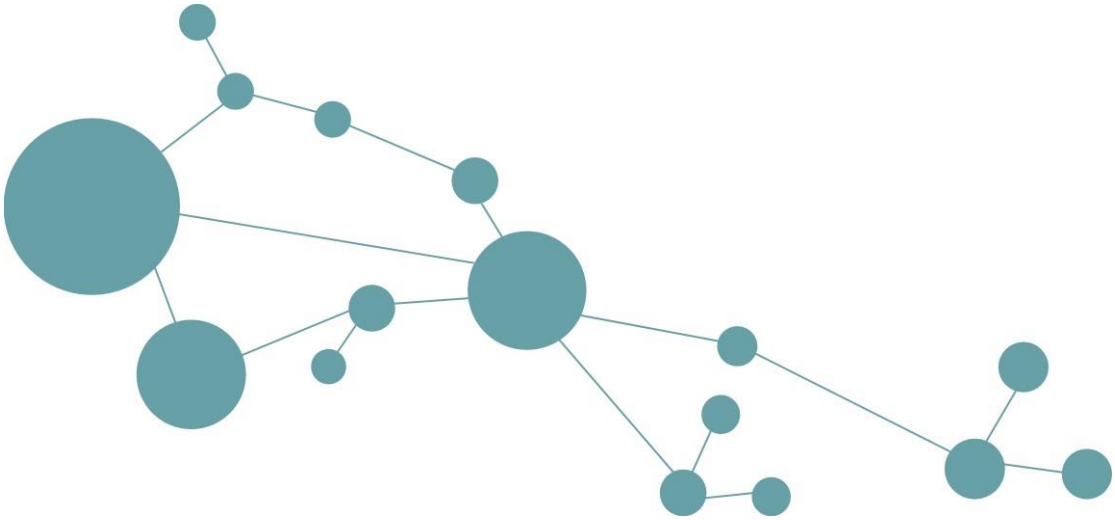


Admin-Tool 3.2

i·views

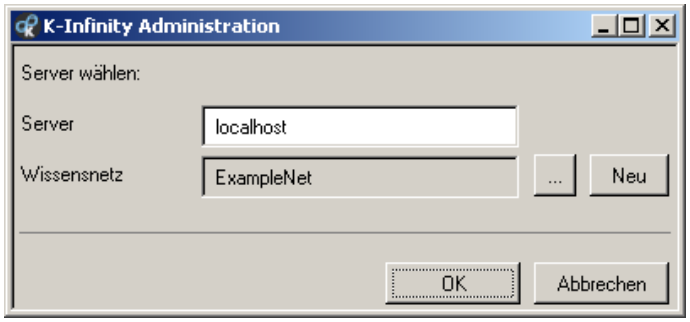
Inhaltsverzeichnis

1 Einleitung

Mit dem Admin-Tool können verschiedene Administrationsaufgaben in einem Wissensnetz durchgeführt werden. Es können Netze angelegt, Benutzer eingesellt und konfiguriert, Wartungsaufgaben durchgeführt und diverse Einstellungsmöglichkeiten vorgenommen werden.

1.1 Anmeldung

Nach dem Start des Admin-Tools (Windows: "admin.exe" ausführen, Unix: "visual admin.im" in der Shell eingeben) erscheint das folgende Anmeldefenster:



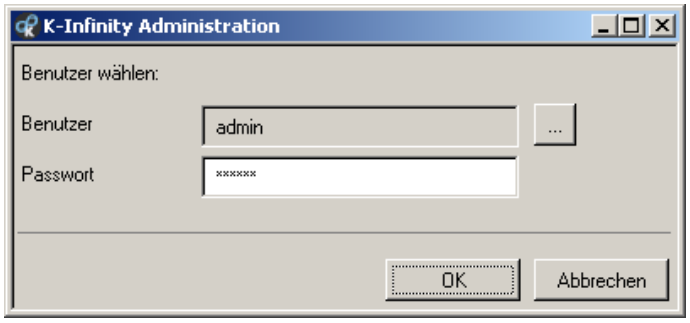
Bei "Server" gibt man den Netzwerknamen des K-Infinity-Servers an. Sollte dieser nicht auf den Standardport (x: 30053) konfiguriert sein, muss man die Portnummer zusätzlich angeben (Schreibweise: "Servername:Portnummer").

Wenn man keinen Server angibt, startet das Admin-Tool im Standalone-Betrieb ohne Server. Die zu administrierenden Netze werden dann im Unterverzeichnis "volumes" des Admin-Tools gesucht.

Das zu administrierende Wissensnetz kann dann mit dem Button "..." ausgewählt werden. Mit "Neu" kann ein neues Wissensnetz angelegt werden.

Informationen über die Softwareversion des Admintools erhält man mit "Info".

Wenn ein Netz ausgewählt und mit "Weiter" das Netz betreten wurde, wird nach einer Benutzerkennung samt Passwort gefragt:



Mit "..." kann der Benutzer aus der Liste der im Netz angelegten Benutzern mit Administratorrechten auswählt werden. Normale Benutzer werden hier nicht zur Auswahl angeboten.

Klickt man nun auf "Start", dann startet das Admin-Tool zum Administrieren des Netzes. Mit "Zurück" kehrt man nochmal zur Auswahl des Netzes zurück.

1.2 Anlegen eines neuen Netzes

Beim Anlegen eines neuen Netzes, wird zuerst nach dem Namen des Wissensnetzes gefragt.

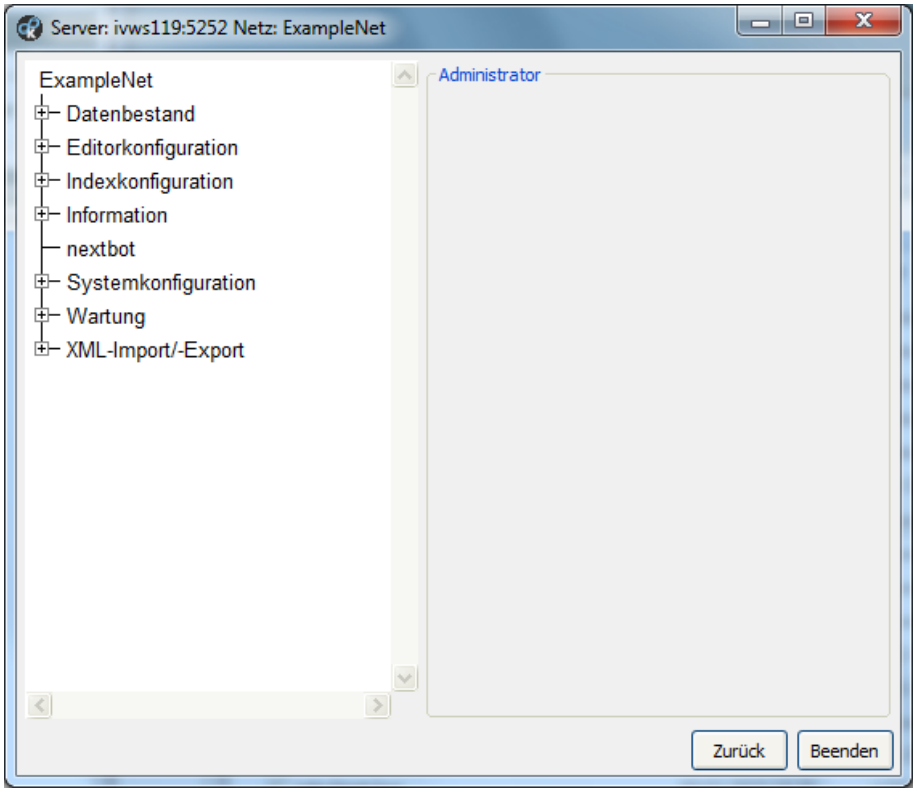
Danach wird nach dem Namen und dem Passwort für den Administrator-Benutzer gefragt.

Im Admin-Tool können später weitere Benutzer hinzugefügt werden. Das Admin-Tool kehrt nun zum Anmeldebildschirm zurück und man kann sich am neu erzeugten Wissensnetz anmelden.

1.3 Hauptfenster

Nach dem Anmelden erscheint das zweigeteilte Hauptfenster. Auf der linken Seite sind die zu konfigurierenden Aspekte des Netzes alphabetisch in einer Baumstruktur aufgeführt, auf der rechten Seite werden die Einstellungsmöglichkeiten angezeigt.

Die verfügbaren Einstellungsmöglichkeiten unterscheiden sich bei projektspezifischen Versionen des Admin-Tools evtl. von den hier aufgeführten.



Hauptfenster des Admin-Tools

Mit dem Knopf "Zurück" gelangt man wieder zum Anmeldefenster. Klickt man auf den Knopf "Beenden", dann wird das Fenster geschlossen und das Admin-Tool beendet sich.

Im Folgenden werden nun die einzelnen Konfigurationsmöglichkeiten beschrieben.

2 Datenbestand

Im Teilbaum "Datenbestand" kann der Datenbestand des Wissensnetzes gesichert, wiederhergestellt, geladen, kopiert und auch gelöscht werden.

2.1 Transfer

Unter dem Punkt "Transfer" stehen vier Aktionen zum Vervielfältigen des Wissensnetzes zur Verfügung.

Die Server-Dienste *Download*, *Download Log* und *Upload* sind nur verfügbar, wenn das Admin-Tool an einem K-Infinity-Server angemeldet ist.

Download

Download lädt ein Wissensnetz vom Server in das lokale Wissensnetzverzeichnis (Unterverzeichnis "volumes") herunter.

Kopieren nach ...

Kopiert das Netz auf dem Server in ein Netz mit neuem Namen, der vor dem Kopieren eingegeben werden muss.

Download Log

Lädt die Logdatei des K-Infinity-Servers in das lokale Verzeichnis herunter.

Upload

Upload eines lokalen Netzes. Dieses liegt im lokalen Wissensnetzverzeichnis (Unterverzeichnis "volumes") und wird auf den Server kopiert.

2.2 Verwalten

Unter dem Punkt "Verwalten" stehen vier Aktionen zur Sicherung und zum Löschen des Wissensnetzes zur Verfügung.

Sichern

Es wird eine Sicherung des gesamten Netzes erstellt. Neben dem *volumes*-Verzeichnis wird ein *backup*-Verzeichnis (bzw. ein Verzeichnis, das bei der Mediatorkonfigurations angeben ist) angelegt. In dieses wird in ein Unterverzeichnis mit Zeitstempel das kopierte Netz abgelegt.

Der Administrator hat nun die Möglichkeit zu warten bis die Sicherung fertiggestellt ist oder direkt weiterzuarbeiten. In diesem Fall wird im Hintergrund das Backup erstellt.

Siehe auch Dokumentation zum K-Infinity-Server für weitere Informationen.

Wiederherstellen

Mit dieser Aktion lassen sich sichergestellte Netze wiederherstellen. Bei diesem Vorgang erscheinen drei Eingabe- bzw. Auswahlfenser.

1. Netz auswählen, das wiederhergestellt werden soll
2. Sicherungskopie des Netzes anhand des Datums auswählen
3. Neuen Netznamen eingeben

Löschen

Beim Löschvorgang wird vor dem Löschen ein Backup des Wissensnetzes erstellt und erst danach das Netz gelöscht. Der Administrator wird automatisch abgemeldet und das Anmeldefenster des Admin-Tools wieder angezeigt. Zur Sicherheit wird vor dem Löschen nochmals nach dem Namen des Netzes gefragt.

Backup löschen

Es lassen sich einzelne Backups löschen. Auch bei diesem Vorgang erscheinen drei Eingabe- bzw. Auswahlfenser.

1. Netz auswählen, dessen Sicherungskopie gelöscht werden soll
2. Zu löschende Sicherungskopie anhand des Datums auswählen
3. Nochmalige Löschbestätigung

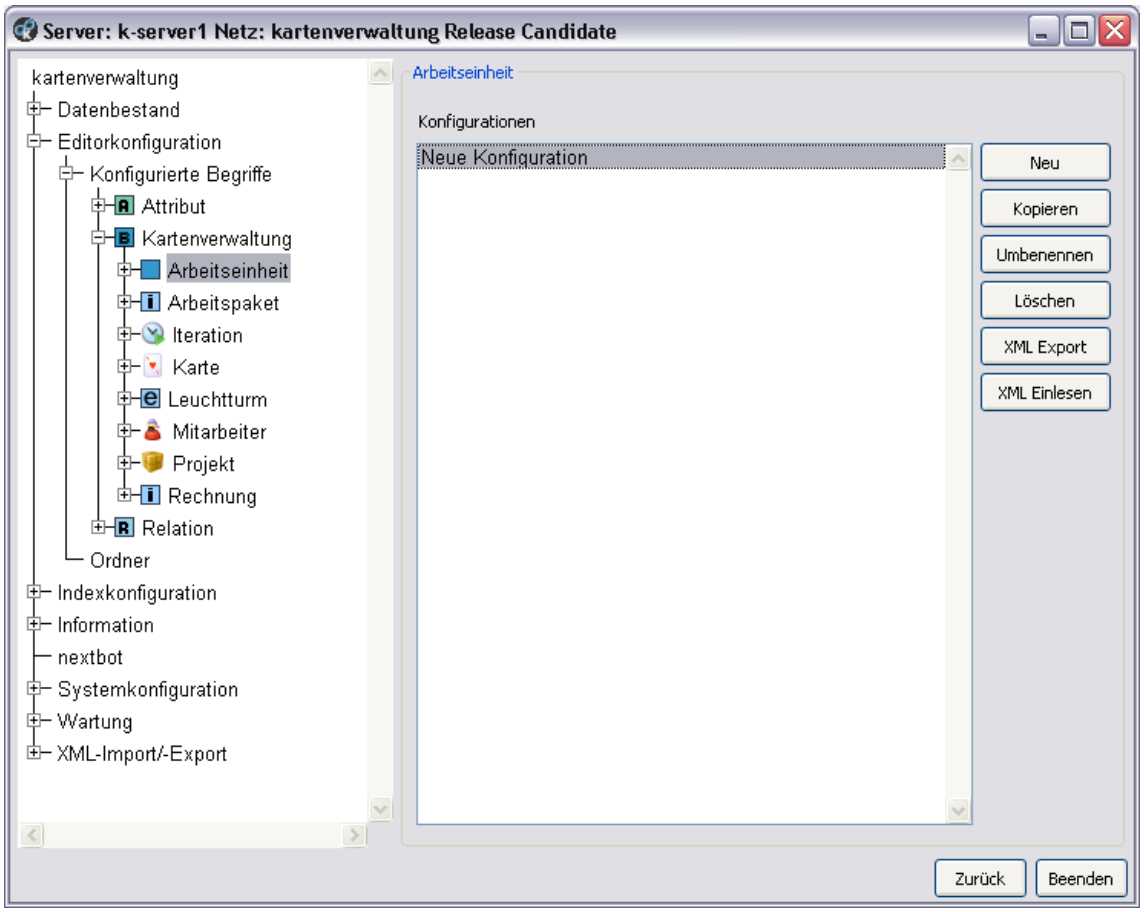
3 Editorkonfiguration

Unter den Aspekt "Editorkonfiguration" kann für jeden Begriff im Wissensnetz konfiguriert werden, welche Eigenschaften der Individuen in den Objektlisten des Knowledge-Builders

angezeigt werden sollen. Auch hier gilt die Vererbung: ist für einen Begriff keine Konfiguration angegeben, dann wird die des Oberbegriffs verwendet.

3.1 Erstellen von Konfigurationen

Die Begriffe mit Editorkonfigurationen werden unter dem Knoten "Konfigurierte Begriffe" gemäß ihrer Zugehörigkeit zu den Teilnetzen im Wissensnetz aufgeführt. Die Bäume für Attribut- und Relationsbegriffe werden neben den Bäumen für die Teilnetze aufgebaut.



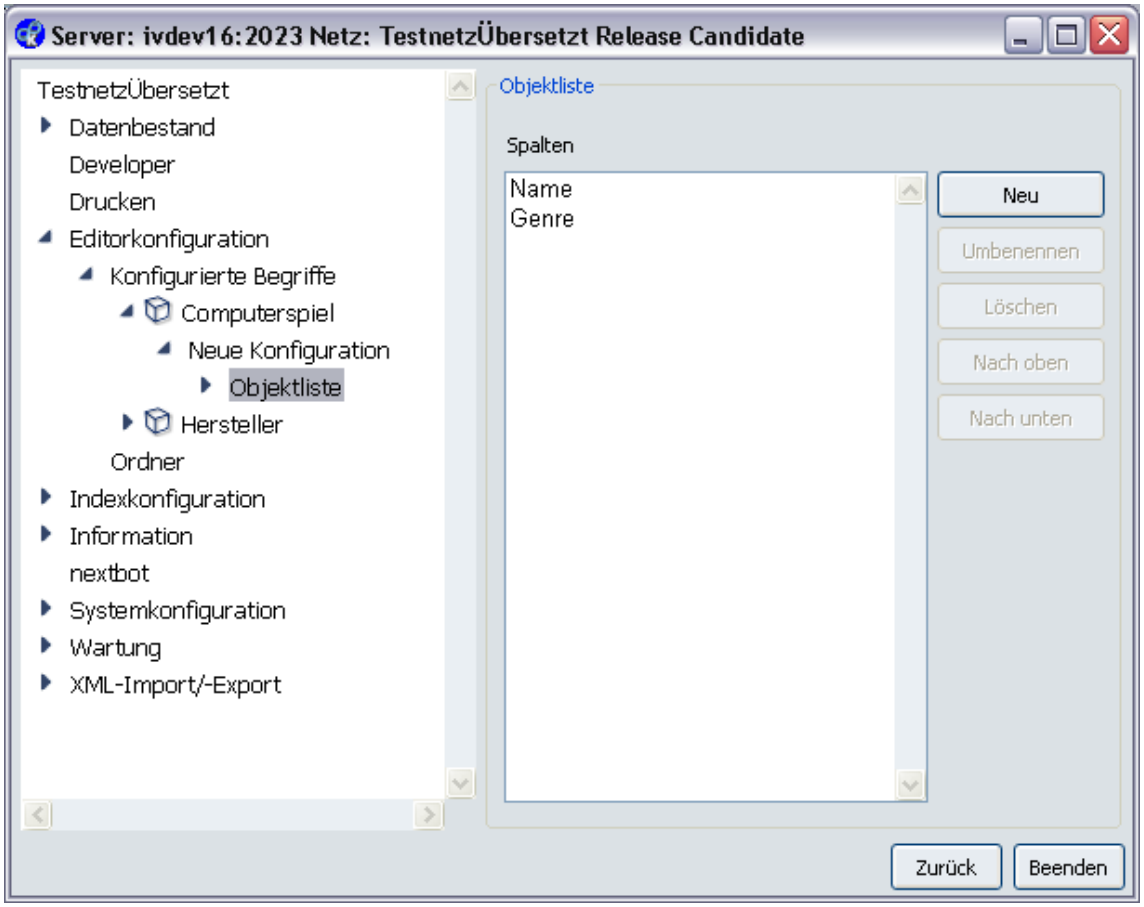
Liste der Konfigurationen für den Begriff Arbeitseinheit

Für jeden Begriff im Baum lassen sich hier mehrere Konfigurationen erstellen, kopieren, umbenennen und löschen. Darüber hinaus kann eine Konfiguration für einen Begriff exportiert und wieder importiert werden. In den Unterstrukturen bietet sich auch die Möglichkeit kleinere Einheiten der Konfiguration zu exportieren und zu importieren. Klickt man auf das Plus neben dem Begriff im Baum, dann werden die Konfigurationen im Baum unter dem Begriff aufgelistet.

