen lassen.

### Import

- Durch Importieren von Dateien können Sie mit zuvor **exportierten** oder in bestimmten Formaten abgelegten Dateien weiterarbeiten. Auch ist es möglich, R-S Objekte zu exportieren und in einem anderen R-S Arbeitsbereich wieder zu importieren.

### Import-Paket

- Enthält die nötigen Daten zum Aufbau eines Transaktions-Formulars im ZIP-Format. Import-Pakete werden z.B. durch den DeveloperClient generiert und enthalten die Rohdokumente, die im R-S gestaltet werden sollen.

### Import-Paket-Status

- Status eines Import-Pakets, z.B. *ANALYZED* (Paket wurde vollständig analysiert und liegt korrekt vor) oder *ERROR* (Beim Import ist ein Fehler aufgetreten).

### Info

- Ist ein Symbol in der Online-Dokumentation (siehe [Verwendete Symbole](#)).

### Inline-Sequenz

- Anweisungen zur Textformatierung (z.B. \b für **bold** oder \cnblue. für [blauen text](#)). Die Interpretation der Inline-Sequenz erfolgt dabei erst bei Ausgabe der Dokumente (z.B. bei der [Vorschau](#)).

### Integer

- Datentyp: Enthält eine ganze Zahl, z.B. 600 oder -412. Der Wertebereich von Integer ist kleiner als der von [Number](#).

[nach oben](#)

---

## J

[nach oben](#)

---

## K

### Kontextbezogene Hilfe

- Unterstützt Sie beim Eingeben von [EVAL-Anweisungen](#), indem es Ihnen eine Auswahl gültiger Eingaben anbietet.

[nach oben](#)

---

## L

### Layout

- Gestaltung, Anordnung und Aufbau einer Seite.

### Logische Beileger

- Beileger, die während des Produktionsprozesses direkt mit den Dokumenten mit gedruckt werden.

[nach oben](#)

---

## M

### Material

- Materialien beschreiben die physischen Eigenschaften von Belegern. Bei jedem Beileger muss ein zuvor erstelltes Material angegeben werden, das Größe, Gewicht und andere Eigenschaften des Belegers beschreibt. Dies ist wichtig für den Druckprozess und den Versand.

[nach oben](#)

---

## N

### Navigations-Pfad

- Navigationsleite, die den Pfad zur momentan geöffneten R-S-Seite graphisch darstellt und rückwärtsgerichtetes Navigieren ermöglicht.

### Number

- **Datentyp**: Enthält eine ganze Zahl, z.B. 518 oder -1620. Der Wertebereich von Number ist größer als der von **Integer**.

### Nutzer

- Benutzer im R-S mit Benutzertyp, Profil und ihm zugewiesenen Rechten.

### Nutzer-Dokumentation

- Teil der Online-Dokumentation, der für **Nutzer** des R-S konzipiert ist und beschreibt, wie man mit dem R-S arbeitet (*siehe auch: **Administrator-Dokumentation***).

### Nutzer-Gruppe

- Eine Nutzer-Gruppe besteht aus einem oder mehreren Nutzer(n). Der Nutzer-Gruppe können Rechte zugewiesen werden.

*nach oben*

---

## O

### OMS

- Output-Management Suite, Produktbundle von profiforms. Beinhaltet Software-Lösungen für Output-Management.

*nach oben*

---

## P

### PDF-Datei

- Dateiformat zur Darstellung von Dokumenten. Vorschauen werden im PDF-Format erzeugt. Zum Öffnen benötigen Sie den Adobe Acrobat Reader.

### Pflichtfeld

- In Pflichtfeldern muss etwas eingegeben werden, bevor Sie das Objekt speichern.

### Physikalische Beileger

- Beileger, die schon vorgefertigt vorliegen und den Dokumenten während des Produktionsprozesses beigefügt werden.

### Produktions-Paket

- Für die Produktion generiertes Paket, das **Formular**, **Schema** und alle nötigen Inhalte für die Weiterverarbeitung zusammenfasst. Produktions-Pakete sind der letzte Schritt in der R-S Dokumentgenerierung.

