

# Konnektoren

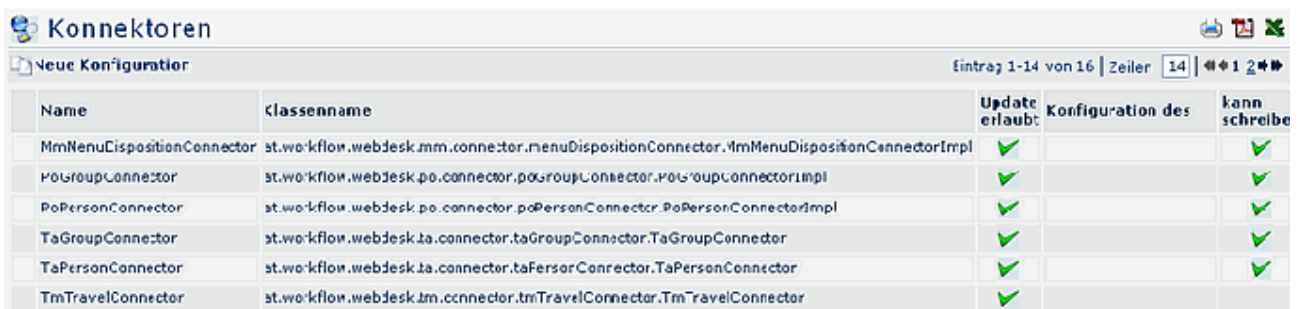
## Allgemeines zu Konnektoren

Die Konnektoren stellen ein Rahmengerüst für den **Import** und **Export** von **Daten** in den Webdesk (bzw. aus dem Webdesk) zur Verfügung.

Die Definition von Quellkonnektoren (Quelle, aus der die Daten kommen) und Zielkonnektoren (wo die Daten hinkommen) erlaubt die Gestaltung des Datenflusses. So können nicht nur Daten aus dem Zeitwirtschaftssystem in den Webdesk synchronisiert werden, sondern auch Datenbanken angesteuert, Textfiles (CSV-Dateien) erstellt werden. So können auch Daten aus dem Webdesk exportiert, und in andere Systeme importiert und auf dem Weg dorthin mittels Scripting angepasst werden (Schnittstellen-Framework).

Prinzipiell gibt es folgende Arten von Konnektoren:

- Lesende Konnektoren
- Schreibende Konnektoren
- Konnektoren die sowohl lesen als auch schreiben können



Name	Klassenname	Update erlaubt	Konfiguration des	kann schreiben
MmMenuDispositionConnector	st.workflow.webdesk.mm.connector.menuDispositionConnector.MmMenuDispositionConnectorImpl	✓		✓
PosGroupConnector	st.workflow.webdesk.po.connector.posGroupConnector.PosGroupConnectorImpl	✓		✓
PoPersonConnector	st.workflow.webdesk.po.connector.poPersonConnector.PoPersonConnectorImpl	✓		✓
TaGroupConnector	st.workflow.webdesk.ta.connector.taGroupConnector.TaGroupConnector	✓		✓
TaPersonConnector	st.workflow.webdesk.ta.connector.taPersonConnector.TaPersonConnector	✓		✓
TmTravelConnector	st.workflow.webdesk.tm.connector.tmTravelConnector.TmTravelConnector	✓		

- **Name**  
Name des Konnektors
- **Klassenname**  
gibt den technischen Klassennamen des Konnektors an
- **Update erlaubt**  
Ziel (oder rechte Seite) des Konnektors
- **Konfigurations des**  
Handelt es sich bei dem Konnektor um eine Konfiguration, so wird hier der urspr. Konnektor angezeigt, von dem die Konfiguration abstammt
- **Kann schreiben**  
definiert ob der Konnektor auch schreibend verwendet werden kann. Konnektoren die diese Option nicht verwenden, können bei den verknüpften Konnektoren nicht als Zielconnectoren verwendet werden.