3.2 Konfiguration der Objektlisten

Durch die Konfiguration der Objektlisten wird der Anzeigeumfang der Tabellenansicht festgelegt. Zusätzlich zur Bestimmung der Spalten können weitere Eigenschaften spezifiziert werden, die bei einer Suche ebenfalls für in dieses Feld eingetragene Suchparameter berücksichtigt werden sollen.

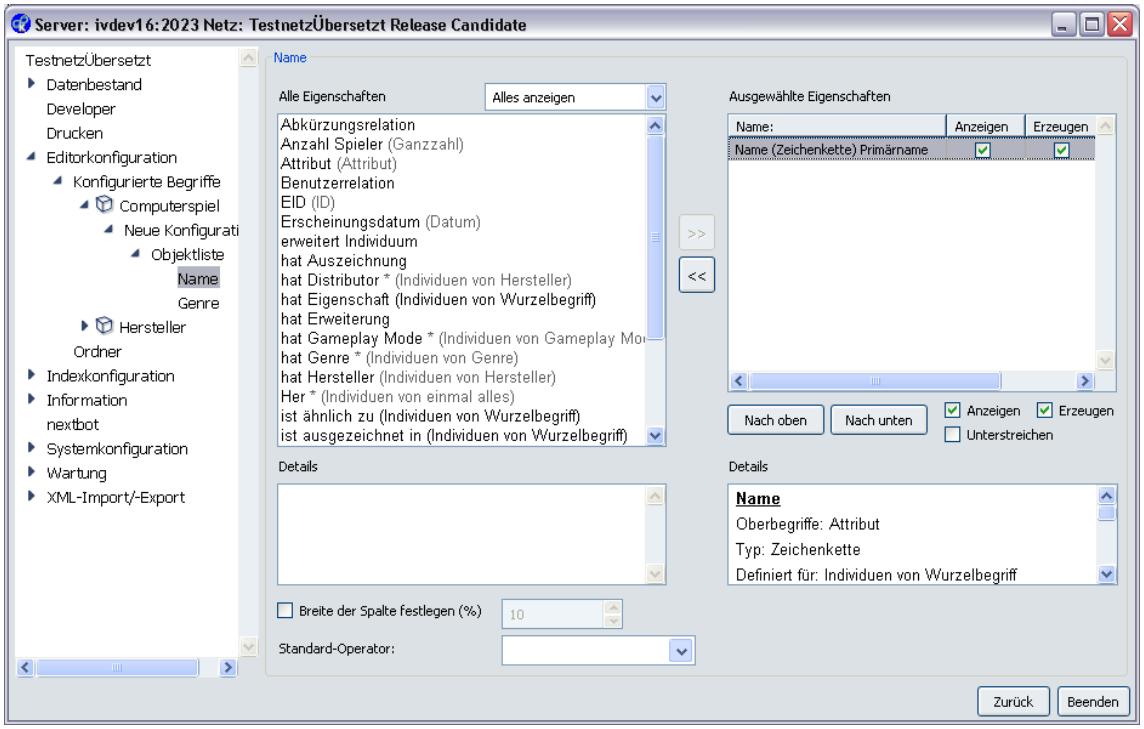
Ist links im Baum eine Objektliste selektiert, dann erscheint rechts im Fenster eine Übersicht über die für die Tabellenansicht konfigurierten Spalten.



Konfiguration der Objektliste

Durch Klicken auf den Knopf „Neu“ lassen sich neue Spalten für die Tabellenansicht anlegen. Der dabei für die Spalte vergebene Name dient nur der Unterscheidung im Konfigurationseditor. Die im Knowledge-Builder angezeigte Spaltenüberschrift ergibt sich aus dem Namen der Eigenschaft, die dieser Spalte an erster Stelle zugeordnet ist. Die Spalten lassen sich hier umbenennen und löschen. Durch die Knöpfe „Nach oben“ und „Nach unten“ lässt sich die Reihenfolge der Spalten in der Tabellenansicht festlegen. Die oberste Spalte hier in der Liste, ist die Spalte ganz links in der Tabellenansicht.

Öffnet man den Unterbaum des Aspekts "Objektliste", dann werden die Spalten im Baum aufgelistet. Selektiert man eine Spalte, dann kann diese im rechten Teilfenster konfiguriert werden.



Das rechte Teilfenster ist in zwei Hälften unterteilt. Die linke Hälfte zeigt die zur Verfügung stehenden möglichen Eigenschaften, die rechte Hälfte zeigt die für die Spalte ausgewählten Eigenschaften. Die Liste der möglichen Eigenschaften kann durch die Auswahl eines Begriffs im Klappmenü eingeschränkt werden.

Wählt man eine Eigenschaft sowohl links oder rechts an, dann werden deren Eigenschaften wie Obebegriff oder Typ im unteren Texteditor "Details" angezeigt.

Das Hinzufügen zu und Entfernen von Eigenschaften aus der Liste der ausgewählten Eigenschaften geschieht durch Selektieren der Eigenschaft in der jeweiligen Liste und Anklicken von „<<“ und „>>“.

Um die Reihenfolge der ausgewählten Eigenschaften festzulegen, muss man die zu positionierende Eigenschaft in der rechten Liste selektieren und durch Anklicken von „Nach oben“ und „Nach unten“ an die korrekte Position bringen. Alle für eine Spalte akkumulierten Eigenschaften werden später bei der Suche zum Auffinden von Objekten herangezogen. Der Name der obersten Eigenschaften wird für die Darstellung der Spalte in der Tabellenansicht verwendet. Außerdem wird die oberste Eigenschaft im Falle des Neuanlegens eines Objekts, sollte das jeweilige Feld in der Tabellenansicht einen Wert enthalten, mit diesem Wert am Objekt ausgeprägt.

Das Häkchen "Anzeigen" gibt an, ob der Wert der Eigenschaft in der Tabellenansicht angezeigt werden soll. Ist dies bei mehreren Eigenschaften der Fall, dann werden deren Werte nebeneinander in der Spalte in der Tabellenansicht angezeigt.

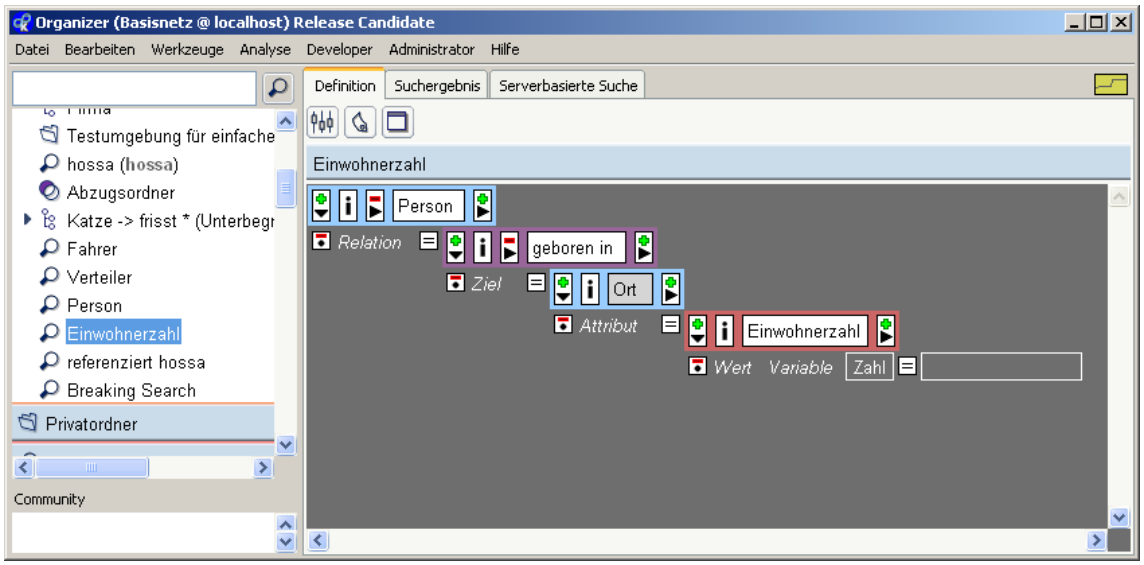
Das Häkchen "Erzeugen" gibt an, ob diese Eigenschaft beim Erzeugen eines neuen Objekts erzeugt werden soll, falls das entsprechende Feld in der Objektliste einen Wert enthält. Sollte bei mehreren Eigenschaften das Häkchen gesetzt sein, dann wird die Eigenschaft ausgeprägt, die weitesten oben in der Liste steht.

Das Häkchen "Unterstreichen" gibt an, ob der Wert der Eigenschaft in der Objektliste unterstrichen werden soll.

3.2.1 Expertensuchbausteine

Eine Besonderheit bei der Konfiguration der Spalten stellt der Auswahlpunkt "Expertensuch-Baustein" in der Liste der zu konfigurierenden Möglichkeiten dar. Über ihn lassen sich Informationen auf Spalten abbilden, die nicht direkt am jeweiligen Individuum angebracht sind, sondern die das Ergebnis einer Expertensuche darstellen, die für die jeweiligen Individuen definiert wird.

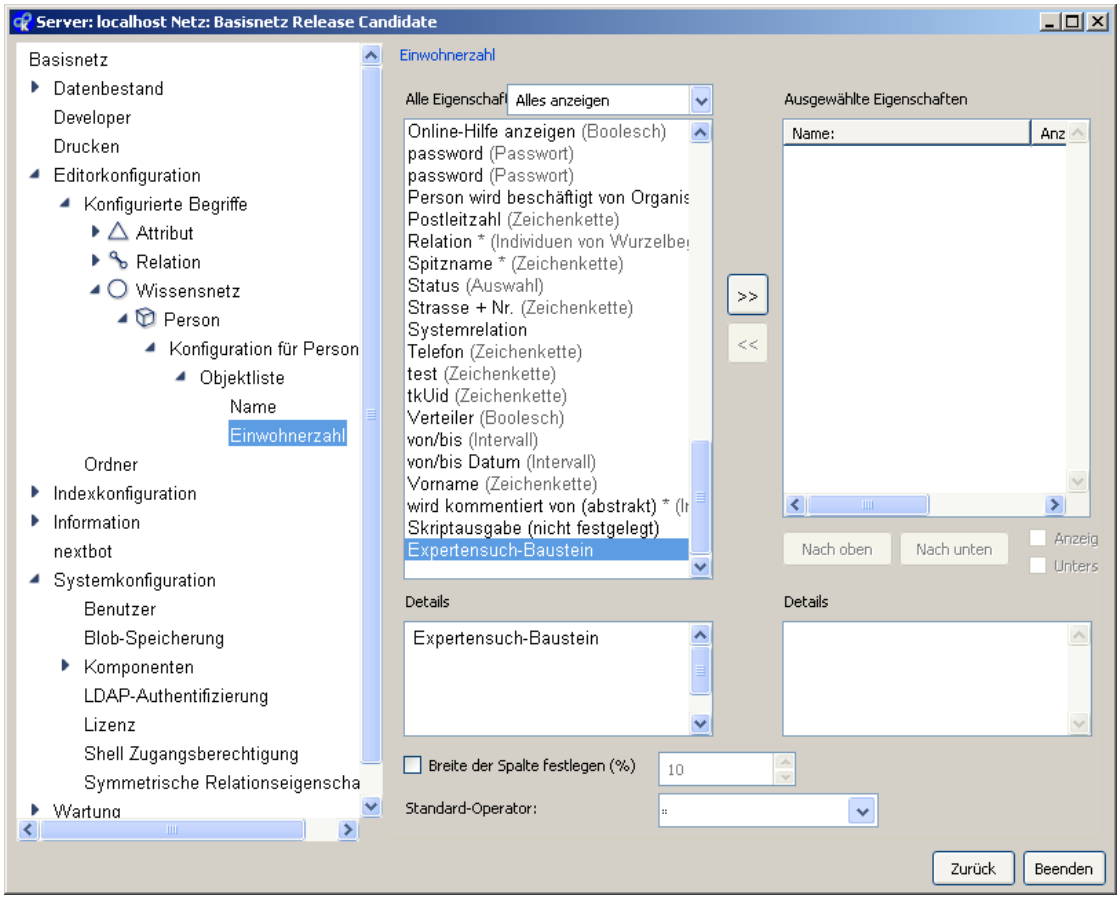
Ein Beispiel: Für Personen soll in einer SPalte der Objektliste angegeben werden, wieviele Einwohner die Geburtstadt hat. Die Person ist mit der Geburtstadt mit einer Relation verknüpft, die Einwohnerzahl ist ein Attribut der Individuen von Stadt. Zunächst muss ein entsprechende Expertensuche formuliert werden:



Expertensuche mit Suchparameter Zahl

Diese muss parametrisierbar sein, damit der Suchwert, der ja in der Objektliste für die Filterung der anzuzeigenden Objekte verwendet wird, übergeben werden kann. Dabei gilt, dass prinzipiell mehrere Parameter eingefügt werden können, diese aber nur dann sinnvoll belegt werden können, wenn ihnen allen derselbe Name gegeben wird, was der Gleichbelegung mit dem übergebenen Suchwert entspricht.

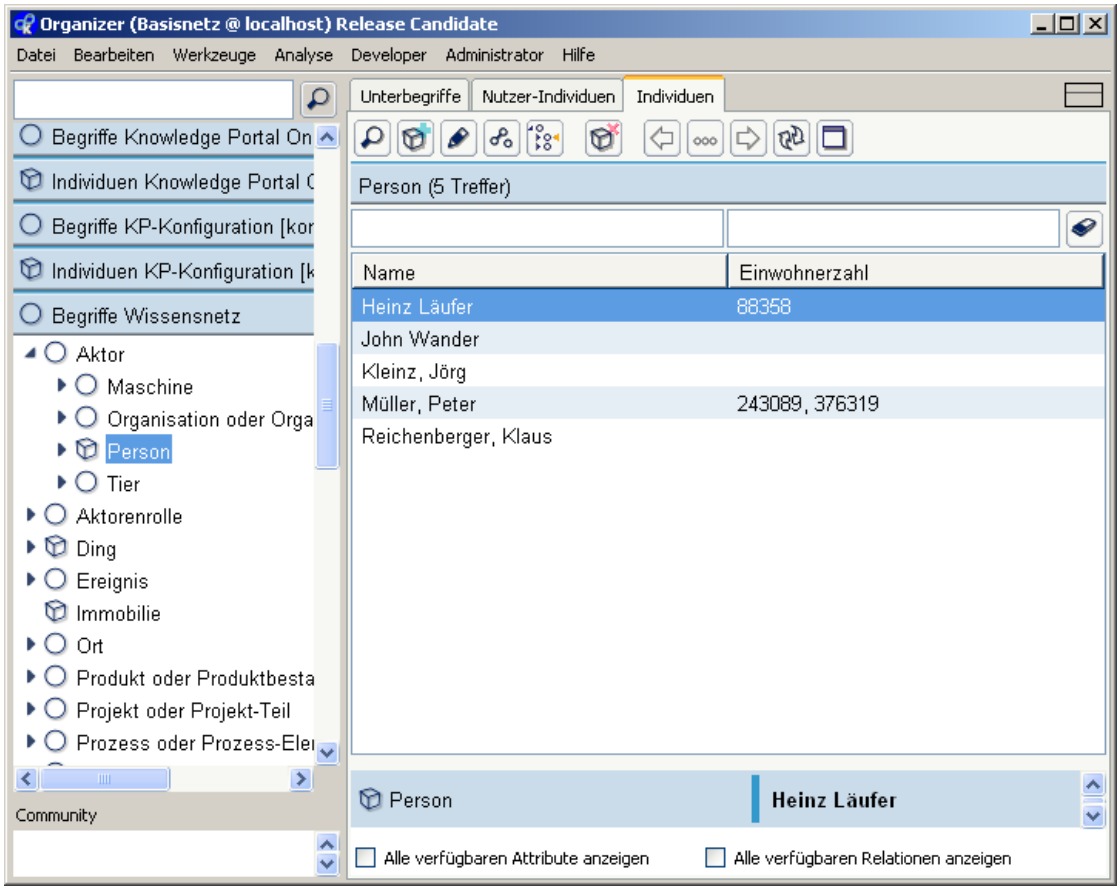
Danach kann die Spaltenkonfiguration der Objektliste erfolgen. Der Auswahlpunkt "Expertensuch-Baustein" ist in der Liste ganz unten aufgeführt:



Spaltenkonfiguration des Expertensuch-Bausteins

Beim Übernehmen werden per Dialog die im Wissensnetz vorhandenen Expertensuchen zur Auswahl angeboten. Die ausgewählte Expertensuche wird in die Editorkonfiguration kopiert, d.h. dass etwaige spätere Änderungen an der Expertensuche sich **nicht** in die Editorkonfiguration durchschlagen! Ein sorgfältiger Umgang mit den gewünschten Expertensuchen sowie ihrer Übernahme in die Editorkonfiguration ist daher vonnöten.

Eine anschliessende Betrachtung der Objektliste spiegelt die Verwendung wider:



Expertensuch-Baustein in der Objektliste

3.2.2 Skriptausgabe

Neben einem Expertensuchbaustein kann auch ein KSkript verwendet werden, um die Ausgabe einer Spalte zu steuern. Das ausgewählte (registrierte) Skript muss dann zu jedem Objekt, das eine Spalte bildet, einen Rückgabewert für die Spalte liefern.

Wird in einer Objektliste ein Suchwert in eine Skript-Tabellenspalte eingetragen, so muss das Ergebnis gefiltert werden, d.h. zu jeder Trefferzeile (passend zu den anderen Spalten) wird der Rückgabewert des Skriptes ermittelt und mit der Sucheingabe verglichen.
Deshalb bedeutet eine solche Suche auf einer Skript-Tabellenspalte auch eine Performanzeinbusse. Diese kann jedoch deutlich abgemildert werden:

Wenn es in dem ausgewählten Skript eine Funktion mit Namen "objectListScriptResults" und einem deklarierten Parameter gibt, so wird diese Funktion mit dem Argument der zugehörigen Sucheingabe aufgerufen, um die Menge der passenden Objekte zurückzuliefern. Die Funktion wird auf dem Wurzelbegriff oder der bisherigen Treffermenge als Ausgangsobjekt aufgerufen - je nach dem, wie die Suche am besten gelöst werden kann und ob die Option "Ausgabe ist von der Eingabe unabhängig" gesetzt ist. Damit diese Variante wirklich effizient wird, ist es empfehlenswert, die Sucheingabe entsprechend auszuwerten und mit dem Ergebnis eine registrierte Expertensuche aufzurufen, um deren Ergebnis an die Objektliste weiterzuleiten.

Beispiel:

Jedem User wird bei anderen Personen die Entfernung zum eigenen Wohnort angezeigt. Die Eingabe im zugehörigen Suchfeld wird als maximale Entfernung in die Suche übernommen. Entfernungen werden in Meter (m) oder Kilometer (km) angezeigt.