### Produktions-Paket-Verknüpfung

- Verlinkungen auf ein Produktions-Paket, die eine Schnittstelle für externe Programme bereitstellen, die durch die Verlinkung Produktions-Pakete im R-S finden und im weiteren Produktions-Prozess nutzen können. Sollte nur genutzt werden, wenn Sie kein docxworld benutzen. Ansonsten müssen zur Bereitstellung der Pakete **docxworld-Links** benutzt werden.

## Produktions-Zyklus

- Abschnitt des R-S Workflows, in dem ein R-S Objekt produktiv gesetzt wird (**ACCEPTED-**, **PRODUCTION-Status**, etc.)  
Sie können bei Produktions-Paketen außerdem Anmerkungen, die während des Produktion-Zyklus aufkamen, abspeichern.

## Profil

- In der Top-Navigation können Sie Ihr Profil aufrufen. Außerdem können Sie dort ihr Passwort ändern.

[nach oben](#)

---

## Q

[nach oben](#)

---

## R

### Real

- **Datentyp**: Enthält eine Gleitkomma-Zahl (z.B. 1.5 oder -7.31)

### Rechte

- Ein Nutzer kann nur die Aktionen ausführen, für die er ausreichende Rechte besitzt. Rechte werden z.B. in der **Rechteverwaltung** verteilt.

### Regel (Ressource)

- Kontrollstruktur, die es Ihnen ermöglicht, abhängig von Bedingungen unterschiedliche Werte zurückzugeben.  
Regeln ermöglichen außerdem, Ressourcen abhängig von Bedingungen einzubinden.

### Register

- Ein Register teilt den Inhalt eines R-S Objekts in thematisch geordnete Untergruppen (Registerkarten) auf.

### Registerkarten

- Registerkarten enthalten jeweils verschiedene Eigenschaften eines R-S Objekts.

### Rejected-Status

- Status eines R-S Objekts. Bedeutet, dass das Objekt zurückgewiesen und nicht **angenommen** wurde.

### Ressource

- Ressourcen können in das vom Transaktions-Formular zur Verfügung gestellte Dokument eingebunden werden.  
Typen von Ressourcen sind z.B. **binäre Ressourcen** (z.B. Bilder), **Variablen** (z.B. Text) oder **Beileger**.

### RichText (Ressource)

- Formatierbarer Text (implementiert nach **W3C XFA-Spezifikation**).

### R-S-Administrator

- Konfiguriert die Einstellungen des R-S und stellt es auf die jeweilige Benutzerumgebung ein.

### R-S Objekt

- R-S Objekte sind die Objekte, die sie im **Detail-Bereich** sehen.

### R-S Server Konfigurationsdatei

- Über sie lassen sich wichtige Einstellungen des R-S konfigurieren.

*nach oben*

---

## S

### Schema

- In einem Schema wird die Darstellung und Gestaltung eines Transaktions-Formulars implementiert.

### Schema-Version

- Nutzbare Version eines Schemas, siehe **Versionierung im R-S**.

### Schriftart

- In Ressourcen dürfen nur Schriftarten verwendet werden, die vom Formular unterstützt werden. Für Bibliotheken müssen explizit Schriftarten **erstellt** werden, da diese nicht Formularegebunden sind.

### Sitemap

- Bietet eine hierarchische Übersicht der gesamten Online-Dokumentation.

### Status

- **Transaktionsformulare, Schemas, Produktions-Pakete** und **Bibliotheken** besitzen einen eigenen Status im R-S Workflow. Dieser Status gibt den Zustand des Objekts an und bestimmt, wie das Objekt verwendet werden darf.

### Stil (Font)

- Ausprägungen für eine (Regular, Fett, Kursiv, Fett kursiv). Welche Ausprägungen für eine Schriftart verwendet werden dürfen ist für jede Schriftart festgelegt.

### String

- Enthält eine Zeichenkette.

### System-Administrator

- Konfiguriert die Einstellungen des Systems, auf dem das R-S installiert wird.

### System-Verwaltung

- Zeigt Details über den aktuellen **Nutzer** und ihr R-S an und ermöglicht die Aktivierung des **Debug-Modus**.