Weiters können im Webdesk auch **verknüpfte Konnektoren** definiert werden. Sie stellen eine Verknüpfung zwischen Konnektoren dar, und ermöglichen so die Definition des Datenflusses (Definition als Ziel- oder Quellkonnektoren).

## Anlegen eines neuen Konnektors / Neue Konfiguration

Um einen neuen Konnektor anzulegen klicken sie auf die Schaltfläche **Neue Konfiguration** > Befüllung folgender Felder:

## Konfiguration von Konnektoren

Speichern 
 Speichern & Schließen 
 Zurück 
 Löschen

Vaterkonnektor:

Name:  \*

Connector kann schreiben:

- **Vaterkonnektor**  
 Selektion des Konnektors, von welchem sich die neue Konfiguration ableiten soll. Zur Auswahl stehen
  - dbConnector
  - seperatorFileConnector
  - templateFileConnector
  - summarizeConnector
- **Name**  
 Name des neuen Konnektors
- **Konnektor kann schreiben**  
 Durch Anhaken des Parameters wird definiert, ob dieser Konnektor schreiben kann (Quellkonnektor);  
 Ist der Parameter nicht angehakt, kann der Konnektor nur lesen (Zielkonnektor)

## DB Konnektor - Datenbank-Konnektor

Diese Konnektordefinition ermöglicht eine Datenbank im Webdesk in einem verknüpften Konnektor verwenden zu können. Damit können über den Webdesk Daten aus einer Datenbank gelesen bzw. in eine Datenbank geschrieben werden.

## Konfiguration von Konnektoren

Speichern 
 Speichern & Schließen 
 Zurück 
 Löschen

Vaterkonnektor:

Name:  \*

Connector kann schreiben:

### Datenbank Eigenschaften

Benutzer:  \* ?

Passwort:  ?

URL:  \* ?

### Teile des SQL Queries

Select Statement:  ?

From Statement:  \* ?

Where Statement:  ?

dbconnector\_escapeCharacter:  ?

### Einschränkungen

Haupttabelle (update/delete):  ?

In Haupttabelle darf geschrieben werden:  ?

## Datenbank Eigenschaften

- **Benutzer**  
Benutzername, der zum Einloggen verwendet werden soll
- **Passwort**  
Passwort, welches zum Einloggen verwendet werden soll
- **URL**  
Eingabe der URL der Datenbank, z.B.: jdbc:sqlserver://servername:port;databaseName=db\_name;);
  - Beispiel für MySQL:  
jdbc:mysql://192.168.3.12:3306/TestDB
  - Beispiel für MSSql:  
jdbc:mssql://192.168.3.12:1433/TestDB

## Teile des SQL Queries

- **Select Statement**  
Der Select-Teil des SQL Statements ist optional. Ist es leer, so wird 'select \*' verwendet.
- **From Statement**  
Der From-Teil des SQL Statements. (z.b.: from Tabellename\_aus\_dem\_Schema). Dieser Parameter muss angegeben werden.
- **Where statement**  
Der Where-Teil des SQL Statements ist optional. Ist er leer so ist keine Einschränkung vorgesehen.
- **dbConnector\_escapeCharacter**  
damit kann eine Maskierung für Sonderzeichen angegeben werden, um diese in SQL- Abfragen verwenden zu können (meistens /)

## Einschränkungen

- **Haupttabelle (update/delete)**  
Die Haupttabelle, in die schlussendlich geschrieben wird. Diese Information wird nur dann verwendet, wenn der Konnektor das Ziel darstellt.
- **In Haupttabelle darf geschrieben werden**  
Der Parameter definiert, ob es erlaubt ist in die Haupttabelle zu schreiben.

## Teste Verbindung

Dieser Button erlaubt zu überprüfen, ob mit den eingestellten Parametern die Verbindung funktioniert.