Das registrierte Skript "**Abstand**":

```
<?xml version="1.0" encoding="UTF-8"?>
<Script>
    <Function arguments="searchValue" name="objectListScriptResults">
        <SetVariable
            value="userInstance()/~$wohnort$/target()/@$position$"
            variable="sourcePos"/>
        <AnalyzeString regex="([0-9]+)\s*([kK]?)" string="var(searchValue)">
            <matchingSubstring>
                <SetVariable
                    value="regexGroup(1)/asNumber()"
                    variable="distance"/>
                <SetVariable
                    value="regexGroup(2)"
                    variable="km"/>
            </matchingSubstring>
        </AnalyzeString>
        <If test="var(distance)!=null and var(km)!=''">
        <do>
            <SetVariable
                value="var(distance) * 1000"
                variable="distance"/>
        </do>
        </If>
        <If test="variable(sourcePos)=null">
        <do>
            <Return value="emptySet()"></Return>
        </do>
        <else>
            <SetVariable
                value="concat('~', var(sourcePos),var('distance'))"
                variable="param"/>
            <ExpertQuery queryName="personenImUmkreis">
                <parameter name="standort" value="var(param)"/>
                <Return value="topic()"/>
            </ExpertQuery>
        </else>
        </If>
    </Function>

    <SetVariable
        value="userInstance()/~$wohnort$/target()/@$position$"
        variable="sourcePos"/>
    <SetVariable
        value="/~$wohnort$/target()/@$position$"
        variable="position"/>
    <If test="var(position)=null">
```

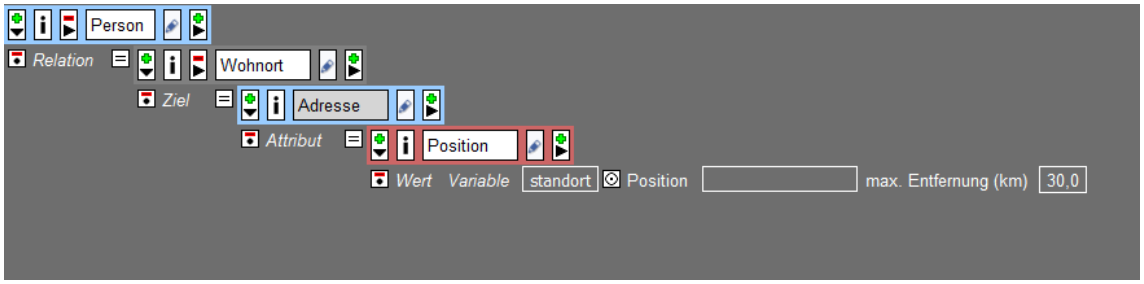
```
    <do>
        <Output>?</Output>
    </do>
    <else>
        <SetVariable
            variable="colon"
            value="false"/>
        <Path path="variable(position)/distanceTo(variable(sourcePos))">
            <Each>
                <SetVariable
                    value="."
                    variable="distance"/>
                <If test="var(colon)='true'">
                <do>
                    <Output>, </Output>
                </do>
                <else>
                    <SetVariable
                        variable="colon"
                        value="true"/>
                </else>
                </If>
                <If test="var(distance)>1000">
                <do>
                    <SetVariable
                        value="var(distance) / 1000"
                        variable="distance"/>
                    <SetVariable
                        value="' km'"
                        variable="suffix"/>
                </do>
                <else>
                    <SetVariable
                        value="' m'"
                        variable="suffix"/>
                </else>
                </If>
                <Output><path path="rounded( var(distance) )"/></Output>
                <Output><path path="var(suffix)"/></Output>
            </Each>
        </Path>
    </else>
    </If>
</Script>
```

Die Funktion **objectListScriptResults()** bestimmt die "**sourcePos**" des Wohnortes der Benutzerinstanz und zerlegt die Eingabe "**searchValue**" in "**distance**" und "**km**". Letzteres wird geprüft, ob die Entfernung von Kilometer in Meter umgerechnet werden muss.
Dann kann die Expertensuche "**personenImUmkreis**" mit dem konstruierten Suchparameter **sourcePos**~**distance** aufgerufen werden, um das Ergebnis für die Treffermenge zurückzugeben.

Im Hauptteil des Skriptes werden die Positionen ermittelt für den Wohnorte der Benutzerinstanz "**sourcePos**" und für das Objektes der jeweiligen Zeile "**position**". Der berechnete Abstand "**distance**" wird je nach Größe als Zeichenkette mit Einheit Metern oder in Kilometern

zurückgeliefert.

Die verwendete Suche sieht so aus:



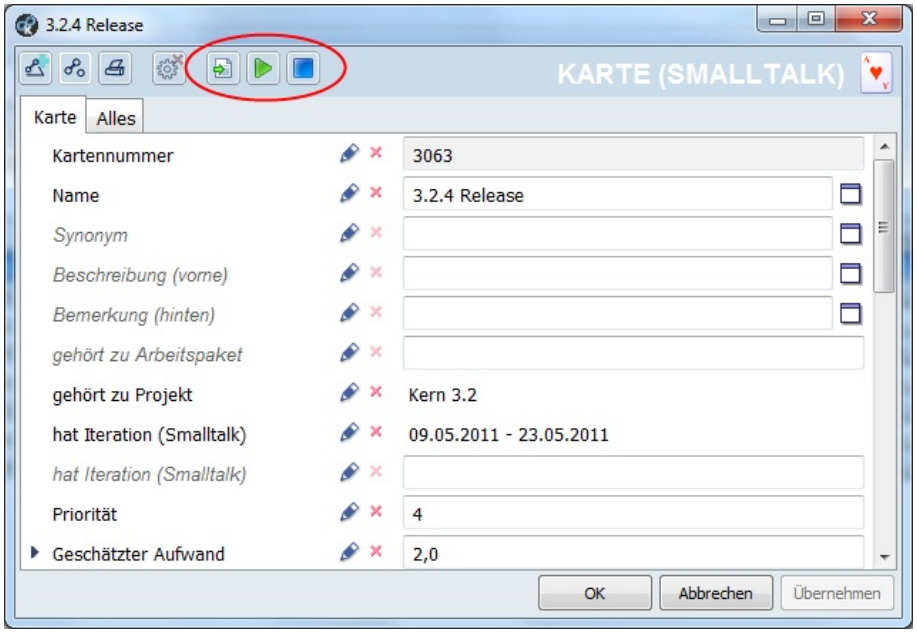
Eine entsprechende Objektliste sieht nach Suche für "35 km" so aus:

		35 km
Name	Wohnort	Entfernung
Frau Rauscher	Römerberg 23, Frankfurt	28 km
Testuser	Julius Reiberstr. 17, Darmstadt	0 m

3.3 Actionscript-Buttons

Im Admin-Tool können innerhalb der Editorkonfiguration Buttons für die Objekteliste und den Detailseditor definiert werden. Nach der Aktivierung eines dieser Knöpfe wird ein vom Administrator eingetragenes Skript ausgeführt.

Ansicht im Knowledge-Builder:

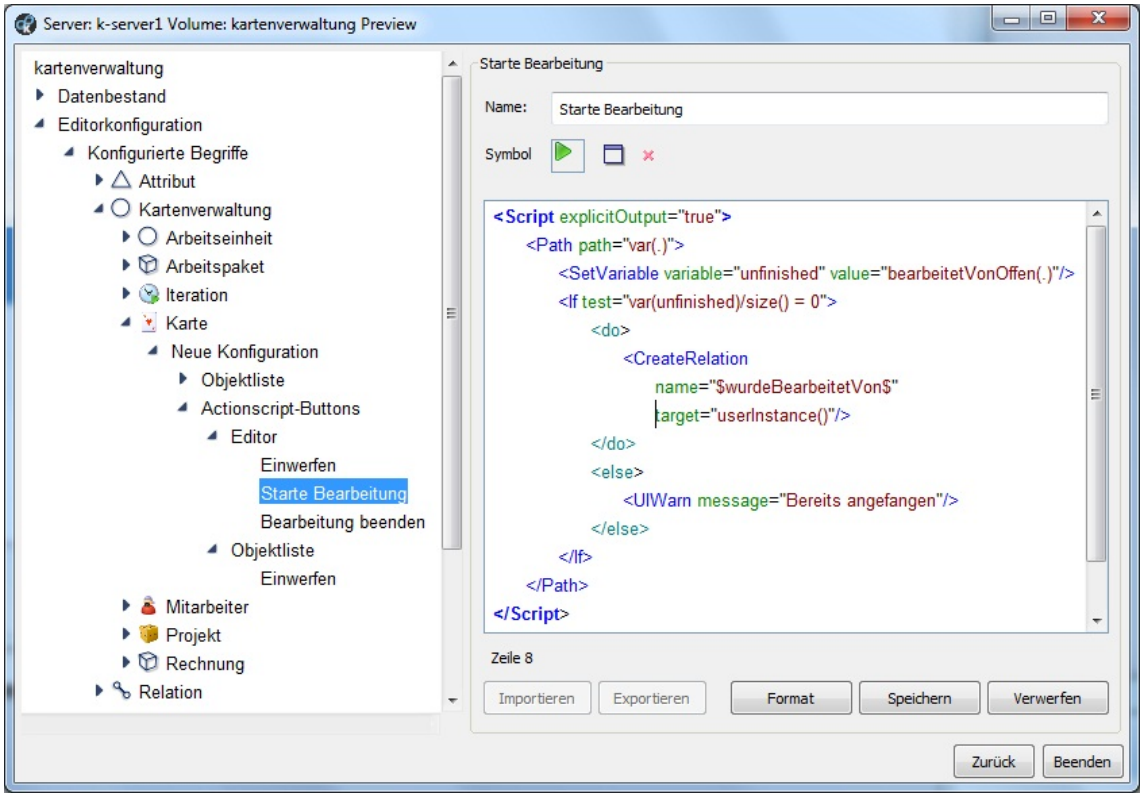


Rot umrandet sind drei Actionscript-Buttons zu sehen.

Konfiguration im Admin-Tool:

Zu den Konfigurationsmöglichkeiten einer beliebigen Konfiguration in der Editorkonfigura-

tion ist die Gruppe *Actionscript-Buttons* hinzugekommen. In den Untergruppen *Editor* und *Objektliste* lassen sich die Actionscript-Buttons definieren.



Es müssen Name, Symbol und Skript angegeben werden.

Der Name wird als Tooltip im Knowledge-Builder angezeigt. Das ausgewählte Symbol, eine beliebige Bilddatei, wird auf Buttongröße skaliert.

Anmerkung: Ist kein Symbol angegeben, wird kein Button angezeigt.

UI-Funktionen

Das Skript kann auf UI-spezifische Funktionen zurückgreifen. Wegen dieser Möglichkeit wird das Script nicht innerhalb einer einzigen Transaktion ausgeführt. Zwar werden von den Funktionen bei Bedarf Transaktionen verwendet, aber diese klammern dann nur die einzelne Funktion.

```
<UIWarn message="${name() + ' stinkt'}"/>
```

Zeigt eine Meldung an

```
<UIRequestString message="'Name'"/>
```

Benutzer kann eine Zeichenkette eingeben

```
<UIConfirm message="'Sind Sie versichert?'"/>
```

Öffnet einen Abbrechen-Dialog

```
<UIChooseObject message="'Person'" objects="//Person/instances()"/>
```

Objekt aus einer Menge auswählen lassen

```
<UIOpenEditor/>
```

Standardeditor für das Objekt öffnen

```
<UIOpenScriptViewer scriptName="EditorScriptName">

    <parameter name="einBegriff" value="//Person"/>

</UIOpenScriptViewer>
```

Öffnet ein Fenster mit der Ausgabe des Skriptes 'EditorScriptName'. Dieses muss im Begriffseditor bei "Zusätzliche Editoren" eingerichtet werden: Für Individuen im Abschnitt "Individuen", für Begriffe und Objektlisten im Abschnitt "Unterbegriffe".Die in parameter-Tags übergebenen Parameter sind als Variablen in dem Skript zugreifbar.

```
<UICreateTopic/>
```

Objektliste: Ein neuer Topic samt Eigenschaften für die Suchfeldeingaben wird angelegt.

Editor: Keine Wirkung, liefert null.

```
<UIOpenTopicList title="Personen" objects="//Person/instances()"/>
```

Öffnet ein Objektlistenfenster mit den durch das Argument *objects* festgelegten Topics.

Variablen

Außerdem kann ein Skript auf vordefinierte Variablen zugreifen.

Editor:

- **selectedTopic** - das bearbeitete Objekt
- **concept** - der Begriff des Objektes. Falls das Objekt ein Begriff ist, ist es der Begriff selbst

Objektliste:

- **selectedTopic** - das ausgewählte Objekt. nil, falls kein oder mehrere Objekte ausgewählt wurden
- **selectedTopics** - die ausgewählten Objekte
- **concept** - der Begriff der Objektliste

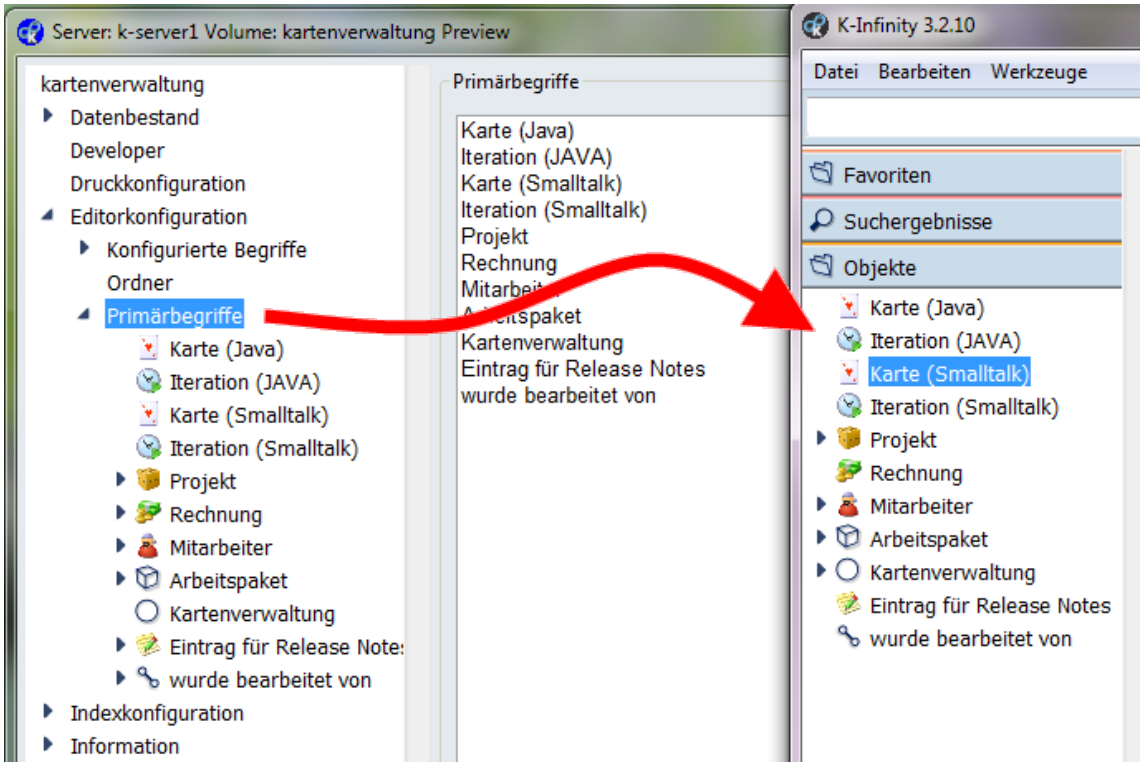
3.4 Konfiguration des Feeders

Der K-Infinity Feeder (k-feeder.exe) ist ein auf die Datenpflege spezialisierter, "abgespeckter" Knowledge-Builder. Im Gegensatz zum Knowledge-Builder sind dort keine Werkzeuge zu Ansicht und Bearbeitung des Wissensnetzschemas zugänglich. Stattdessen verfügt der Feeder über zusätzliche Konfigurierbarkeit bezüglich der Präsentation der Wissensnetzinhalte.

Die Konfiguration des Feeders erfolgt über ein spezialisiertes Admin-Tool (admin_k-feeder.exe). Dort finden sich im Bereich "Editorkonfiguration" zusätzliche Elemente, die im folgenden beschrieben werden.

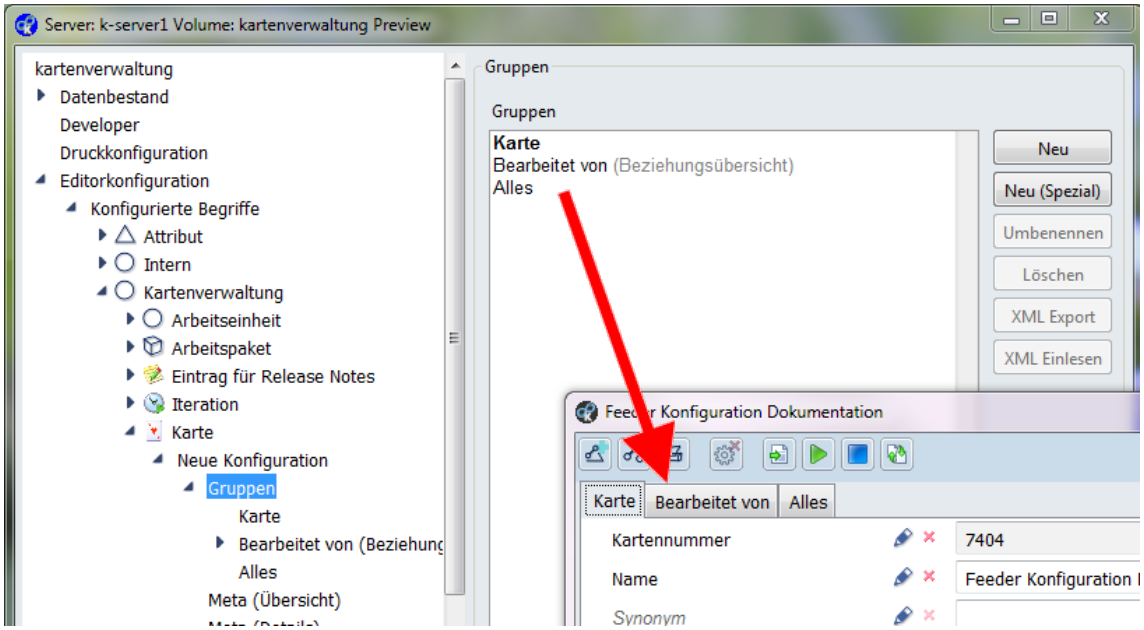
3.4.1 Primärbegriffe

Die sogenannten Primärbegriffe, sind die Begriffe, die im K-Feeder auf oberster Ebene zur Ansicht/Bearbeitung angeboten werden. Sie werden im Baum unterhalb des Ordners "Objekte" angezeigt. Das folgende Bild illustriert den Zusammenhang zwischen Konfiguration (links) und entsprechender Darstellung im K-Feeder (rechts).



3.4.2 Konfiguration der Gruppen

Über den Konigurationsbereich "Gruppen" kann das Erscheinungsbild des Topic-Editors für einen konfigurierten Begriff verändert werden. Im Gegensatz zum Knowledge-Builder ist der Topic-Editor des K-Feeders immer in Bereiche (=Gruppen) gegliedert, die als Tabs/Reiter visualisiert werden. Das folgende Bild illustriert den Zusammenhang zwischen Konfiguration (links oben) und Resultat im K-Feeder (unten rechts) für den Begriff "Karte".



Der K-Feeder kennt "normale" Gruppen, die aus einer definierten Folge von Objekteigenschaften (Attribute, Relationen, Erweiterungen) bestehen und spezialisierte Gruppen, die eine auf die Eigenschaftsart bzw. Benutzungsart zugeschnittene Sicht auf das Objekt ermöglichen. Eine Beschreibung aller Gruppenarten folgt in den folgenden Unterkapiteln.