[nach oben](#)

---

## T

### Testdaten

- In das ZIP-Binär-Paket Ihres Transaktionsformulars können Sie Testdaten einbinden, um eine exemplarische Druckvorschau zu ermöglichen. Testdaten können auch durch den Nutzer im Schema hochgeladen werden.

### Testproduktion

- Sie können Produktions-Pakete für eine Testproduktion verfügbar machen. Dazu muss das Produktions-Paket nicht ACCEPTED-Status haben. Eventuell müssen Sie Testproduktionen (v.a. in Docxworld) vorher freischalten lassen. Sie benötigen in Docxworld außerdem einen Testvertrag.

### Test-Status

- Status eines R-S Objekts. Bedeutet, dass sich das Objekt im Testzustand befindet.

### Test-Zyklus

- Abschnitt des R-S Workflows, in dem ein R-S Objekt getestet wird (TEST-Status). Sie können bei Produktions-Paketen außerdem Anmerkungen, die während des Test-Zyklus aufkamen, abspeichern.

### **Tipp**

- Ist ein Symbol in der Online-Dokumentation (siehe [Verwendete Symbole](#)).

### **Top-Navigation**

- Bietet grundlegende Elemente zur Steuerung ihrer R-S Applikation.

### **Transaktions-Formular**

- Rohdokument, die das Layout, den Grundaufbau und weitere Daten eines Dokumentes durch ein Import-Paket aus dem Formular-Designer (z.B. DeveloperClient) vorgibt.

### **Transaktions-Formular-Version**

- Nutzbare Version eines Transaktions-Formulars, siehe [Versionierung im R-S](#).

### **Tutorial**

- Tutorials sind Lernhilfen zum R-S oder zu einzelnen Funktionen des R-S. Zum R-S gibt es momentan noch keine Video-Tutorials.

[nach oben](#)

---

## **U**

### **Überschreiben von Ressourcen**

- Wenn Sie das Überschreiben von Ressourcen aktivieren, erlauben Sie es Ressourcen innerhalb eines Schemas, gleichnamige Ressourcen mit geringerer Priorität zu überschreiben (Kommt beim Einbinden von Bibliotheken zum tragen).

[nach oben](#)

---

## **V**

### **Variable (Ressource)**

- Enthält einen beliebigen, benutzerdefinierten Wert (z.B. 5 oder "Hallo").

### **Version**

- **Schemas, Formulare, Bibliotheken** und **Produktions-Pakete** verlangen das Erstellen verschiedener Versionen, wobei die Objekte als Vorlagen für die verschiedenen Version dienen.

### Vertrag

- Vertrag, docxworld: Wenn Sie docxworld-Kunde sind, können Sie die fertigen Produktions-Pakete mit docxworld-Links für den weiteren Produktions-Prozess verfügbar machen (ansonsten benötigen Sie **Produktions-Paket Verknüpfungen**)

### Vorschau

- Bezeichnet eine Dokumentenvorschau, die mit Hilfe von Testdaten generiert wird.

*nach oben*

---

## W

### Warnung

- Ist ein Symbol in der Online-Dokumentation (siehe **Verwendete Symbole**).

### Workflow

- Beschreibt den Arbeits- und Statusablauf innerhalb des R-S (siehe **Workflow**).

### WhiteSpace

- Ermöglicht es, Inhalte (Ressourcen) dynamisch auf freien Bereich in Dokumenten zu platzieren.

*nach oben*

---

## X

*nach oben*

---

## Y

*nach oben*

---

## Z

### Zugriffs-Pfad

- In einem Zugriffs-Pfad definieren Sie, welche Berechtigungen ein Nutzer für welches R-S Objekt besitzt.

### Zugriffs-Typ

- Sie können festlegen, ob ein Nutzer Zugriff auf die im Zugriffs-Pfad definierten R-S Objekte und Aktionen hat ("grant") oder nicht ("deny").