## Seperator File Connector

Ein Filekonnektor kann prinzipiell lesen und schreiben, kann aber auch so definiert werden, dass er nur liest oder nur schreibt.

Kann ein File Konnektor schreiben, so muss dies hier eingegeben werden, damit er auch in die Auswahlliste der möglichen Konnektoren aufscheint.

## Verwendete Datei / Lesen



Verwendete Datei / Lesen

Wähle Datei

Wähle Dateipfad am Server

c:\temp\CSG.csv

Dateiupload und extrahieren der Spalten

- **Wähle Datei**  
Aktuell von dem Konnektor verwendete Datei zum Einlesen. Ist dieses Feld befüllt, so wurde die Datei bereits in der Datenbank gespeichert.

- **Wähle Dateipfad am Server**  
Die Datei kann mit "Durchsuchen" auf dem Server ausgewählt werden.
- **Dateiupload und extrahieren der Spalten**  
Datei wird auf den Server geladen und analysiert. Dabei wird versucht die Spaltennamen aus der ersten Zeile auszulesen oder die Spaltennamen werden automatisch generiert.

### Verwendete Datei /Schreiben

- **Wähle Datei am Server**  
Speichert die Datei mit dem angegebenen Namen in die Datenbank. Folgende Platzhalter stehen zur Verfügung: \$d (=Tag), \$M (=Monat), \$y (=Jahr), \$h (=Stunde), \$m (=Minute) und \$s (=Sekunde)
- **Als Dokument speichern / Formatierungsmuster angeben**  
Schreibt die Datei unter dem angegebenen Pfad am Server. Folgende Platzhalter stehen zur Verfügung: \$d (=Tag), \$M (=Monat), \$y (=Jahr), \$h (=Stunde), \$m (=Minute) und \$s (=Sekunde)

### Dateiaufbau

- **Ein Trennzeichen wird benutzt, um die Spalten zu kennzeichnen**  
Die Daten in der Datei sind so aufgebaut, dass ein Trennzeichen das Spaltenende kennzeichnet. Meistens wird hierfür ein ; verwendet (z.B. in CSV- Files)
- **Spalten haben eine fixe Länge**  
Spalten werden nicht von einem Trennzeichen gekennzeichnet sondern haben immer eine bestimmte Länge z.B. 15 Zeichen.
- **Erste Zeile definiert Namen**  
Wenn gewählt, sollte die erste Zeile der Datei die Namen der Spalten beinhalten.
- **Kommentarkennzeichner**  
Definiert das Muster, das verwendet wird, um eine Zeile als Kommentar zu kennzeichnen.
- **Trennzeichen**  
Das Trennzeichen wird vom Konnektor benutzt, um Spalten in der Datei zu kennzeichnen.
- **Stringkennzeichner**  
Der Stringkennzeichner muss vor und nach der Zeichenfolge stehen und wird benutzt um diesen zu kennzeichnen. Dies kann durchaus nützlich sein, z.B. wenn ein Trennzeichen innerhalb der Zeichenfolge vorkommt.

### Definition der Spalten

## Definition der Spalten

Spaltenname	Offset	Länge	Bündigkeit	Auffüllen mit	Vorgabe	Prefix	Postfix	
<input type="text" value="PERSNR"/>								▲▼ □
<input type="text" value="NAME"/>								▲▼ □
<input type="text" value="Richtige OE"/>								▲▼ □
<input "falsch"""="" type="text" value="*Orga "/>								▲▼ □

Zeige erstellte Dateien (nur Dateien bei denen der Konnektor als Ziel diente werden angezeigt.)