Jede Gruppenart verfügt über "Optionen", "Zugriffsrechte" und "Übersetzungen". Über "Zugriffsrechte" lässt sich steuern, für wen bzw. wann die jeweilige Gruppe zur Anzeige kommt. (Die Anzeige der Daten selbst unterliegt dann natürlich der üblichen Rechteprüfung, die man an dieser Stelle nicht konfigurieren kann). Die Konfiguration der "Übersetzungen" ermöglicht die Konfiguration übersetzter Gruppennamen. Je nach eingestellter Sprache des Betriebssystems werden im K-Feeder dann auf den Tabs/Reitern die jeweilige Übersetzung des Gruppennamens angezeigt.

Unter den Optionen finden sich folgende Einträge:

- **Standard** Konfiguriert, ob diese Gruppe beim Öffnen des Editors automatisch angewählt ist. Diese Gruppe ist in der Konfigurationsübersich fett dargestellt.
- **Standard (neue Objekte)** Konfiguriert, ob diese Gruppe bei neu erstellten Objekte automatisch angewählt ist.
- **Alle verfügbaren Attribute anzeigen** Alle in dieser Gruppe konfigurierten Attribute werden angezeigt, auch wenn diese nicht ausgefüllt sind. Diese Option wirkt nur bei Gruppenarten, die Attribute anzeigen.
- **Alle verfügbaren Relationen anzeigen** Alle in dieser Gruppe konfigurierten Relationen werden angezeigt, auch wenn diese nicht ausgefüllt sind. Diese Option wirkt nur bei Gruppenarten, die Relationen anzeigen.
- **Mehrfachvorkommen anzeigen** Diese Option wirkt in Kombination mit den beiden vorangegangenen Optionen und gibt an, ob mögliche Eigenschaften auch dann angezeigt werden, wenn die jeweilige Eigenschaftsart bereits ausgefüllt vorliegt.
- **Eigenschaften können bearbeitet werden** Diese Option ermöglicht es, alle Eigenschaften der Gruppe auf "nur lesend" zu schalten. Auch hier gilt: Bearbeitungsverbote durch die Zugriffsrechtekonfiguration können durch diese Option nicht aufgehoben werden.

- **Gruppierende Element automatisch aufklappen** Über diese Option lässt sich einstellen, dass alle gruppierenden UI-Elemente (z.B. eine Liste von vielen Relationen desselben Typs) beim Öffnen des Editors automatisch aufgeklappt dargestellt werden.
- **Nur zum Drucken verwenden** Markiert die Gruppe als "unsichtbar" für die Darstellung im Editor. Bei der Verwendung einer speziellen Druckkomponente (nicht Bestandteil des K-Feeders) werden diese Gruppen zum Drucken angeboten.

3.4.2.1 Gruppe mit Eigenschaften

Eine Gruppe mit Eigenschaften zeigt ausgewählte Eigenschaften eines Objekts.

Über die "Eigenschaften"-Konfiguration dieser Gruppe steuert man, welche Eigenschaften zur Anzeige kommen und in welcher Reihenfolge. Aus der Menge aller verfügbaren Eigenschaften im linken Bereich lassen sich über die Schaltfächen "$>$" und "$<$" im rechten Bereich die für diesen Reiter gewählten zusammenstellen. Dabei gibt es folgende Besonderheiten:

- **Abkürzungsrelationen** können wir normale Relationen eingeblendet werden, wenn auch im Editor nur lesend.
- **Erweiterungen** werden mittels der Eigenschaft "hat Erweiterung" eingeblendet. Dabei ist zu beachten, dass der K-Feeder Erweiterungen wie Auswahlattribute darstellt.
- Die Spezialelemente "**Sonstige Attribute**", "**Sonstige Relationen**" und "**Sonstige Eigenschaften**" stehen als Platzhalter für alle Attribute/Relationen/Eigenschaften des Objekts, die nicht speziell konfiguriert wurden. Eine Konfiguration beispielsweise bestehend aus "Name" und "Sonstige Attribute" würde im Editor zunächst den Namen und dann alle anderen Attribute anzeigen.

Für jede gewählte Eigenschaft lässt sich separat einstellen, ob diese nur angezeigt werden soll, wenn sie ausgefüllt ist (**Existierende anzeigen**), wenn sie nicht ausgefüllt ist (**Wenn nicht vorhanden, zusätzliche anzeigen**), oder immer (**Zusätzliche anzeigen**).

Die Option "**Eigenschaft in einer Tabelle auflisten**" befindet sich noch im Experimentalstadium.

3.4.2.2 Beziehungsübersicht

Die Gruppe "Beziehungsübersicht" stellt ausgewählte Relationen des Objekts tabellarisch dar. Dabei sind die Einträge/Zeilen der Tabelle alle Relationen, die über die Konfiguration "**Filter**" gewählt wurden. Hinweis: Wenn kein Filter konfiguriert ist, werden alle Benutzerrelationen angezeigt.

In der Konfiguration "**Tabelle**" werden die Spalten der tabellarischen Ansicht analog zu einer Objektliste konfiguriert. Zusätzlich zu den normalen Eigenschaften der Relation, wie sie in jeder Tabellenkonfiguration zur Verfügung stehen, erlauben **virtuelle Eigenschaften** die Anzeige der Relationsquelle, des Relationsziels, dem Typ von Quelle, Ziel und der Relation.

Die Konfiguration "**Sortierung**" erlaubt das Setzen der initial zu sortierenden Spalte.

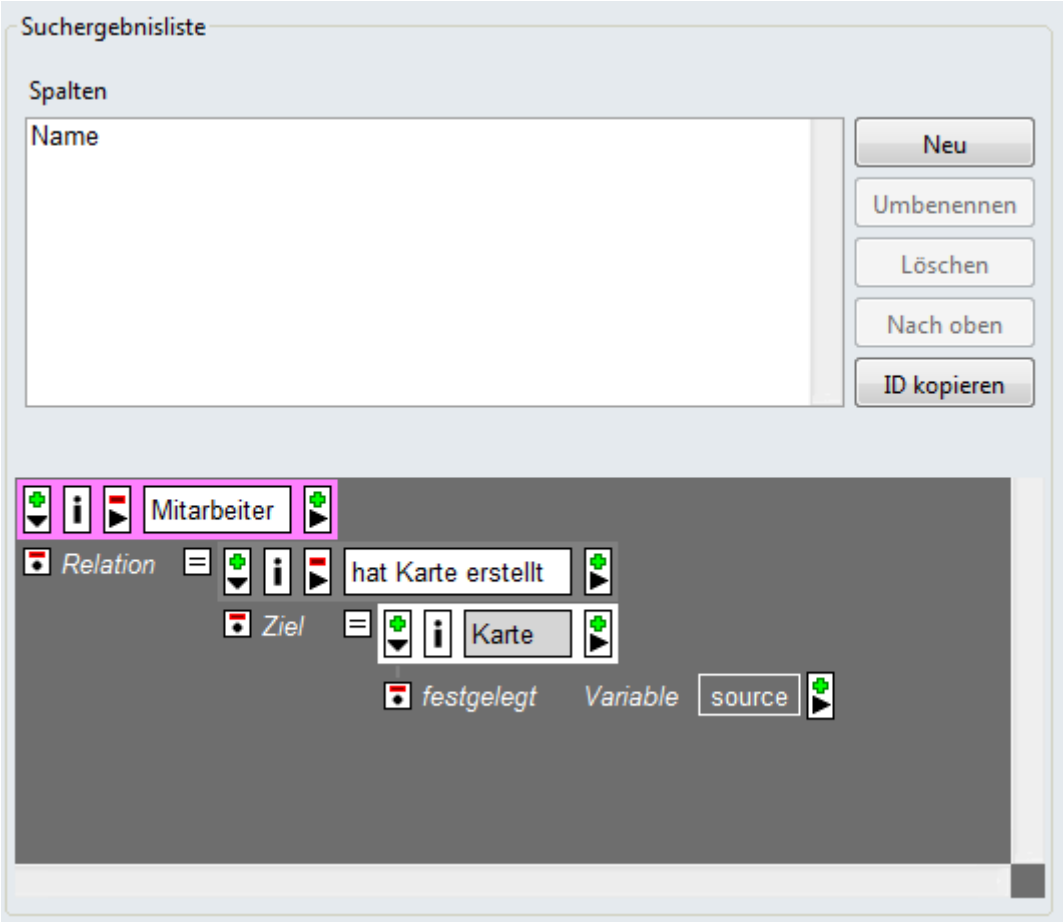
3.4.2.3 HTML

Die Gruppe "HTML" zeigt eine konfigurierte HTML-Ansicht des Objekts. Als **Skript** zur Auswahl stehen alle HTML-Skripte, die im Knowledge-Builder auf den Reiter "zusätzliche Editoren" konfiguriert wurden.

3.4.2.4 Suchergebnisliste

Die Suchergebnisliste zeigt ein Expertensuchergebnis in Tabellenform an. Die Suche kann in der Konfiguration definiert werden. Das aktuelle Objekt kann dabei als Parameter source an die Expertensuche übergeben werden.

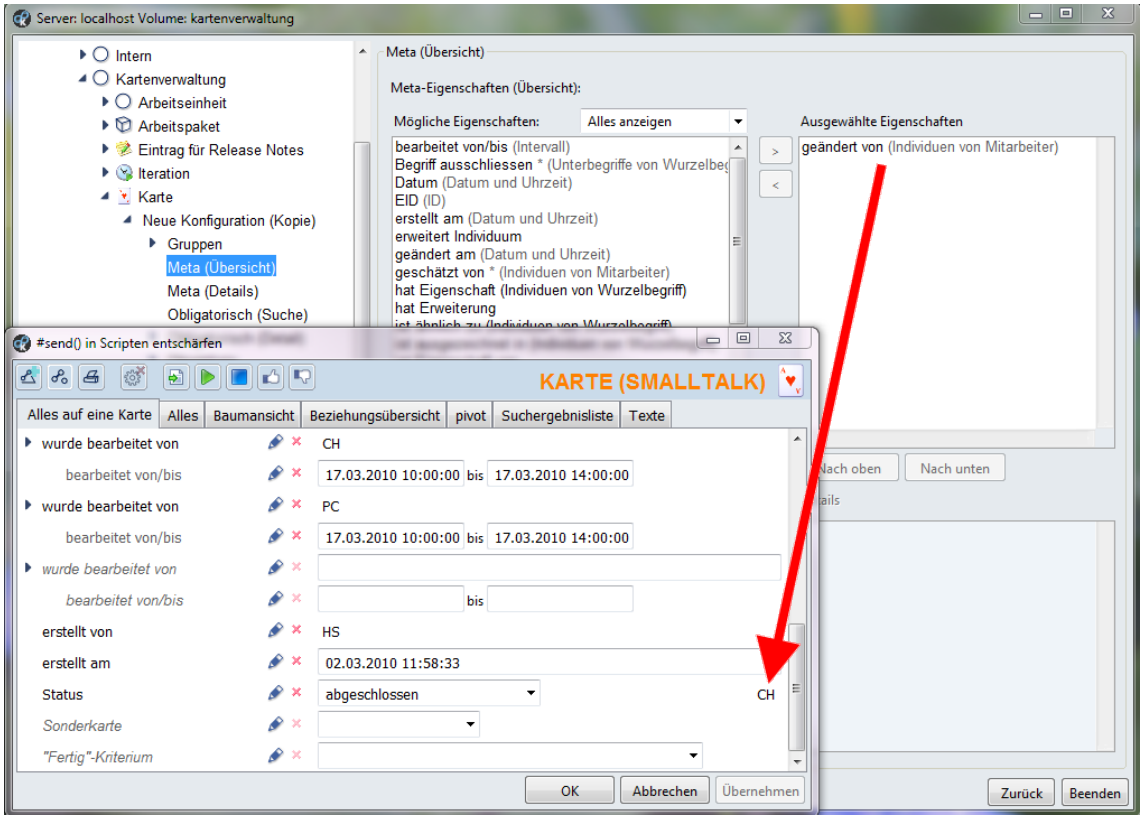
das Vorgehen bei der Konfiguration der Tabelle entspricht dem der gewöhnlichen Objektliste.



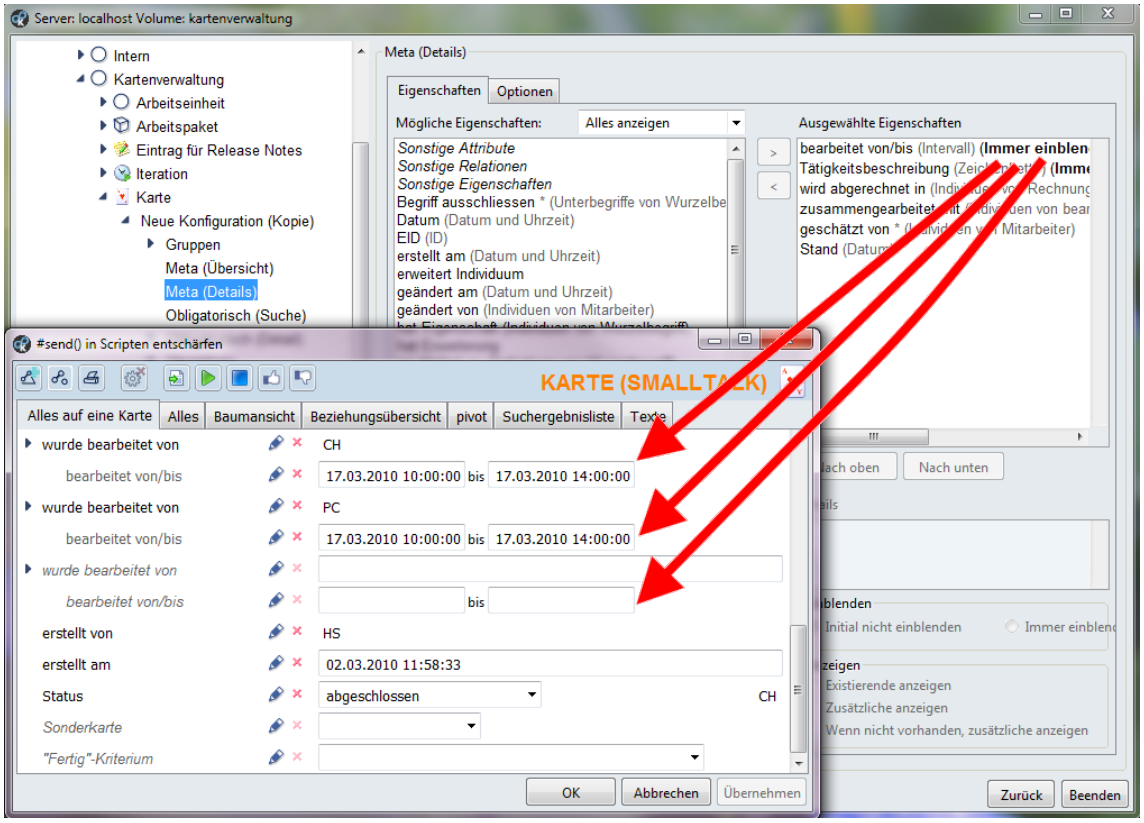
Beispielkonfiguration

3.4.3 Metaeigenschaften

Über die Konfigurationselemente "**Meta (Übersicht)**" und "**Meta (Details)**" lassen sich Art und Reihenfolge der angezeigten Metaeigenschaften einstellen. Der "Übersicht"-Bereich befindet sich im Topic-Editor rechts von der zugehörigen Eigenschaft, der "Detail"-Bereich ist eine eingerückte Liste unter der zugehörigen Eigenschaft. Die Konfigurationen wirken für alle Gruppen mit Eigenschaften gleichermaßen.



Beispiel für die Konfiguration von "Meta (Übersicht)"



Beispiel für die Konfiguration von "Meta (Details)"

3.4.4 Pflichtfelder für die Suche

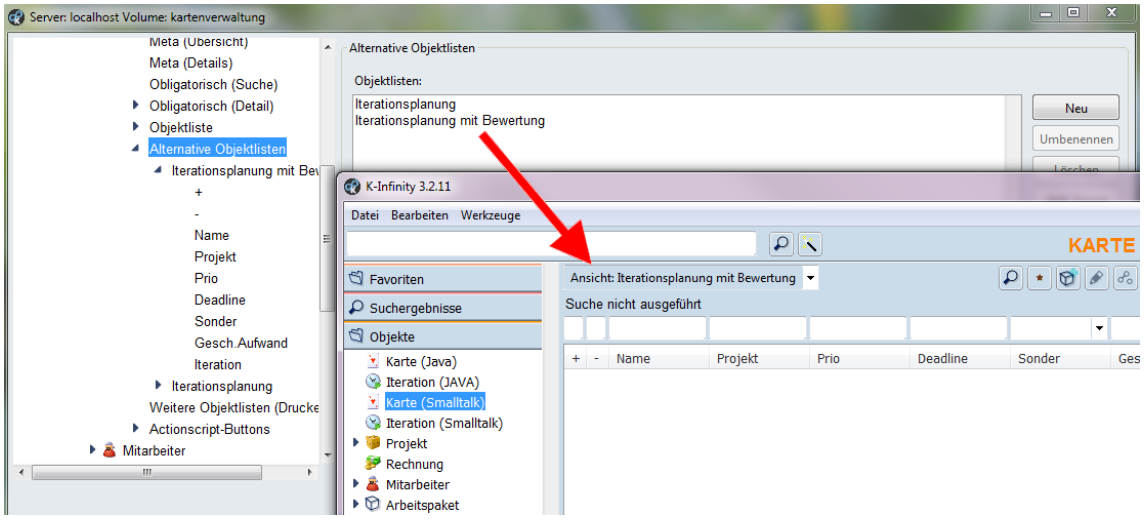
Die Konfiguration "Obligatorisch (Suche)" ist veraltet und wurde durch einen Schalter in der Konfiguration von Objektlistenspalten ersetzt.

3.4.5 Obligatorische Eigenschaften

Im Konfigurationsbereich "**Obligatorisch (Detail)**" können für jeden konfigurierten Objekttyp Pflichtfelder festgelegt werden. Die Konfiguration sieht mehrere Pflichtfeld-Gruppen vor. Ein Objekt kann nur dann gespeichert werden, wenn es für mindestens eine Pflichtfeld-Gruppe alle konfigurierten Eigenschaften besitzt. Fehlende Eigenschaften werden im Editor rot markiert.

3.4.6 Alternative Objektlisten

Über die Konfiguration "Alternative Objektlisten" lassen sich zusätzliche Objektlisten für einen Objekttyp zusammenstellen. Existieren solche Alternativen, dann werden sie im K-Feeder in einem "Drop-Down" zur Auswahl angeboten.