*nach oben*

---

## Hinweise zur Online-Dokumentation

Teile dieser Online-Dokumentation können die Nutzer und Administratoren des R-S für Nutzer zur eigenen Verwendung ausdrucken, es gelten die **rechtlichen Hinweise**. Die verwendeten Symbole, Schriftarten und deren Bedeutung werden auf dieser Seite erläutert: **Verwendete Symbole**. Abbildungen des R-S, Grafiken und Diagramme wurden teilweise aus Platzgründen in ihrer Größe bearbeitet.

Für eine optimale Darstellung der Online-Dokumentation sollte Ihre Bildschirmauflösung 1680x1050 Pixel oder höher betragen, mindestens jedoch 1440x900. Falls Sie weitergehende Fragen haben, die in der Dokumentation nicht thematisiert werden finden Sie **hier** eine Liste ergänzender Dokumentationen. Ansonsten wenden Sie sich bitte an unseren **Support**.

## Verwendete Symbole

In der Nutzer-Dokumentation gibt es diverse Symbole und Zeichen.

Um Ihnen einen Überblick zu verschaffen, sind hier die wichtigsten Symbole erklärt.

Symbol	Beschreibung
	Dieses Symbol gibt zusätzliche Informationen zu einem Textabschnitt.
	Dieses Symbol gibt Informationen, die unbedingt beachtet werden müssen.
	Dieses Symbol gibt Hinweise, die beachtet werden sollten.
	Dieses Symbol enthält ein Beispiel für einen Textabschnitt.
	Verfügbare Aktionen werden durch diese Hintergründe gekennzeichnet.
	Dieses Symbol beinhaltet vor allem Inhaltsverzeichnisse, Auflistungen und weiterführende Links.
<b>Klicken Sie hier, um den Text auszuklappen</b> Hallo!	Beinhaltet Informationen, die Sie durch einen Klick auf die Überschrift ausklappen können.
<i><a href="#">R-S Dokumentation</a></i>	Verlinkung auf eine andere Seite der R-S Dokumentation.
	Symbole entsprechen den Symbolen 1 - 4, können aber auch als Symbole in Tabellen verwendet werden.
	Diese Symbole kennzeichnen Vor- und Nachteile oder geben zusätzliche Funktionen oder Merkmale bzw. Warnungen an.
	Zwingende Angabe / Pflichtfeld / Zwingende Aktion.

Symbol	Beschreibung
	Nichteingabe erzeugt zwar keinen Fehler, ist aber notwendig für die korrekte Arbeitsweise des R-S.
	Optionale Angabe.
	Das Symbol im R-S, das Pflichtfelder anzeigt.
	Wenn dieses Symbol in den Screenshots auftaucht, gibt es unter der Abbildung Erklärungen zu der/den Nummer/n.
	Dieses Symbol fasst mehrere Elemente in einer Abbildung zusammen.
	Dieses Symbol hebt Bildausschnitte hervor.
<i>Abb. A (1)</i>	Gibt im Text an, dass sich eine Erklärung auf eine Abbildung mit dem jeweiligen Buchstaben und der jeweiligen Nummer bezieht. Enthält eine Seite nur eine Abbildung, kann statt <i>Abb. A (1)</i> auch nur <i>(1)</i> stehen. Enthält eine Seite mehrere Abbildungen, beziehen sich Angaben ohne explizite Abbildungsangabe wie <i>(1)</i> immer auf Abbildung A.
<STRG><SPACE>	Tastenkombination.
"Auswählen"	Das zwischen den " " stehende bezieht sich auf einen Ausdruck in einer Abbildung.
STRING, OBJEKTNAME	Kennzeichnet Datentypen, Ressourcen oder R-S Objekte.