- **Spaltenname**  
Name der Tabellenspalte
- **Offset**  
Definiert an welcher Stelle einer Zeile die Spalte beginnt.
- **Länge**  
Länge der vorhergehenden Spalte
- **Bündigkeit**  
Dieses Feld wird nur beim Schreiben berücksichtigt und definiert die Anordnung der Spalte beim Schreiben.
- **Auffüllen mit**  
Definiert die Zeilenfolge, die zum Auffüllen des Spalteninhaltes verwendet wird.
- **Vorgabe**  
Der default Wert. Wird verwendet wenn kein Wert gegeben ist.
- **Präfix**  
Der retournierte Wert wird um den Präfix angereichert.
- **Postfix**  
Der retournierte Wert wird um den Postfix angereichert.
- **Zeile hinzufügen / Ausgewählte Zeile entfernen**  
Durch Selektion der Spalte und Anklicken der Schaltfläche werden neue Zeilen hinzugefügt bzw. entfernt. Speichern.
- **Zeige erstellte Dateien**  
Es werden nur Dateien angezeigt, bei denen der Konnektor als Ziel diente.

## Template File Connector



**Konfiguration von Konnektoren**

Speichern Speichern & Schließen Zurück Löschen

Vaterkonnektor:

Name:

**Velocity Template**

Velocitytemplate: 

```
#foreach( $row in $rows)
  Vorname= $row.Vorname
  Nachname=$row.Nachname
#end
```

Alle Zeilenschaltungen (CRs + LFs) entfernen?

### Velocity Template

- **Velocity Template**

Hier können Sie Velocity- Scripts ausführen. Eine Referenz der Velocitybefehle finden Sie auf [hier](#)<sup>1</sup>

- **Alle Zeilenschaltungen (CRs + LFs) entfernen?**

Es werden alle Carriage Returns (CR = Eingaben von Return) und Line Feeds (LF = Eingabe von Space; Leerzeichen) automatisch aus der Zielfeile entfernt

Zusätzlich kann der Template Fileconnector gleichzeitig ein zweites File schreiben:

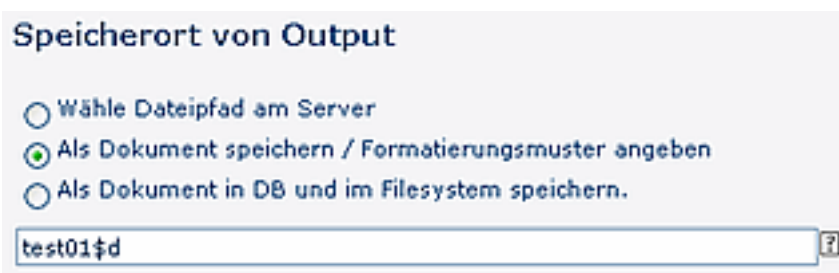


Template FileConnector für Weiterleitung

Save Object () und postProcess () werden auf eine 2. Instanz weitergeleitet, welche bereits vom Template File Konnektor selbst erzeugt wird. Die Entitätsnamen und die UID zur Speicherung von Ergebnis-Files muss an die 2. Instanz weitergegeben werden.

Hier wird der Name des 2. Template File Konnektors angegeben, an die im Falle der Ausführung von Schreiboperationen auf der aktuellen Konfiguration diese weitergeleitet werden.

### Speicherort von Output



**Speicherort von Output**

Wähle Dateipfad am Server

Als Dokument speichern / Formatierungsmuster angeben

Als Dokument in DB und im Filesystem speichern.

- **Wähle Datei am Server**  
Speichert die Datei mit dem angegebenen Namen in die Datenbank. Folgende Platzhalter stehen zur Verfügung: \$d (=Tag), \$M (=Monat), \$y (=Jahr), \$h (=Stunde), \$m (=Minute) und \$s (=Sekunde)
- **Als Dokument speichern / Formatierungsmuster angeben**  
Schreibt die Datei unter dem angegebenen Pfad am Server. Folgende Platzhalter stehen zur Verfügung: \$d (=Tag), \$M (=Monat), \$y (=Jahr), \$h (=Stunde), \$m (=Minute) und \$s (=Sekunde)
- **Als Dokument in DB und im Filesystem speichern**  
Speichert die Datei mit dem angegebenen Namen in die Datenbank. Folgende Platzhalter stehen zur Verfügung: \$d (=Tag), \$M (=Monat), \$y (=Jahr), \$h (=Stunde), \$m (=Minute) und \$s (=Sekunde)