Beispiel für alternative Objektlisten: Konfiguration im Admin-Tool oben links und Resultat im K-Feeder unten rechts

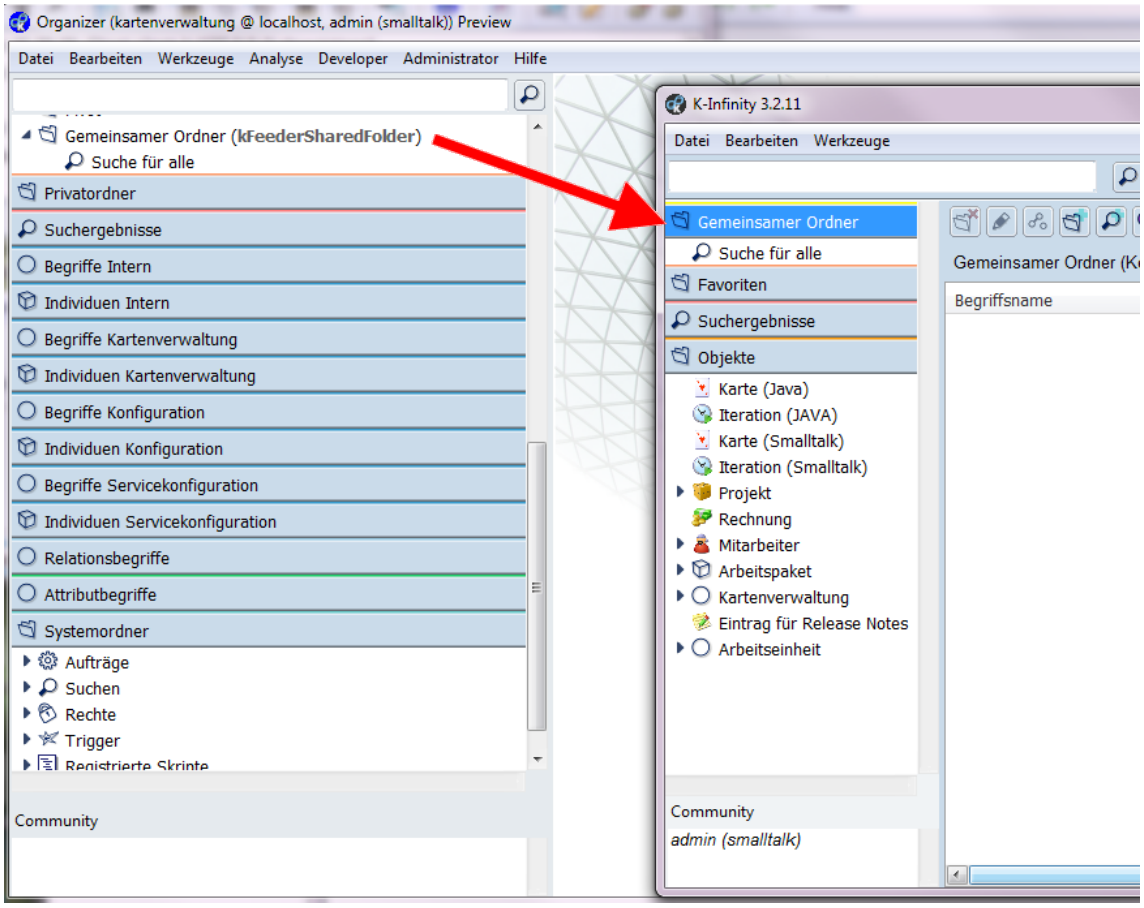
3.4.7 Relationszielwahl

Der K-Feeder verwendet standardmäßig einen aufwändigen tabellarischen Dialog zur Wahl eines Relationsziels. Alternativ steht eine einfache Relationszielwahl in einem übersichtlicheren und schlankeren Dialog zur Verfügung. Über das Attribut „**einfache Zielwahl**“ (interner Name: „**showSimpleTargetList**“) an Unterbegriffen von „Benutzerrelation“ kann man im Knowledge-Builder einstellen, dass für die entsprechende Relation und deren Unterrelationen in der Vererbungshierarchie der einfachere Dialog verwendet werden soll.

3.4.8 Arbeitsordner

Standardmäßg kennt der K-Feeder im Gegensatz zum Knowledge-Builder keinen gemeinsamen Arbeitsordner. Um die geteilten Ordner im Arbeitsordner der Wissensnetzadministratoren nicht frei zugänglich zu machen, werden diese im K-Feeder nicht eingeblendet.

Um einen gemeinsamen Ordner für den K-Feeder einzurichten, erstellt man in Knowledge-Builder einen neuen Ordner und setzt diesem die ID "**kFeederSharedFolder**".



Beispiel für die Konfiguration eines geteilten Ordners, hier mit dem Namen "Gemeinsamer Ordner"

4 Indexkonfiguration

4.1 Expertensuche optimieren

In diesem Bereich finden sich Einstellungen zur Optimierung der Expertensuche durch Aktivierung/Deaktivierung geeigneter Indizes. Es wird ein Überblick über alle aktuellen Einstellungen gegeben und ggf. Empfehlungen zur Aktivierung oder Deaktiverung von Indizes gegeben. Ein grüner Haken bedeutet: die Einstellungen sind in Ordnung, ein roter Blitz: die Einstellungen müssen geprüft und ggf. korrigiert werden.

Achtung: Es ist nicht immer notwendig, einen Index zu aktivieren, auch wenn das System dies vorschlägt. Ein Index sollte nur aktiviert werden, wenn nach der entsprechenden Eigenschaft häufig gesucht wird, denn Indexstrukturen benötigen Speicherplatz und Rechenzeit bei der

Pflege.

Damit man nicht immer wieder vom System gewarnt wird, einen bestimmten Index zu aktivieren, kann man den entsprechenden Eintrag mit „ignorieren“ markieren.

Den selektierten Eigenschaften kann mit den Schaltflächen "Hinzufügen" einen Index zuordnen. Analog werden Zurodnungen der Indexer über "Entfernen" wieder aufgehoben. Ebenso kann per Schaltfläche den selektierten Eigenschaften der Status ignorieren gesetzt bzw zurückgesetzt werden.

Um eine bessere Übersicht über die zu überprüfenden Eigenschaften zu bekommen, können mit den Checkboxen "ausreichend indexierte ausblenden" bzw "ignorierte ausblenden" die Liste der Eigenschaften entsprechend gefiltert werden.

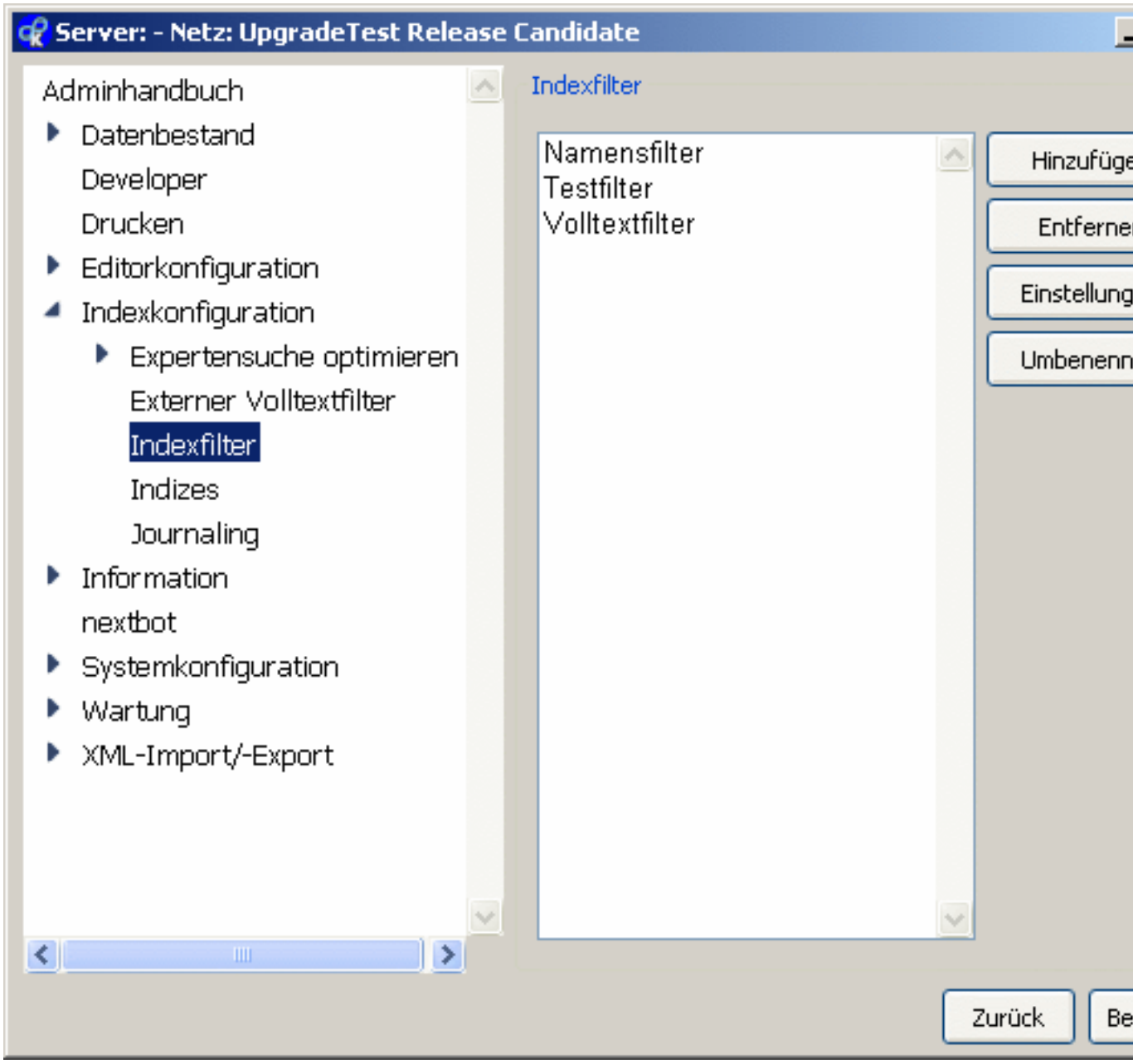
Beim Abschnitt "Index für Relationen" können, basierend auf der Selektion, die inversen Relationen selektiert werden (Shift halten, um dies mit der bestehenden Selektion zu verbinden). Die Spalte Indexrichtung zeigt an, in welche Richtungen die Relation gelöst werden kann. Die Pfeile verlaufen von der Quelle links zum Ziel rechts. Es werden hierbei auch die Indexer der inversen Relationen berücksichtigt, es kann also eine Relation durchaus in beide Richtungen gelöst werden ("<->") , ohne dass sie selbst indexiert wird.

Hinweis: Bei manchen Attributen (Geo und Intervall) werden zwei Indexer benhötigt (Start-/Stopwert bzw X/Y-Koordinate), um die Werte komplett zu erfassen.

4.2 Indexfilter

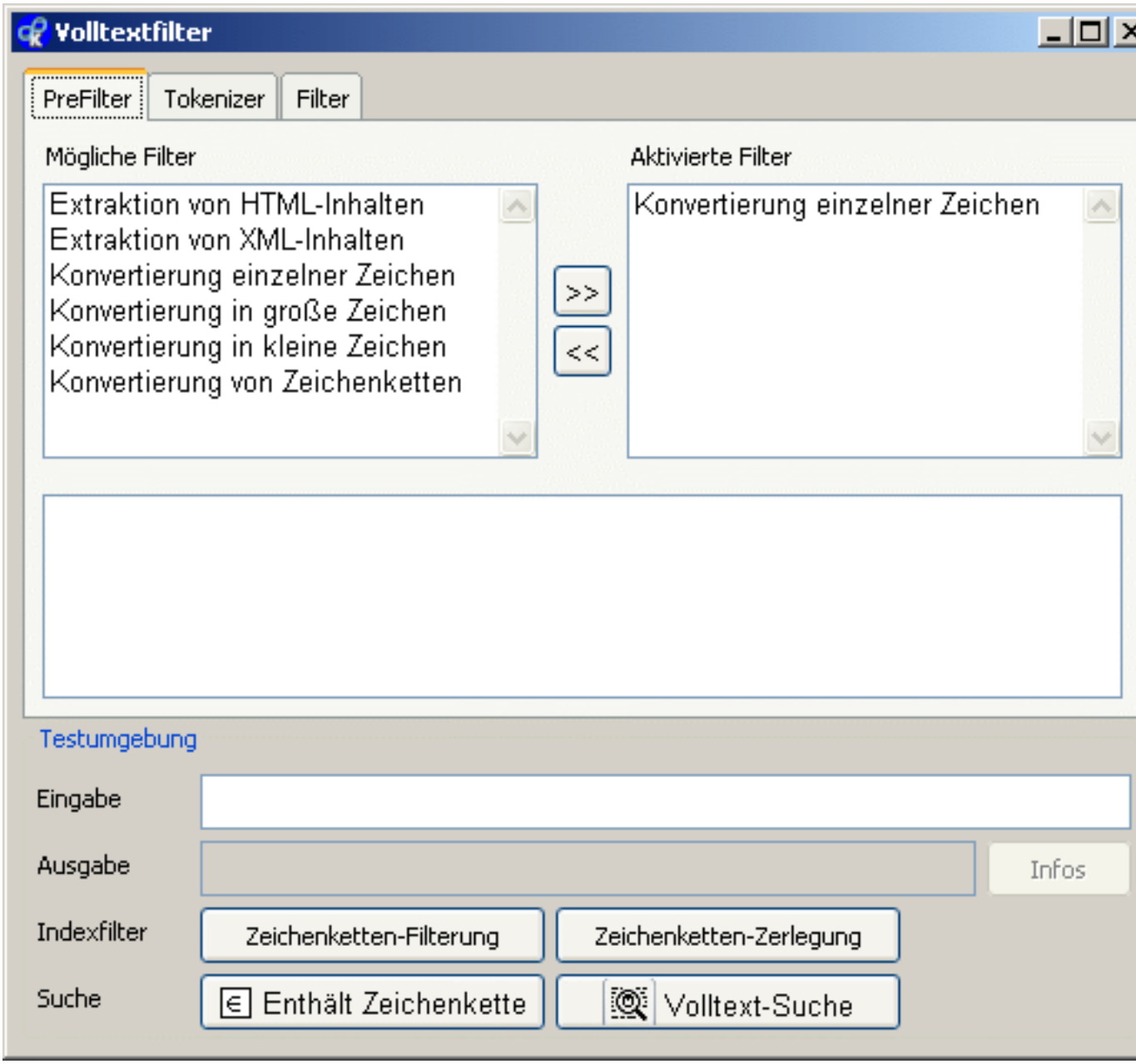
Indexfilter dienen dazu, den im Index zu speichernden Wert eines Attributes vor der Indexierung zu ändern oder die Indexierung zu verhinden. Zum Beispiel können häufig gebrauchte Stoppwörter wie „das“ ausgeschlossen oder die HTML-Entity „ö“ in das entsprechende Zeichen „ö“ umgewandelt werden.

Über die Indexerkonfiguration gelangt man zu den Indexfiltern.



Mit den Schaltflächen "Hinzufügen" und "Entfernen" kann ein neuer Indexfilter erstellt bzw. entfernt werden. Der Name eines neuen Indexfilters ist frei wählbar und kann nachträglich mit der Schaltfläche "Umbenennen" angepasst werden.

Mit der Auswahl des zu bearbeitenden Indexfilter und dem Anklicken des Knopfes „Einstellungen“ kommt man zum Filterbearbeitungs-Fenster:



Der obere Bereich zeigt zunächst die Konfiguration des PreFilters, über den Reiter können auch Tokenizer und Filter angezeigt und bearbeitet werden. Die Testumgebung im unteren Bereich ist unabhänig vom aktuell gewählten Reiter. Zum überprfüfen des Indexfilters kann man hinter Eingabe die zu überprüfende Zeichenkette eingeben. Je nach Schaltfläche wird das Ergebnis hinter Ausgabe angezeigt.

Die Schaltflächen **Zeichenketten-Filterung** und **Zeichenketten-Zerlegung** überprüfen, welche Zeichenketten ein zusammensteckbaren Indexer mit jeweiligem Indexfilter tatsächlich indexieren würde. Die **Zeichenketten-Filterung** verwendet nur den PreFilter, mit einer **Zeichenketten-Zerlegung** werden zusätzlich Tokenizer und Filter genutzt, der entsprechend konfigurierte Indexer wird so zu einem Volltextindexer. Dabei sind die einzelnen Tokens von {} umklammert.

Zusätzlich kann man überprüfen, wie der Expertensuch-Operator **Enthält Zeichenkette** die

Sucheingabe verarbeitet bzw wie die **Volltext-Suche** die Sucheingabe zerlegt.

Anmerkung: Wenn der Indexfilter bereits einem Indexer zugeordnet wurde, so kann man ihn entweder rein lesend öffnen oder der Indexer wird markiert, da er nun wieder Synchronisiert werden muss.

4.2.1 Der Filtervorgang

Der Filtervorgang lässt sich in drei Teile unterteilen. Im ersten Teil wird die zu indexierende Zeichenkette durch die PreFilter bearbeitet. Dabei findet eine Ersetzung oder Änderung auf Zeichenebene statt. Danach folgt das Zerlegen in einzelne Token. Im dritten und letzten Schritt werden weitere Filter angewandt. Diese entfernen einzelne Token, fassen Token zusammen oder wandeln deren Zeichenketten um.

4.2.2 Allgemeines zu Filtern

Im linken Bereich werden alle Filter angezeigt, die zur Verfügung stehen. Durch die Schaltflächen $>>$ und $<<$ können die Filter zur bestehenden Konfiguration, im rechten Bereich zu sehen, hinzugefügt beziehungsweise entfernt werden. Da die Reihenfolge der Filter für die Funktionsweise relevant ist, können diese mit Hilfe des Kontextmenüs nach oben und unten verschoben werden.

Durch die Auswahl eines Filters in der Liste der aktivierten Filter wird im unteren Bereich, sofern eine Konfigurationsmöglichkeit besteht, diese dort angezeigt.

Die Filter geben mit dem ersten Wort ihrer Beschreibung darüber Auskunft nach welcher Funktionsweise sie verfahren:

- Extraktion nur im PreFilter-Vorgang einsetzbar
  Aus einer Zeichenkette wird eine neue Zeichenkette durch das Herausziehen von Elementen gebildet.
- Filterung nur im Filter-Vorgang einsetzbar
  Die Tokens, für die diese Aussage zutrifft werden, werden nicht in den Index aufgenommen.
- Konvertierung im PreFilter- also auch im Filter-Vorgang vorhanden
  Die Eingangszeichenkette oder der Text eines Tokens wird verändert.
- Zusammenfassung nur im Filter-Vorgang einsetzbar
  Einzelne Tokens werden zu einem Token zusammengefasst.

Die Benutzerschnittstelle für die Filter, die eine Konfigurationsmöglichkeit besitzen, ist bisher immer gleich, auch wenn der Filter mit der Einstellung anders verfährt. Im folgenden wird diese Oberfläche beschrieben.

Beispiel für die Konfigurationsmöglichkeiten eines Filters

	Hinzufügen von Worten oder Wortabhängigkeiten
	Laden der vollständigen Liste aus einer Datei im CSV-Format
	Speichern der gesamten Liste in eine Datei im CSV-Format
	Selektion oder komplette Liste löschen.

Anmerkung: Es wird nach dem ersten Buchstaben gruppiert

4.2.3 PreFilter

Die Eingangszeichenkette wird nacheinander in jeden PreFilter gesteckt, bearbeitet und am Ende dieses Filtervorgangs in einer Zeichenkette an den Tokenizer weitergereicht.