## Weiterführende Informationen

Die weiterführenden Informationen gliedern sich in folgende Unterkapitel (bitte auf den jeweiligen Link klicken):

- [\*Sitemap\*](#)
- [\*Download der Dokumentation\*](#)
- [\*Online-Archiv dieses Produkts\*](#)
- [\*Ergänzende Dokumentationen\*](#)
- [\*Service und Support\*](#)

## Sitemap

### *Rechtliche Hinweise*

### *Was ist neu in Version 2.5*

### *Einführung*

- *Umgebung des R-S*
- *Die Benutzertypen*

### *Nutzer-Dokumentation*

- *Grundlagen*
  - *Status - Workflows und Versionierung*
    - *Status einer Objekt-Version setzen*
  - *Arbeitsbereiche*
  - *Ressource-Typen*
    - *Variablen*
    - *RichText*
    - *Binäre Ressourcen*
    - *Regeln*
    - *Physikalische Beileger*
    - *Logische Beileger*
    - *WhiteSpaces*
    - *Datenstrom-Variablen*
  - *Basisobjekte im R-S*
    - *Transaktions-Formulare*
    - *Schemas*
    - *Produktions-Pakete*
    - *Bibliotheken*
    - *Material*
      - *Material für logische Beileger*
      - *Material für physikalische Beileger*
    - *Docxworld-Links*
    - *Produktions-Paket Verknüpfungen*
  - *Erweiterungen*
- *Erste Schritte*
  - *Anmeldung am R-S Client*
  - *Nutzeroberfläche und Navigation*
    - *Nutzeroberfläche des R-S*
      - *Bereich - Header*
      - *Bereich - Seitennavigation*
      - *Bereich - Detail*
        - *Suchfunktion*
        - *Register*
      - *Bereich - Tab*
    - *Bedienung des R-S*
      - *Pflicht-Eingabefelder*
      - *Sichtbarkeit und Berechtigungen*
      - *Hilfe-System*
      - *Datentypen und Eingabe-Feld-Validierung*
- *Dashboard*
- *Arbeiten mit Transaktions-Formularen*

- **Formular erstellen, ändern und löschen**
  - **Formular-Version**
    - **Formular-Version erstellen, ändern und löschen**
    - **Formular-Version kopieren**
    - **Status einer Formular-Version setzen**
    - **Registerkarten bei Formular-Versionen**
      - **Registerkarte - Details in Formular-Versionen**
        - **Import eines Formulars**
          - **Adobe XDP und OMS - Import-Paket**
            - **Aufbau des ZIP-Import-Paketes**
            - **Aufbau der Paket-Beschreibungs-Datei**
            - **Layout-Bereiche im Adobe XDP**
            - **Produkte - Konfiguration**
      - **Registerkarte - Seiten**
      - **Registerkarte - Testdaten**
      - **Registerkarte - Datenstrom-Variablen in Formular-Versionen**
      - **Registerkarte - Beileger**
      - **Registerkarte - WhiteSpace**
      - **Registerkarte - Schriftarten**
      - **Registerkarte - Umgebungen**
- **Arbeiten mit Schemas**
  - **Schema erstellen, ändern und löschen**
  - **Schema-Version**
    - **Schema-Version erstellen, ändern und löschen**
    - **Schema-Version kopieren**
    - **Status einer Schema-Version setzen**
    - **Schema-Version exportieren**
    - **Schema-Version importieren**
    - **Registerkarten bei Schema-Versionen**
      - **Registerkarte - Details in Schema-Versionen**
      - **Registerkarte - Bibliotheken**
      - **Registerkarte - Ressourcen**
      - **Registerkarte - Erweiterungen**
        - **Seitendesign-Ansicht**
        - **Erweiterung für Seiten verknüpfen, ändern und löschen**
        - **Erweiterung für physikalische Beileger erstellen, ändern und löschen**
        - **Erweiterung für logische Beileger erstellen, ändern und löschen**
        - **Erweiterung für WhiteSpaces erstellen, ändern und löschen**
        - **Konflikte und Markierungen bei Erweiterungen**
      - **Registerkarte - Vorschau in Schema-Versionen**
      - **Registerkarte - Datenstrom-Variablen in Schema-Versionen**
- **Arbeiten mit Bibliotheken**
  - **Bibliotheken erstellen, ändern und löschen**
  - **Bibliotheks-Version**
    - **Bibliotheks-Version erstellen, ändern und löschen**
    - **Bibliotheks-Version kopieren**
    - **Status einer Bibliotheks-Version setzen**
    - **Bibliotheks-Version exportieren**
    - **Bibliotheks-Version importieren**
    - **Bibliotheks-Ressourcen**
- **Ressourcenverwaltung**
  - **Ressourcen erstellen und ändern**
    - **Binäre Ressource erstellen und ändern**
    - **Variable erstellen und ändern**
    - **Regel erstellen und ändern**
      - **Spalte einfügen, bearbeiten und löschen**
      - **Zeile hinzufügen, bearbeiten, löschen und verschieben**
  - **Physikalischen Beileger erstellen und ändern**