## Spalten Definition

- **Spaltenname**  
Definiert den Namen der Spalte, der später in den verknüpften Konnektoren ersichtlich ist
- **Zeile hinzufügen / ausgewählte Zeilen entfernen**
- **Zeige erstellte Dateien**  
Es werden nur Dateien angezeigt, bei denen der Konnektor als Ziel diene

## Summarize Konnektor

- **Quellkonnektor**  
Selektion des Quellkonnektors
- **Spalte für Personalnummer**  
xx
- **Überschrift der Zielgruppe**

XX

- **Überschrift der Summe**

XX

- **Null-Werte rausfiltern**

Durch Anhaken des Parameters werden Zeilen mit 0-Beträgen nicht angezeigt,

## Überschriften der Spalten

Die Spaltenüberschriften sind abhängig vom Quellkonnektor.

- **Quellfeld**

XX

- **Javascript**

XX

- **Zielgruppe**

XX

- **Hinzufügen / Löschen**

Durch Anklicken der Schaltfläche können neue Überschriften hinzugefügt werden, bzw. unerwünschte Überschriften gelöscht werden.

Bei Fragen zur detaillierten Funktionsweise der Parameter kontaktieren Sie bitte die Workflow EDV GmbH.

## Notes Konnektor

**Konfiguration von Konnektoren**

**Speichern** **Speichern & Schließen** **Zurück** **Löschen**

Vaterkonnektor: notesConnector

Name: NotesKonnektor

Connector kann schreiben:

### Datenbank Eigenschaften

Servername:

Benutzername:

Passwort:

Datenbank:

Such String:

Spaltenname	Art des Feldwertes	Index
<b>Zeile hinzufügen</b>   <b>ausgewählte Zeile(n) entfernen</b>		

### Datenbank Eigenschaften

- Servername
- Benutzername
- Passwort
- Datenbank
- Such String



Spaltenname	Art des Feldwertes	Index
<input type="text"/>	Wert des Feldes lt. Index <input type="button" value="v"/> <input type="button" value="i"/>	<input type="text"/> <input type="button" value="i"/> <input type="button" value="u"/> <input type="button" value="d"/>
<input type="button" value="Zeile hinzufügen"/>		<input type="button" value="ausgewählte Zeile(n) entfernen"/>
<input type="button" value="Server- und Datenbankverbindung testen"/>		

- Spaltenname
- Art des Feldwertes  
Dieser Parameter definiert, ob der Wert des Notes-Feldes laut Index, bzw. als Zeichenkette getrennt mit Strichpunkt (sofern mehrere Einträge vorhanden sind) ausgegeben wird
- Index  
Wenn bei Type "Wert des Feldes laut Index" angegeben wird, muss der Index angegeben werden

## TmTravel Konnektor

 **Konfiguration von Konnektoren**

Vaterkonnektor:

Name:

zu setzender Reisestatus:

Überleitungsdatum setzen:

- **Zu setzender Reisestatus**  
Dieser Parameter ermöglicht es, den Status zu bestimmen, der nach Abschluss gesetzt werden soll. Der angegebene Status wird nach erfolgreichem Auslesen und Durchführung des Syncs auf allen selektieren Dienstreisen gesetzt, Beispiel: Expense\_Transfered (Abrechnung Übergeleitet)
- **Überleitungsdatum setzen**  
Ist diese Option aktiviert, so wird nach erfolgreichem Auslesen und Durchführung der Synchronisation über das Connectorframework das Überleitungsdatum mit Heute befüllt!

1. <http://www.jajakarta.org/velocity/velocity-1.2/docs/vtl-reference-guide.html>