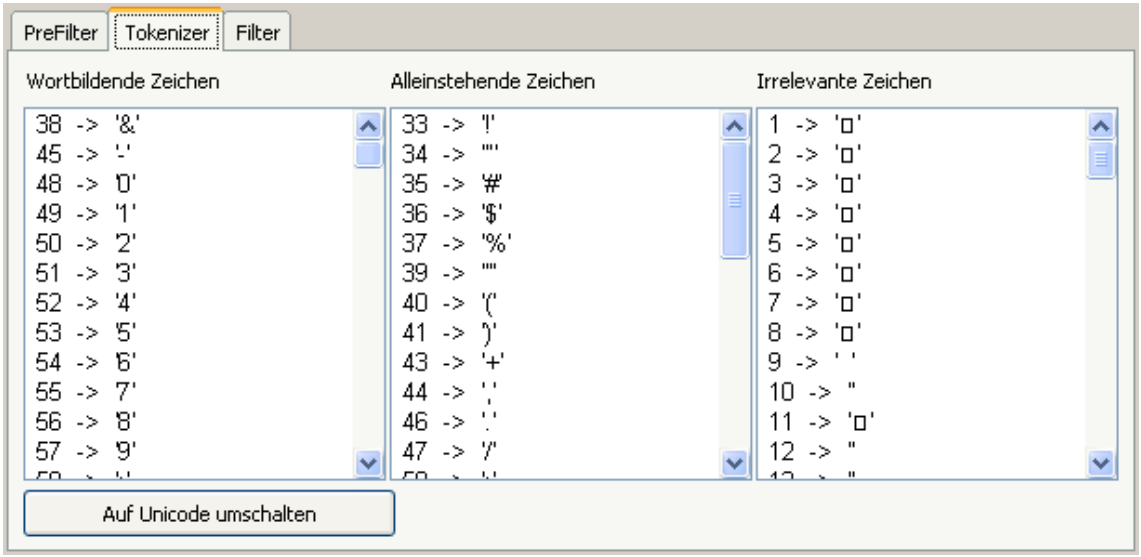
Mögliche PreFilter:

- **Extraktion von HTML-Inhalten**
  CharaterData und Attribut-Werte werden extrahiert, die durch den Attribut-Namen in der Konfiguration ausgewählt werden können. Bei der Ermittlung der relevanten Wörter für den Index wird der HTML-Parser verwendet.
- **Extraktion von XML-Inhalten**
  CharaterData und Attribut-Werte werden extrahiert, die durch den Attribut-Namen in

der Konfiguration ausgewählt werden können. Bei der Ermittlung der relevanten Wörter für den Index wird der XML-Parser verwendet.

- **Konvertierung einzelner Zeichen**
  einzelne Zeichen werden wie in der Konfiguration angegeben in andere Zeichen umwandelt, z.B. ä in ä
- **Konvertierung in kleine Zeichen**
  alle Zeichen werden in kleine Zeichen umgewandelt
- **Konvertierung von Zeichenketten**
  mehrere Zeichen können wie in der Konfiguration angegeben in andere Zeichen umgewandelt werden, z.B. ä in ae

4.2.4 Tokenizer



Tokenizerkonfiguration

Die Zeichenkette, die bereits durch den PreFilter gelaufen ist, wird durch den Tokenizer in einzelne Token zerlegt.

Die einzelnen Zeichen sind in eine der drei Gruppen einsortiert und lassen sich durch Drag und Drop einer anderen Gruppe zuordnen:

- **Wortbildende Zeichen**
  diese Zeichen bilden mit anderen Zeichen ein Token
- **Alleinstehende Zeichen**
  diese Zeichen bilden ein Token, das nur aus einem Zeichen besteht
- **Irrelevante Zeichen**
  diese Zeichen werden in keinem Token aufgenommen, sie trennen die voranstehenden und folgenden Zeichen in unterschiedliche Token auf.

Voreingestellt sind nur die Zeichen 1 bis 255 konfiguriert, man kann mit der Schaltfläche "Auf Unicode umschalten" auch alle Unicode-Zeichen konfigurieren. Ansonsten werden die Zeichen >255 als irrelevante Zeichen interpretiert.

Anmerkung: Einige Zeichen lassen sich nicht verschieben. Sie sind durch Filter (z.B. beim Filter

Zusammenfassung von XML-Tags sind < und > alleinstehende Zeichen) an eine bestimmte Gruppe gebunden.

4.2.5 Filter

In diesem Filtervorgang werden die Token aus dem Tokenizer kommend verarbeitet.

Mögliche Filter:

- **Extraktion von XML-Attributwerten**
  aus dem Text eines Tokens können ohne die Verwendung eines Parsers die Attribut-Werte extrahiert werden und als einzelne Token weiter verarbeitet werden. Die zu berücksichtigenden Attribut-Namen können in der Konfiguration eingestellt werden.
  Notwendiger Vorfilter: **Zusammenfassung von XML-Tags inkl. Attributen**
- **Filterung alleinstehender Zeichen**
  alleinstehende Zeichen jeglicher Art werden nicht in den Index aufgenommen
- **Filterung kurzer Zahlen**
  Zahlen die kürzer als drei Zeichen sind, werden nicht indexiert
- **Filterung von Stoppwörtern**
  die Token dessen Wort sich in der konfigurierten Wortliste befindet, werden nicht indexiert
- **Filterung von XML-End-Tags**
  XML-End-Tags werden nicht indexiert, z.B. </def>
  Notwendiger Vorfilter: **Zusammenfassung von XML-Tags** oder **Zusammenfassung von XML-Tags inkl. Attributen**
- **Filterung von XML-Tags**
  XML- -Tags werden nicht indexiert, z.B. <def>
  Notwendiger Vorfilter: **Zusammenfassung von XML-Tags** oder **Zusammenfassung von XML-Tags inkl. Attributen**
- **Filterung von Zahlen**
  Zahlen werden nicht indexiert
- **Konvertierung deutscher Worte auf Grundformen**
  es findet ein einfaches Wortstemming statt, ge wird vom Wortanfang entfernt, 'ch', 'ei', 'ie' und doppelte Zeichen bleiben erhalten, vom Ende werden 'e', 's', 'n', 't', 'em', 'er' und 'nd' entfernt, unter Berücksichtigung der Länge des Wortes, das übrig bleibt
- **Konvertierung in kleine Zeichen**
  alle Zeichen werden in kleine Zeichen umgewandelt
- **Konvertierung in kleine Zeichen, außer dem ersten**
  alle Zeichen außer dem ersten werden in kleine Zeichen umgewandelt.
- **Zusammenfassung eingebetteter SGML-Entities**
  ein gefundenes SGML-Entity wird mit den Token davor und dahinter zu einem neuen Token zusammengefasst, z.B. Wände
  Notwendiger Vorfilter: **Zusammenfassung von SGML-Entities**
- **Zusammenfassung von Abkürzungen**
  Abkürzungen, die in der Konfiguration eingegeben sind, werden zu einem Token zusammengefasst
- **Zusammenfassung von SGML-Entities**
  SGML-Entities werden zusammengeführt, z.B. ä

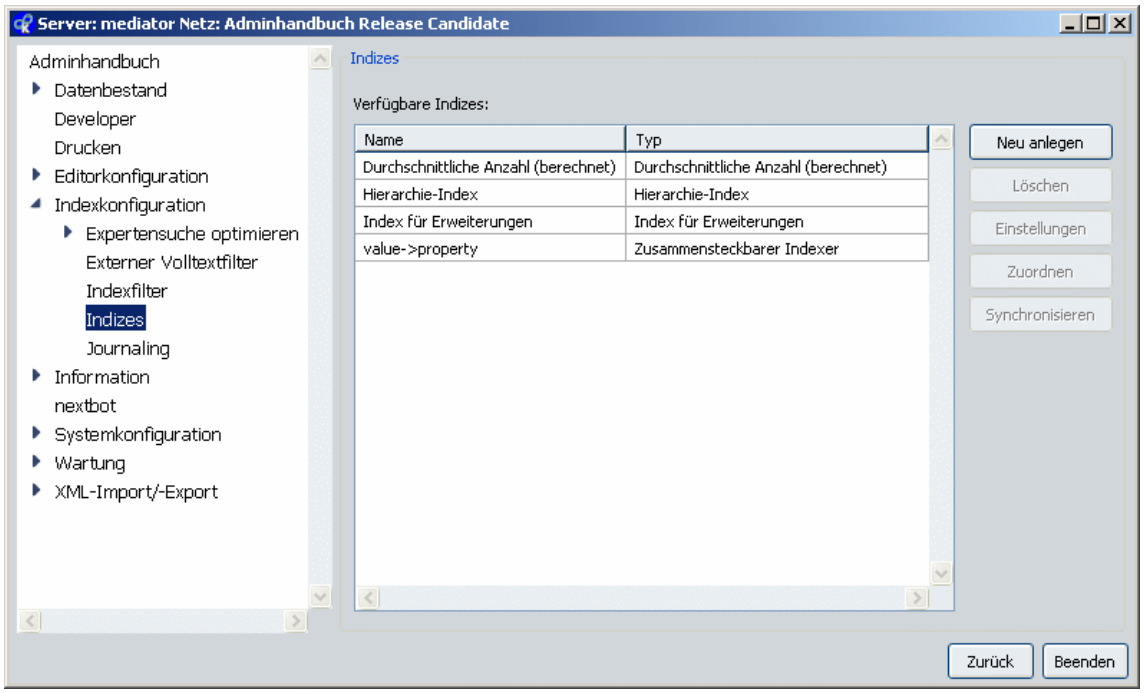
- **Zusammenfassung von XML-Tags**
  XML-Tags werden zusammengeführt, z.B. <art>
- **Zusammenfassung von XML-Tags inkl. Attributen**
  XML-Tags und ihre enthaltenen Attribute werden zusammengeführt, z.B. <art posit="mitte">

4.3 Indizes

Mit Hilfe dieser Einstellungsmöglichkeit lassen sich die Indexstrukturen verwalten. Unter "Verfügbare Indizes" werden alle verfügbaren Indextypen aufgeführt. Diese Liste kann sich bei projektspezifischen Admin-Tools von der hier aufgeführten Liste unterscheiden. Jeder Indextyp kann für bestimmte Arten von Attributen oder Relationen verwendet werden.

Falls ein Index grau dargestellt wird, ist der Index zur Zeit deaktiviert, ist er rot hervorgehoben, so ist der Index momentan nicht synchron.

Auf der rechten Seite befinden sich Schaltflächen zum Erzeugen, Löschen, Konfigurieren, Zuordnen und Synchronisieren.



Im folgenden wird die Konfiguration der **zusammensteckbaren Indexer** beschrieben, da diese am flexibelsten einsetzbar sind und nahezu jeden Einsatzbereich abdecken.

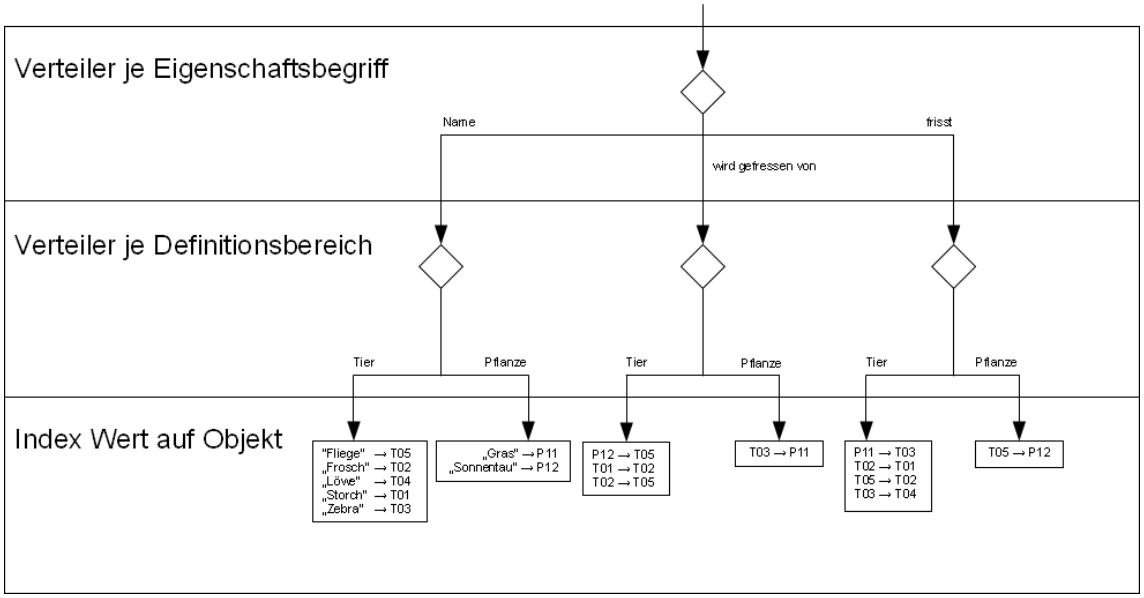
4.3.1 Zusammensteckbare Indizes

Zusammensteckbare Indizes erlauben es dem Administrator, einen Indexer aus vorgefertigten Bausteinen zusammenzustellen, um ein zugehöriges Verhalten des Indexers zu erreichen. Ein zusammensteckbarer Indexer besteht aus Verteilerstufen, die duch eine Index-Stufe abgeschlossen werden, welche die Datenspeicherung regelt. Dabei kann ein Indexer sowohl Attribute als auch Relationen indexieren. Wenn dem Indexer ein optionaler Indexfilter zugewiesen wird, so lässt sich das Indexerverhalten noch weiter beeinflussen, es können dann nur noch passende Eigenschaften-Typen dem Indexer zugeordnet werden.

Da Eigenschaften Attribute und Relationen umfassen, wird im folgenden ein Attributswert bzw. Relationsziel als Wert der Eigenschaft bezeichnet.

4.3.1.1 Indexerbausteine

Es wird unterschieden zwischen den aufschlüsselnden Indexerbausteinen und den indexierenden Indexerbausteinen. Ein aufschlüsselnder Indexerbaustein partitioniert den Index nach unterschiedlichen Aspekten. Dahinter folgt entweder eine weitere Aufschlüsselung, oder ein indexierender Indexerbaustein, welcher die Indexeinträge speichert.



Darstellung der Arbeitsweise

Die Abbildung zeigt exemplarisch, wie ein zusammensteckbarer Indexer aus drei Bausteinen (ohne Wertefilter) die Indexeinträge gruppiert. Dieser Index kann nun effizient beantworten

- Welche Tiere beginnen mit S
- Welche Pflanzen fressen andere Lebewesen
- Welche Tiere fressen Zebra (T03)
- u.s.w.

Fragestellungen wie zum Beispiel

- Welche Lebewesen beginnen mit S
- Welche Lebewesen fressen Fliegen (T05)

können ebenfalls beantwortet werden, hierzu wäre bereits eine Indexer-Konfiguration ohner **Verteiler je Definitionsbereich** ausreichend (und ist je nach Datenlage effizienter).

4.3.1.1.1 Verteiler je Eigenschaftsbegriff

Der wichtigste Baustein, ohne den die meisten indexierenden Bausteine nicht eingefügt werden können. Er sollte in der Regel an erster Stelle kommen und partitioniert die Einträge nach deren Eigenschaftsbegriff.

4.3.1.1.2 Verteiler je Definitionsbereich

Ermöglicht eine Partitionierung nach den jeweiligen Begriffen der eigenschaftstragenden Objekte. Der Baustein kann nur bei Eigenschaften an Individuen sinnvoll verbaut werden.

Kann eine Eigenschaft an mehreren Objekttypen vorkommen, und wird in einer Suche nur eine Teilmenge dieser Objekttypen gesucht, so beschleunigt dieser Baustein die Suche durch entsprechende Indexzugriffen.

4.3.1.1.3 Verteiler je Hauptbegriff

Analog zum Verteiler je Definitionsbereich wird je nach Hauptbegriff des eigenschaftstragenden Objektes partitioniert. Diese Aufteilung ist nur in Sonderfällen sinnvoll und dient hauptsächlich der Abbildung nicht mehr benötigter Indexstrukturen älterer K-Infinity-Versionen.

4.3.1.1.4 Verteiler je Objekt

Bei der Indexierung zum Zusammenfassen der Relationsziele am Quellobjekt kann dieser Baustein verwendet werden. Wie der vorherige Baustein dient er dem Abbilden älterer Indexer und ist sein K-Infinity 3.1 nur bei Einwegrückrelationen sinnvoll.

4.3.1.1.5 Verteiler je Eigenschaftswert

Dient zum Partitionieren nach Relationsziel bzw. nach Attributwert. Indexiert werden kann dann nur noch die Eigenschaft (siehe Index Eigenschaft).

4.3.1.1.6 Index Wert/Ziel auf Objekt

Mit diesem Indexbaustein werden Attributwert auf Objekt bzw. Relationsziel auf Relationsquelle im Index hinterlegt. Diese Indexierungsart ist dann sinnvoll, wenn Expertensuchen nach Objekten mit gegebenen Werten an den indexierten Attributen (bzw. mit gegebenen Zielen an den indexierten Relationen) unterstützt werden sollen.

4.3.1.1.7 Index Objekt auf Wert/Ziel

Dieser Indexbaustein indexiert genau umgekehrt wie der „Index Wert/Ziel auf Objekt“ und kann bei Attributen genutzt werden, um für Objektlisten die Spaltenwerte der indexierten Attribute zu ermitteln. Bei Relationen kann er genauso verwendet werden wie der „Index Wert/Ziel auf Objekt“, wenn entweder die inverse Relation indexiert ist oder das Quellobjekt durch die Suche bereits stärker eingeschränkt ist als das Zielobjekt.

Möchte man Expertensuchen mit der indexierten Relation in beide Richtungen (Quelle-Ziel und Ziel-Quelle) unterstützen, so kann die Relation entweder mit diesem und dem „Index Wert/Ziel auf Objekt“ indexiert werden, oder die Relation und ihre inverse Relation werden beide mit einer der beiden Index-Arten indexiert. Hierbei kann es eine Rolle spielen, ob der Indexbaustein mit einem „Verteiler je Definitionsbereich“ kombiniert ist, denn durch die Verwendung dieses Verteiler-Bausteines für einen Index auf der inversen Relation kann eine Partitionierung mittels der Zieldomain erreicht werden.

4.3.1.1.8 Index Wert/Ziel auf Eigenschaft

Mit diesem Indexbaustein werden Wert auf Attribut bzw. Ziel auf Relation im Index hinterlegt. Diese Indexierungsart ist dann sinnvoll, wenn an den indexierten Attributen und Relationen auch Suchen nach weitere Metaeigenschaften unterstützt werden sollen. Damit dieser Index in einer Suche auch für die Objekte der Eigenschaft (analog dem „Index Wert/Ziel auf

Objekt“) benutzt werden kann, muss die jeweilige Eigenschaft im zugehörigen Begriffseditor bei „Eigenschaft ist iterierbar“ auf „Aktiv“ bleiben.

4.3.1.1.9 Index Eigenschaft auf Wert/Ziel

Dieser Indexbaustein unterstützt Expertensuchen, bei denen Ziele der Relationen gesucht sind. Dafür muss die stärkste Einschränkung über Metaeigenschaften der Relation erfolgen. Einfache Quelle-Ziel Bedingungen werden jedoch nicht unterstützt.

4.3.1.1.10 Index Eigenschaft

Zusammen mit dem Verteiler je Eigenschaftswert kann das selbe Verhalten wie bei einem Index Wert/Ziel auf Eigenschaft erreicht werden. Bei sehr vielen gleichen Werten bzw. Zielen kann so eine kompaktere Speicherung erreicht werden, andernfalls bietet diese Kombination keine Vorteile.

4.3.1.1.11 Index Eigenschaftswert

Dieser Index speichert nur die Attributswerte bzw. die Relationsziele. Eine Verwendung ist dann sinnvoll, wenn ein „Verteiler je Objekt“ vorgeschaltet ist und wenige Objekte viele Werte/Ziele haben.

4.3.1.1.12 Index Redundante Speicherung für Relationseigenschaften

Dieser Baustein kann nur alleine verwendet werden und dient der schnelleren Anzeige von Metaeigenschaften an Relationen, wenn symmetrische Relationseigenschaften verwendet werden. Auf technischer Ebene wird keine Indexstruktur angelegt, der Indexer kann aber über die selben Konfigurations- und Programmschnittstellen angesprochen werden.