- **Logischen Beileger erstellen und ändern**
  - **Design des logischen Beilegers**
- **WhiteSpace erstellen und ändern**
- **RichText erstellen und ändern**
- **Erforderliche Ressource erstellen und ändern**
- **Ressourcen löschen**
- **Ressourcen duplizieren**
- **Ressourcen zurücksetzen**
- **Ressourcen exportieren**
- **Ressourcen importieren**
- **Ressourcen Zustände**
- **Arbeiten mit Produktions-Paketen**
  - **Paket ändern**
    - **Registerkarte - Details in Produktions-Pakete**
    - **Registerkarte - Historie**
    - **Registerkarte - Vorschau in Produktions-Pakete**
    - **Registerkarte - Test-Zyklus**
    - **Registerkarte - Produktions-Zyklus**
    - **Registerkarte - Paket-Prüfung**
  - **Status eines Pakets setzen**
- **Arbeiten mit Materialien**
  - **Material für physikalischen Beileger erstellen, ändern und löschen**
  - **Material für logischen Beileger erstellen, ändern und löschen**
- **Arbeiten mit Docxworld-Links**
  - **Docxworld-Links erstellen, ändern und löschen**
- **Arbeiten mit Produktions-Paket Verknüpfungen**
  - **Verknüpfung erstellen und ändern**
- **Mehr zur Arbeit mit Objekten**
  - **Löschen von Objekten**
  - **Wechseln des Modus**
  - **Eval-Anweisungen, Operatoren und Datentypen**
    - **Funktionen in Eval-Anweisungen**
      - **AddDate**
      - **AddI**
      - **AddR**
      - **And**
      - **ClearS**
      - **ConcatS**
      - **Dividel**
      - **DivideR**
      - **EmptyS**
      - **FillS**
      - **If**
      - **IndexS**
      - **IsEqualI**
      - **IsEqualR**
      - **IsEqualS**
      - **IsGreaterI**
      - **IsGreaterR**
      - **IsGreaterS**
      - **IsLessI**
      - **IsLessR**
      - **IsLessS**
      - **IsNullI**
      - **IsNullR**
      - **IsNullS**
      - **LengthS**
      - **MultiplyI**

- *MultiplyR*
- *Nand*
- *Nor*
- *Not*
- *NowS*
- *Or*
- *PlaceholderS*
- *RemoveAllS*
- *ReplaceAllS*
- *RoundR*
- *SepSign*
- *SubS*
- *SubtDate*
- *Subtracl*
- *SubtracR*
- *TokenS*
- *ToLowerS*
- *ToUpperS*
- *TrimS*
- *INLINE-Sequenzen*
  - *b*
  - *cn*
  - *dn*
  - *f*
  - *fn*
  - *fs*
  - *i*
  - *n*
  - *normal*
  - *ol*
  - *plain*
  - *sl*
  - *t*
  - *u+*
  - *ul*
  - *up*

## **Administrator-Dokumentation**

- **Administrator-Client UI**
  - **Anmeldung am R-S Admin-Client**
  - **Nutzeroberfläche des RS-Admin-Clients**
  - **Funktionen des Admin-Clients**
    - **Navigation - Nutzer**
      - **Nutzer erstellen, ändern und Rechte vergeben**
    - **Navigation - Arbeitsbereiche**
      - **Arbeitsbereich erstellen und ändern**
      - **Grafische Objektübersicht**
    - **Schriftarten**
    - **Binärpakete**
- **Administrator-Client Konsole**
- **Installation**
  - **System-Voraussetzungen**
    - **R-S Client**
    - **R-S Server**
  - **Datenbanken**
    - **Datenbank - Derby**
    - **Datenbank - MySQL**