4.3.1.1.13 Eindeutigkeitsprüfung

Die Bausteine **Index Wert/Ziel auf Objekt** und **Index Wert/Ziel auf Eigenschaft** können um eine **Eindeutigkeitsprüfung** ergänzt werden. Üblicherweise werden auf diese Weise ergänzte Bausteine zur Konsistenzprüfung eindeutiger Kennzeichner verwendet. Sie stehen in der Auswahlliste der hinzufügbaren Indexbausteine zur Auswahl (z.B. Index Wert/Ziel auf Objekt (Eindeutigkeitsprüfung)).

Wenn ein neuer Wert geschrieben werden soll und einen gleichen Wert im Index vorfindet, so kann dieser neue Wert nicht übernommen werden. Werte werden dann als gleich erkannt, wenn sie auch von allen Verteilern des Indexes gleich gruppiert werden. Möchte man zum Beispiel eine Eindeutigkeitsprüfung nur je Domain (zum Beispiel erlaubt dies die Koexistenz von „modern“ als Individuum von Verb und als Individuum von Adjektiv), so muss ein **Verteiler je Definitionsbereich** im Index enthalten sein.

Wenn auch ein Wertefilter konfiguriert wird, so wird die Eindeutigkeitsprüfung auf den gefilterten Werten durchgeführt. So kann zum Beispiel „arm“ und „Arm“ als gleich erkannt werden.
Anmerkung: ein Wertefilter, der Zeichenketten zerlegt (für Volltext) kann zwar mit der Eindeutigkeitsprüfung kombiniert werden, dies erscheint in der Regeln nicht sinnvoll, da bereits eine Teilzeichenkette nach der Zerlegung zu einem Duplikat führen kann, zum Beispiel „Das Haus“ und „Haus und Hof“.

Der **Index Wert/Ziel auf Objekt** kann bei mehrfach vorkommenden Eigenschaften keine doppelten Werte dieser Eigenschaften an einem Objekt als Duplikate erkennen. Es könnten also zwei gleichartige Attribute mit gleichem Wert am selben Objekt existieren, nicht jedoch

an verschiedenen Objekten. Will man dies nicht erlauben, so muss am Attributbegriff das mehrfache Vorkommen deaktiviert werden oder stattdessen ein **Index Wert/Ziel auf Eigenschaft** zur Eindeutigkeitsprüfung verwendet werden.

4.3.1.2 Wertefilter

Eine Manipulation der zu Indexierenden Werte kann man durch einen geeigneten Wertefilter erreichen. Ein Indexer, an dem ein Wertefilter konfiguriert ist, kann nur noch für passende Attribute verwendet werden.

4.3.1.2.1 Wertzerlegung

Für Geokoordinaten und Intervall-Attribute kann kein atomarer Attributwert indexiert werden. Stattdessen wird mit Längengrad und Breitengrad bzw Intervall-Startwert und Intervall-Stopwert nur eine Komponente des Wertes indexiert. Für eine komplette Indexierung muss ein entsprechender Indexer für die jeweils andere Komponente des Wertes konfiguriert werden.

4.3.1.2.2 Zeichenketten-Manipulationen

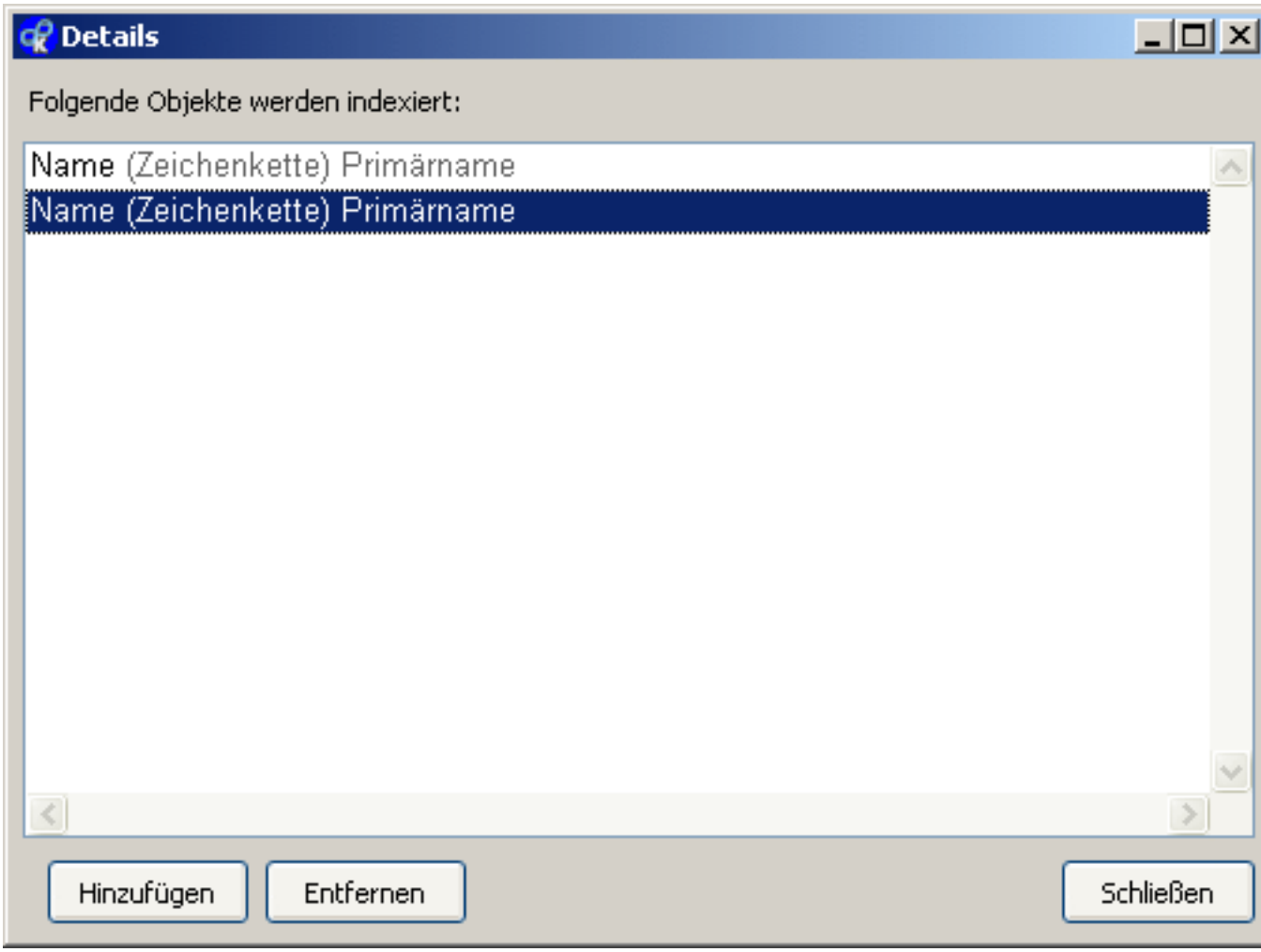
Für Zeichenketten können im Admintool Volltext-Filter konfiguriert werden. An diesen kann konfiguriert werden, welche Manipulation an den Zeichenketten vorgenommen werden und wie die Zeichenketten in einzelne Worte zerlegt werden sollen. In Expertensuchen werden dann zusätzliche Operatoren angeboten, die mit der jeweiligen Bezeichnung des Filters ergänzt sind, damit gezielt mittels dieses Filters gesucht werden kann.

Mittels einer „Zeichenketten-Filterung“ können die Zeichenketten in manipulierter Form indexiert werden, bei einer Suche werden dann alle Attributswerte als Treffer interpretiert, die durch den Filter auf die gleiche Zeichenkette abgebildet werden wie die Sucheingabe.

Mittels einer „Zeichenketten-Zerlegung“ können aus einem Text mehrere (manipulierte) Teilzeichenketten (Tokens) indexiert werden. Der zugehörige Index ermöglicht dann Expertensuchen, die mittels der Operatoren „Enthält Worte“ und „Enthält Phrase“ innerhalb der Zeichenketten suchen.

4.3.2 Zuordnen

In diesem Fenster ordnet man die Eigenschaften zu, die durch den Indexer indexiert werden.

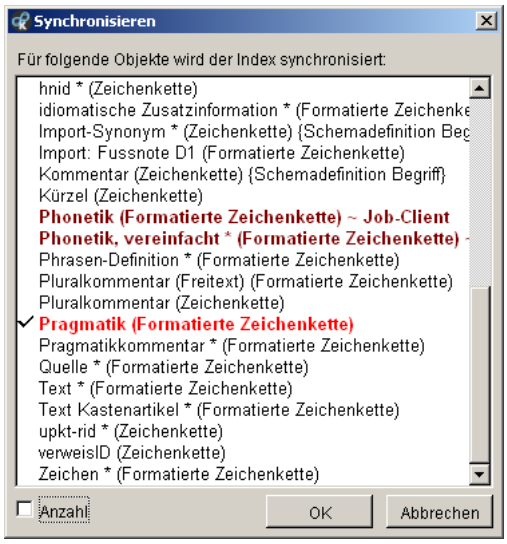


Die Liste "Folgende Objekte werden indexiert" zeigt die Attribut- und Relationsbegriffe, die im Index enthalten sind. Mit den Knöpfen „Hinzufügen“ und „Entfernen“ können Objekte zum Index hinzugefügt bzw. aus dem Index entfernt werden.

Nach dem Hinzufügen von Objekten muss eine Synchronisation für diese Objekte durchgeführt werden

Falls der Index synchronisiert werden muss, werden in der Objektliste die Objekte, die nicht vollständig indexiert sind, rot angezeigt.

4.3.3 Synchronisieren



Synchronisieren

Im ersten Schritt muss entschieden werden, ob die Synchronisation sofort ausgeführt („Jetzt durchführen“) oder ob ein System-Skript geschrieben werden soll („Skript schreiben“). Nach dem Speichern des System-Skripts kann dieses vom Administrator zu gegebener Zeit ausgeführt werden.

Im zweiten Schritt wird die Liste mit den zugeordneten Objekte angezeigt. Die Objekte, die synchronisiert werden müssen, sind rot gekennzeichnet. Sie können nun ausgewählt werden und der Indexvorgang kann mit OK gestartet werden.

Mit einem Häkchen bei Anzahl wird die Anzahl der Individuen pro Objekt angezeigt.

4.3.4 Volltextindexierung von Dateiinhalten

4.3.4.1 Arbeitsablauf zur Indexierung von Datei-Attributen

Für Volltextindexierung von Datei-Attributen gibt es die interne Variante im Wissensnetz und die externe Variante mittels Lucene-Index. Bei größeren Dokumentmengen (1GB+) empfehlen wir die Verwendung des Lucene-Index.

Die Konfiguration des Lucene-Index ist in einem separaten Abschnitt beschrieben. Der folgende Text bezieht sich ausschließlich auf die Wissensnetz-interne Variante.

Der Arbeitsablauf bei der Indexierung von Datei-Attributen über einen Volltextindex sieht wie folgt aus.

- Änderung des Datei-Attributs (auch beim Neuanlegen) wird per Trigger registriert
- Trigger generiert einen Job zu Extraktion des evtl. vorhanden Textinhalts des Datei-Attributs
- Extrahierter Textinhalt wird im Volltextindex eingetragen

Damit diese Einzelschritte durchlaufen werden können, sind einige Konfigurationsmaßnahmen wie folgt zu erledigen.

Einrichten des Triggers:

Die Trigger müssen aktiviert sein und für das betreffende Datei-Attribut muß ein Trigger eingefügt werden, welcher auf Modifikation des Attributs reagiert. Der Trigger muß nach der Ausführung der Modifikation folgendes Skript ausführen:

```
<Script>
<ExtractBlobText textAttribute="$textWithFulltextindex$"/>
</Script>
```

Einzustellender Operationsparameter ist "Eigenschaft". Der Parameter "textAttribute" des Skript-Tags "ExtractBlobText" gibt den internen Namen desjenigen (u.U. ebenfalls am Begriff des Topics noch anzulegenden) Textattributs an, in welches der extrahierte Text geschrieben wird. Auf diesem Textattribut muß der Volltextindex eingerichtet werden.

Einrichten des Jobclients zur Textextraktion:

Damit die Textextraktion durchgeführt werden kann, muß ein K-Infinity Jobclient gestartet werden, welcher in der Eigenschaft "jobPools" den "KExtractBlobTextJob" aufführt. Zusätzlich muß im Arbeitsverzeichnis des Jobclients das externe Shell-Skript "kfilter.bat" liegen, ebenso wie das Unterverzeichnis "filters", in welchem sich die für die Textextraktion benötigten Utilities befinden müssen. Zuletzt muß noch das Verzeichnis "temp" vorhanden oder zumindest anlegbar sein. Der Arbeitsablauf stellt sich hierbei wie folgt dar:

- Der Jobclient führt den Extraktionsjob aus. Dazu wurde ihm das Datei-Attribut übergeben
- Der Jobclient schreibt den Inhalt des Datei-Attributs in eine temporäre Datei im Verzeichnis "temp"
- Dann ruft er das Shell-Skript "kfilter.bat" auf. Dabei übergibt er den Namen der temporären Datei, den Namen der ebenfalls temporären Ziel-Textdatei sowie die Datei-Extension der zu extrahierenden Datei. Das Shell-Skript aktiviert anhand der Extension je nach Dateityp unterschiedliche Utilities, welche den Text extrahieren und in die Zieldatei schreiben.
- Die Zieldatei wird vom Jobclient gelesen und ihr Inhalt in das gewünschte Textattribut abgelegt. Der Inhalt wird danach indexiert.

Als Empfehlung, wenn absehbar ist, dass pro Topic nur ein Datei-Attribut angelegt wird (z.B. bei einer typischen Dokumentrepräsentation), kann man das Skript im Trigger wie folgt anpassen:

```
<Script>
<If test="size(@$textWithFulltextindex$) = 0">
<ExtractBlobText textAttribute="$textWithFulltextindex$"/>
</If>
</Script>
```

In diesem Fall findet die Textextraktion nur dann statt, wenn es offensichtlich vorher noch keinen (evtl. bereits extrahierten) Textinhalt gegeben hat. Auf diese Weise herrscht eine enge Kopplung zwischen dem Datei-Attribut und seinem Texthilfsattribut, so dass beide per Trigger zusammengelöscht werden können. Im vorher aufgezeigten Szenario wird für jede Änderung des Datei-Attributs die Textextraktion durchgeführt und auch ebensoviele Texthilfsattribute angelegt, so dass sich nicht mehr genau sagen läßt, welches von den vielen jetzt gültig ist.

Hinweis für die Installation auf Unix/Linux

Für den Betrieb des Jobclients zur Extraktion von Datei-Attribut-Inhalten muss gewährleistet

sein, daß die folgende Installationsstruktur eingehalten wird:

- Jobclient-Imagedatei, Jobclient.ini-Datei und kfilter.bat Shell-Skript-Datei in einem Verzeichnis
- filters als Unterverzeichnis dieses Arbeitsverzeichnisses enthält alle benötigten Utilities
- die PATH-Variable der Shell muss "." enthalten, da sonst das kfilter.bat-Skript nicht aufgerufen werden kann
- das Unterverzeichnis "temp" muss vorhanden oder zumindest anlegbar sein

4.3.4.2 Installation der Lucene-Indexierungsinfrastruktur

Textdokumente können per Lucene-Volltextindex durchsucht werden.

Dazu muss ein Jobclient eingerichtet werden, der die Indexierung und die Suche ausführt. Die Indextdateien des Lucene-Index müssen dem Job-Client zugänglich sein.

Weitere Informationen über Lucene findet man unter http://lucene.apache.org. Die Anbindung wurde mit den Versionen 2.2, 2.4, 2.9 und 3.1 getestet.

Auf dem Server, auf dem der Jobclient ausgeführt wird, muss eine Java Runtime-Umgebung (Version >= 5) installiert sein. Außerdem benötigt man drei Java-Pakete:

- lucene-core-<version>.jar
- lucene-analyzers-<version>.jar
- com/iviews/indexService/analysis/GermanXXXAnalyzer.class

Sämtliche Indexoperationen werden vom Jobclient durchgeführt.

4.3.4.2.1 Konfiguration des Jobclients

In der Konfigurationsdatei jobclient.ini ist der Abschnitt [JNI] für die Anbindung von Lucene anzupassen:

```
[JNI]
classPath=lucene\lucene-core-<version>.jar;lucene\lucene-analyzers-<version>.jar;lucene
;libraryPath=C:\Program Files (x86)\Java\jdk1.6.0_13\jre\bin\client\jvm.dll
```

"classPath" muss ein gültiger Java-Classpath sein. Der Classpath verweist auf die beiden JAR-Pakete sowie das Verzeichnis, in dem sich die Unterverzeichnisstruktur com/iviews./... befindet.

"libraryPath" kann normalerweise auskommentiert bleiben. Falls jedoch mehrere JVMs installiert sind, kann es evtl. notwendig sein, die passende JVM (jvm.dll) anzugeben.