- **Systemsicherheit**
- **Verwendete Software**
- **System-Berechtigungen**
  - **Arbeitsbereich-Administrator**
  - **Nutzer, Nutzer-Gruppen und Rechte**
    - **Nutzer verwalten**
    - **Rechte**
      - **Berechtigungen**
    - **Gruppen**
  - **System-Nutzer**
- **Systembetrieb**
  - **Backup**
  - **Binär-Pakete**
    - **Aufbau des ZIP-Binär-Paketes**
    - **Aufbau der Binär-Paket Beschreibungs-Datei**
    - **Programme in Binär-Paketen**
      - **ReportWriter**
        - **Funktion - GetTranspromoResource**
  - **HTTP-Status-Codes durch Client- oder Server-Fehler im WEB-Browser**
  - **Restore**
  - **R-S Server Konfigurationsdatei**

## **Anleitungen**

- **Logische Beileger des R-S über SupplementWorkitem des ReportWriters**
- **Optionale Beileger und Porto**
  - **TCI des ReportWriters**
  - **Konfigurationen Transaktions-Formulars**
  - **Konfiguration im R-S**
  - **Beispiel - optionale Beileger**
  - **Algorithmen**

## **Glossar**

### **Hinweise zur Online-Dokumentation**

- **Verwendete Symbole**

### **Weiterführende Informationen**

- **Sitemap**
- **Download der Dokumentation**
- **Online-Archiv dieses Produkts**
- **Ergänzende Dokumentationen**
- **Service und Support**

## Download der Dokumentation

Format	Erstellungsdatum	Größe	Download-Datei
PDF	 16.09.2022	<15 MB	<b>PDF-Datei</b>



Bitte beachten Sie unsere **rechtlichen Hinweise**, bevor Sie die Dateien herunterladen!  
 Die Dateien entsprechen dem Datum in der Spalte "Erstellungsdatum". Die Online-Dokumentation ist nur zu diesem Zeitpunkt der Erstellung aktuell.  
 Wir weisen Sie darauf hin, dass interaktive Multimedia-Inhalte in der Online-Dokumentation im PDF-Format nicht angezeigt werden können.

## Online-Archiv dieses Produkts

Online Archiv
<input type="checkbox"/> <i>Redaktions-System 2.4</i>

 Gerne unterstützen wir Sie auch mit den älteren Versionen des R-S. Beachten Sie jedoch, dass nur die beiden Vorgänger der aktuellen Version verwendet werden dürfen. Bitte beachten Sie auch, dass die Dokumentationen älterer Versionen eventuell unvollständig sind und nicht aktualisiert werden.

## Ergänzende Dokumentationen

Hier gelangen Sie zu anderen Online-Dokumentationen der Firma profiforms gmbh. Bitte beachten Sie, dass der jeweilige Link eine neue Seite öffnet und Sie die aktuelle Dokumentation verlassen.

Grundsätzlich ist das R-S dafür vorgesehen die Dokumente in der Druckvorschau und in der Dokumentproduktion durch verschiedene Output-Management-Systeme produzieren zu können. In den aktuellen Projekten (unter anderem auch in docxworld<sup>24</sup>) ist das R-S aber speziell auf die Outputproduktlinie der Firma profiforms abgestimmt. Kunden nutzen zum Beispiel den ReportWriter um Dokumente auf der Basis von Rohdaten (XML, RDI, CSV ...) oder nutzen Logistics um PDF-basierte Dokumente weiter zu verarbeiten.

### ReportWriter

[ReportWriter 5.9](#)

### Logistics

[Logistics 5.9](#)

### Spooler

[Spooler 3.9](#)

### docxworld

[docxworld 5.8](#)

## Service und Support

Sie haben Fragen oder Probleme zu/mit einem unserer Produkte und verfügen über einen gültigen Support- und Update-Vertrag?

Dann kontaktieren Sie uns bitte:

- ... über unsere Webseite: <http://www.profiforms.de>
- ... über den Ihnen bekannten/zugeordneten Projekt-/Vertriebs-Mitarbeiter
- ... über unseren Service Desk: <https://support.profiforms.de/servicedesk/customer/portals>