Außerdem müssen folgende Jobs konfiguriert werden:

```
jobPools=KQueryJob,KLightweightIndexJob,KRemoveIndexJob,KAddAllToIndexJob,KSyncIndexJob,KLuceneAdm
```

KQueryJob	Suche im Volltextindex

KLightweightIndexJob	Inkrementelle Indexpflege
KAddAllToIndexJob	Indexaufbau
KRemoveIndexJob	Entfernen des Index
KLuceneAdminJob	Erstellen/Löschen/Optimieren von Volltextindizes
KSyncIndexJob	Indexneuaufbau

In der Konfigurationsdatei können auch weitere optionale Angaben gemacht werden:

```
[lucene]
; Index-Verzeichnis (überschreibt Konfiguration im Netz)
directory=lucene-index
; Von außen sichtbarer Hostname des Services
hostname=luceneserver
; Service port
port=8100
; Anzahl der Query-Server (Standard = 1)
numberOfServers=1
```

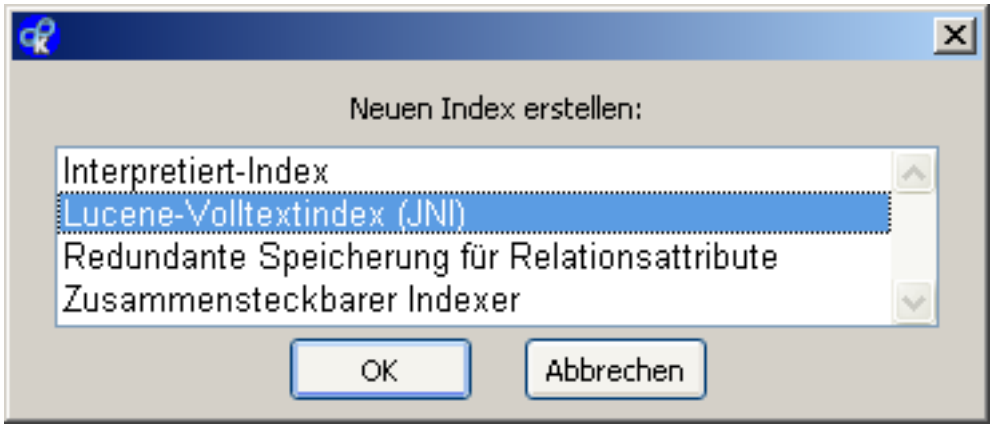
Für die Extraktion von Dateiinhalten muss außerdem die Verwendung von Apache-Tika eingerichtet werden:

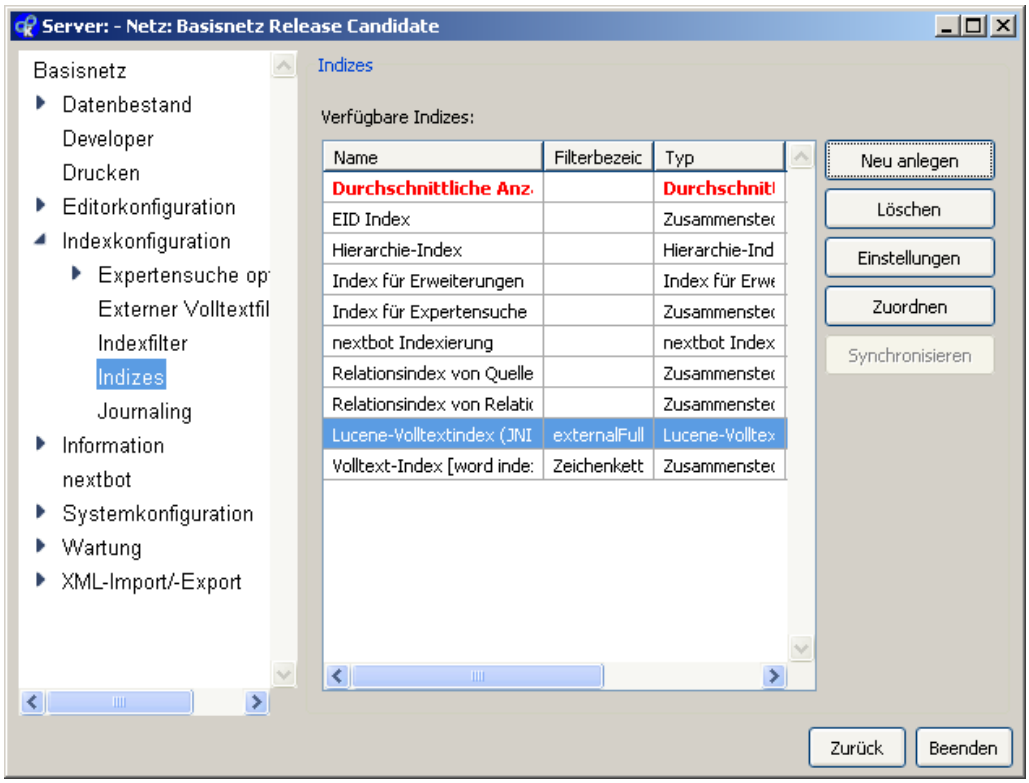
- Von der Webseite http://tika.apache.org/ die aktuelle tika-app (z.B. tika-app-1.3.jar) herunterladen und ins Verzeichnis des jobclients legen.
- Die jobclient.ini um folgenden Eintrag ergänzen:

```
[text-extraction]
tikaJavaParams=-Xmx1200M
tikaJarPath=tika-app-1.3.jar
; Optional: Java-Pfad
; extractorPath=java.exe
; Optional: Maximale Größe der Binärdateien,
;  für die eine Textextraktion durchgeführt wird
; extractedTextSizeLimit=100000
```

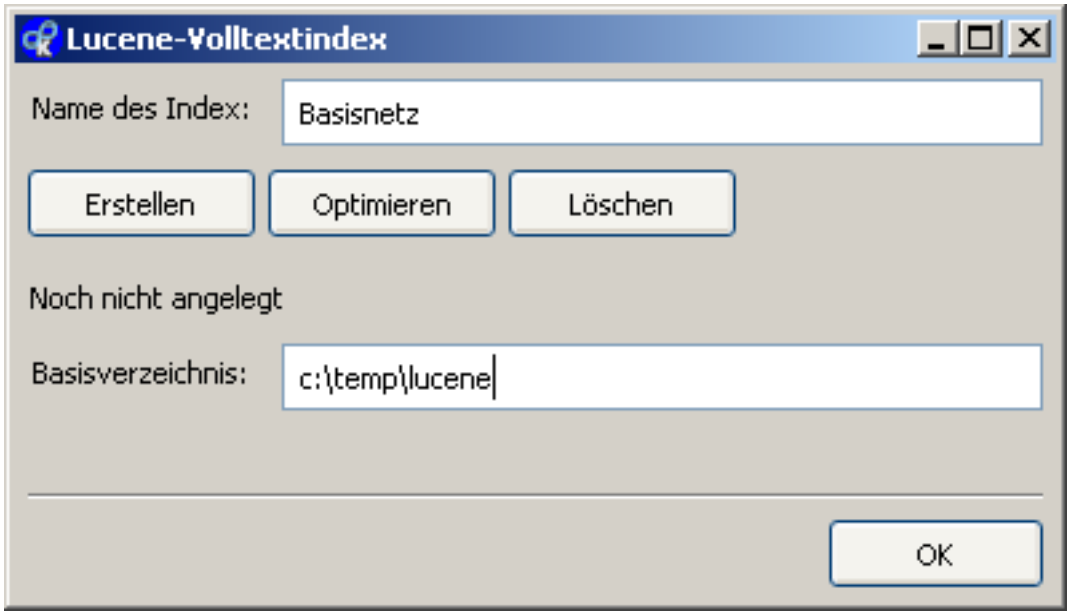
4.3.4.3 Konfiguration des Lucene-Volltextindex im Admin-Tool

Attribute, die volltextindexiert werden sollen, müssen im Admin-Tool zum Index "**Lucene-Volltextindex (JNI)**" hinzugefügt werden (mit Hilfe des Buttons "Zuordnen"). Dazu erst einen neuen Lucene-Volltextindex (JNI) anlegen (s.u.).





Der Button "Einstellungen" öffnet den Dialog zur Konfiguration des Lucene-Anschlusses. Hier kann man folgende Einstellungen/Aktionen durchführen:



- Basisverzeichnis: gibt das Verzeichnis an, in dem der Index gespeichert wird.
- Name des Index: ist der eindeutige Name des Lucene-Indexes. Wichtig: Falls Lucene von mehreren Wissensnetzen verwendet wird, muss unbedingt bei jedem Wissensnetz ein anderer Indexname verwendet werden.
- Erstellen: erzeugt den Index, falls er noch nicht vorhanden ist. Man wird nach dem Analyzer gefragt. Dieser ist dafür zuständig, einzelne Worte im Text zu erkennen und diese gegebenenfalls auf ihre Grundform abzubilden (Stemming). Man sollte hier "default" wählen, da "german" und "german_xxx" Stemming durchführen und sich dieses auch auf Personennamen auswirkt, was das Suchverhalten teilweise sehr schwer nachvollziehbar macht.
- Optimieren: sorgt dafür, dass der Index nach häufigen Änderungen wieder optimiert wird.
- Löschen: entfernt den Index, wobei z.Zt evtl. noch Dateien im Indexverzeichnis übrig bleiben, die von Hand nachträglich gelöscht werden müssen.

Alle obigen Operationen werden vom Jobclient ausgeführt, sollte dieser nicht gestartet sein, so wird darauf hingewiesen. Zur Konfiguration des Jobclients bitte das vorige Kapitel konsultieren.

Wenn ein netz mit konfiguriertem Volltextindex kopiert wird, muss danach ein neuer Volltextindex konfiguriert und aufgebaut werden. Ansonsten würden das Original und die Kopie auf den gleichen Index zugreifen, was zu fehlerhaftem Verhalten führt.

Die Volltextsuche wird in der Benutzungsoberfläche als "einfache Suche" angeboten. Hierfür muss im Organizer-Menü **Datei ->** **Einstellungen ->** **Suche** durch einen Administrator eine Suche "Volltext-suche (Lucene)" konfiguriert werden. Die einzelnen Benutzer können sich diese Suche dann - ebenfalls über den Menüpuntk **Einstellungen ->** **Suche** für das Suchfeld (Schreibtisch oben links) freischalten.

4.3.4.4 Metadatenextraktion

Mit der Metadatenextraktion können die Metadaten einer Datei, wie unter anderem Autor, Erstellungsdatum und Titel, extrahiert und in das Wissensnetz importiert werden. Die Metadaten können als Metaattribute des die Datei beinhaltenden Datei-Attributs oder als Attribute des Topics importiert werden. Voraussetzungen für den Import sind, dass das Datei-Attribut mit einem Lucene-Volltextindex indexiert wird und dass der Text per Tika-TextExtraktion volltextextrahiert wird.

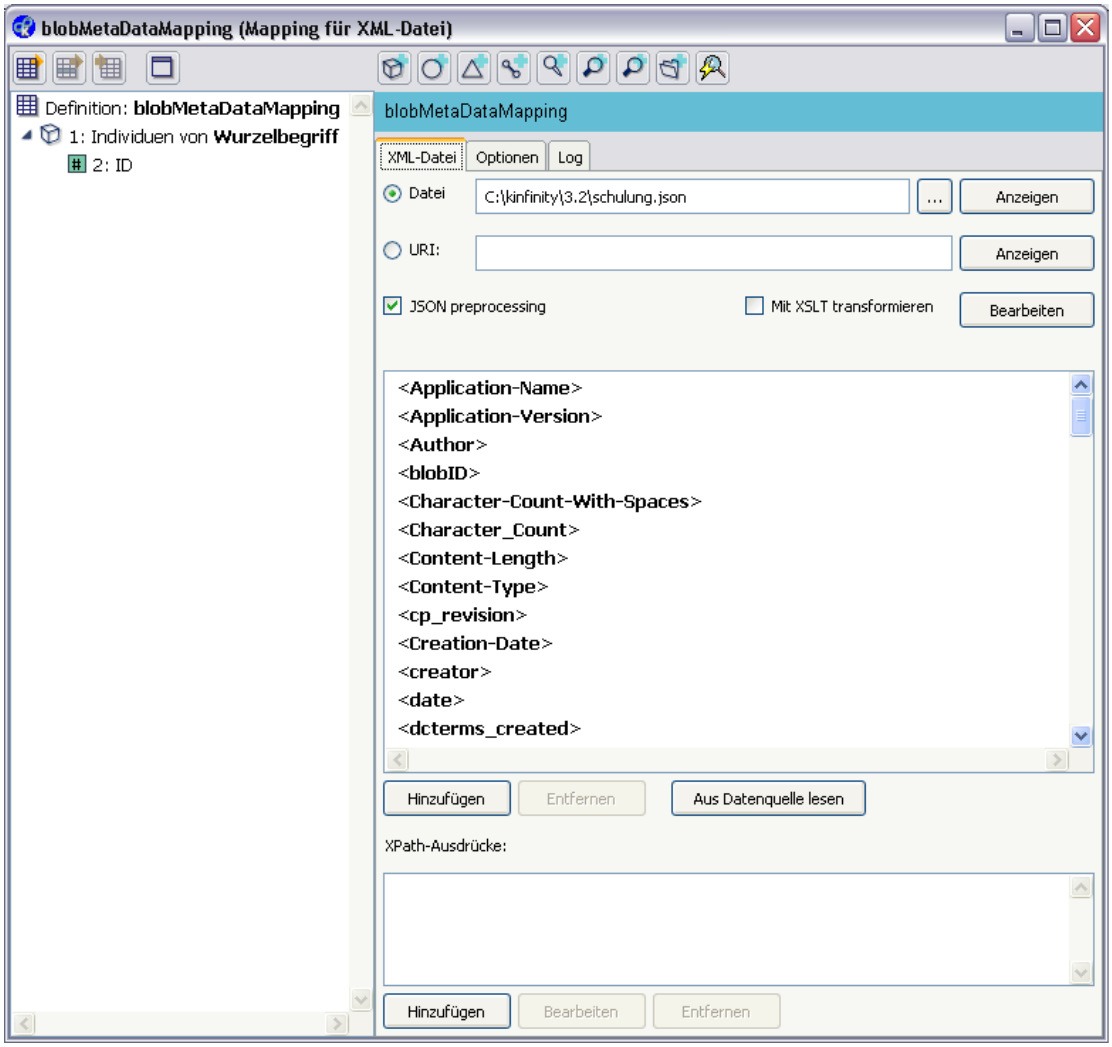
Sind diese Voraussetzungen erfüllt, dann muss nur noch ein Import-Mapping konfiguriert werden, dass den Import der Metadaten beschreibt. Dieses Mapping wird dann automatisch bei der Lucene-Volltextindexierung ausgeführt.

Damit das Mapping bei der Indexierung gefunden wird, müssen folgende Namensvorgaben erfüllt sein :

- Der Mapping-Ordner muss die externe ID "**blobMetaData**" haben.
- Das Import-Mapping ist ein XML-Mapping und hat den Namen "**blobMetaDataMapping**".

Zur Erstellung eines Mappings siehe die Dokumentation für Knowledge-Builder 3.2 - Kapitel 6.3.1 "Neues Mapping".

Die Abbildung zeigt die Einstellungen, die für den Metadatenimport immer vorgenommen werden müssen.



Folgende Punkte sind zu beachten:

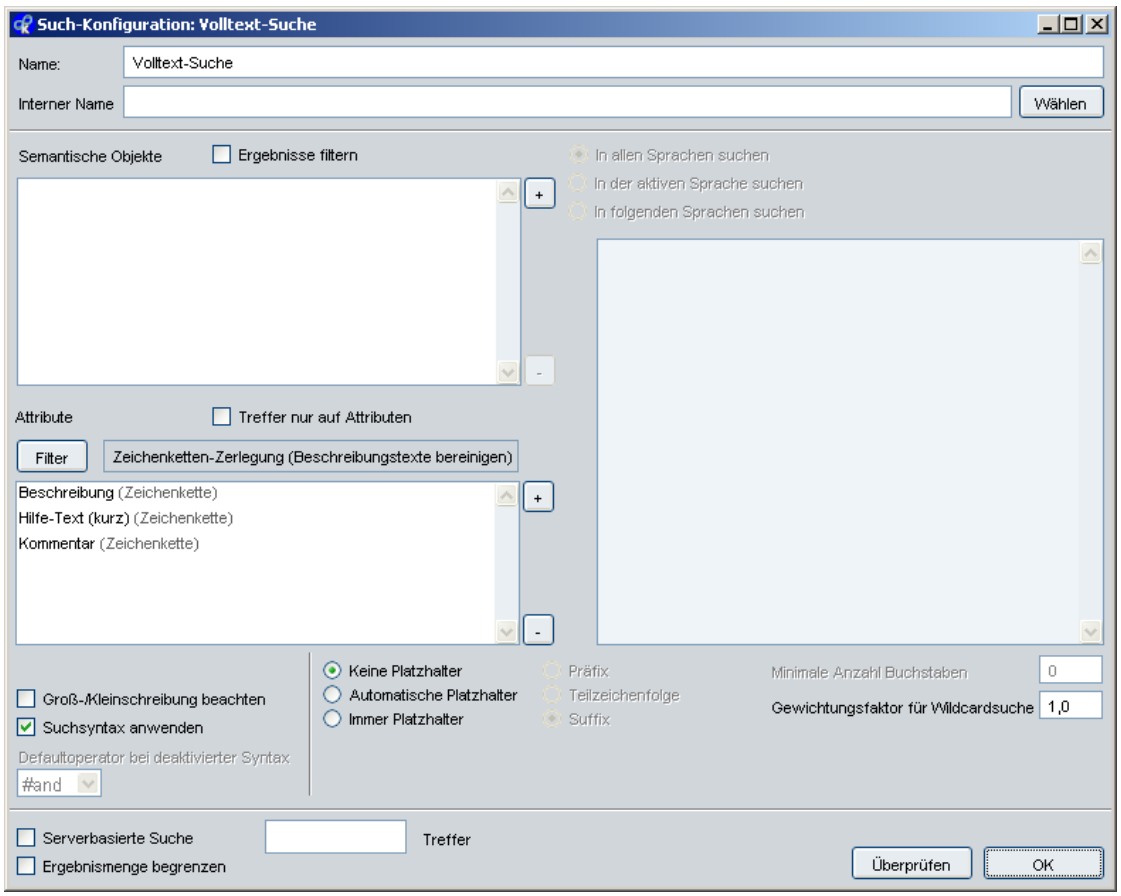
- Die für die Konfiguration notwendige Datei muss vom Typ JSON sein.
- Damit diese für die Konfiguration lesbar wird, muss das Feld "JSON preprocessing" angewählt sein. Klickt man dann auf "Aus Datenquelle lesen", werden die Tags aus der Datei angezeigt.
- Das Datei-Attribut oder dessen Topic, das die Metadaten als Attribute erhalten soll, wird über ein Individuen-Mapping abgebildet. Der Begriff des Individuums muss "**Wurzelbegriff**" sein, damit eine korrekte Abbildung möglich ist.
- Das Individuum des Individuen-Mappings wird über seine Frame-ID identifiziert. Diese wird über das entsprechende ID-Mapping abegebildet. Der Wert hierfür ergibt sich aus dem Tag <**blobID**> für das Datei-Attribut, oder aus dem Tag <**topicID**> für dessen Topic.

Die eigentlich zu extrahierenden Metadaten können beliebig auf weitere Attribut-Mappings abgebildet werden.

Das oben abgebildete Mapping (blobMetaDataMapping.xml) und die dort für die Konfiguration verwendete JSON-Datei schulung.json befinden sich in dem Verzeichnis R:\K-Infinity\Volltextindex\Tika\Me

4.3.4.5 Suche im Volltextindex

Der Volltextindex kann über die Suche "Volltextsuche" abgefragt werden. Die Suchkonfiguration sieht wie folgt aus:



Es werden alle an den Volltextindex angeschlossenen Attribute durchsucht, die Suche kann über einen Jobclient erledigt werden und die zulässige Ergebnismenge kann begrenzt werden.

4.4 Journaling

Das Pflegen von Indexstrukturen benötigt einen nicht vernachlässigbaren Anteil der Rechenzeit, die beim Erstellen, Aktualisieren oder Löschen von Attributen und Relationen anfällt. Ein Journal puffert die Indexänderungen zunächst und schreibt sie gebündelt in den Index, wenn das Limit für das jweilige Journal erreicht ist. Indexstrukturen, die eine hohe Aktualisierungsfrequenz haben, sollten Journaling aktivieren. Dies ist z.B. beim Import großer Datenmengen der Fall.

Zum Aktivieren des Journaling stellt man das Limit auf einen Wert größer Null (4096 ist ein gebräuchlicher Wert).

Zum Deaktivieren des Journaling stellt man das Limit zurück auf Null. Dabei werden auch alle bisher gepufferten Einträge zurückgeschrieben.

Achtung: Im Gegensatz zu relationalen Datenbanken hat das Journaling in K-Inifinity keinen Einfluss auf die Suche (abgesehen von kleinen Performance -Einbußen). Es können also auch gepufferte Einträge gefunden werden.

5 Information

Unter dem Aspekt "Information" findet man Informationen über den aktuellen Zustand des Wissensnetzes und der Software. Die Informationen beinalten eine Auflistung aller internen Namen, eine Übersicht über die am Netz angemeldeten Jobclients, welche Clients mit dem Netz verbunden sind und welche Version die Software hat.

5.1 Interne Namen

Wählt man den Punkt "Interne Namen" an, dann erscheint im rechten Teilfenster eine tabellarische Auflistung aller im Wissensnetz verwendeten internen Namen. Die Tabelle gibt Auskunft darüber, an welchem Obekt und für welchen offiziellen Namen die internen Namen verwendet werden.

Interne Namen

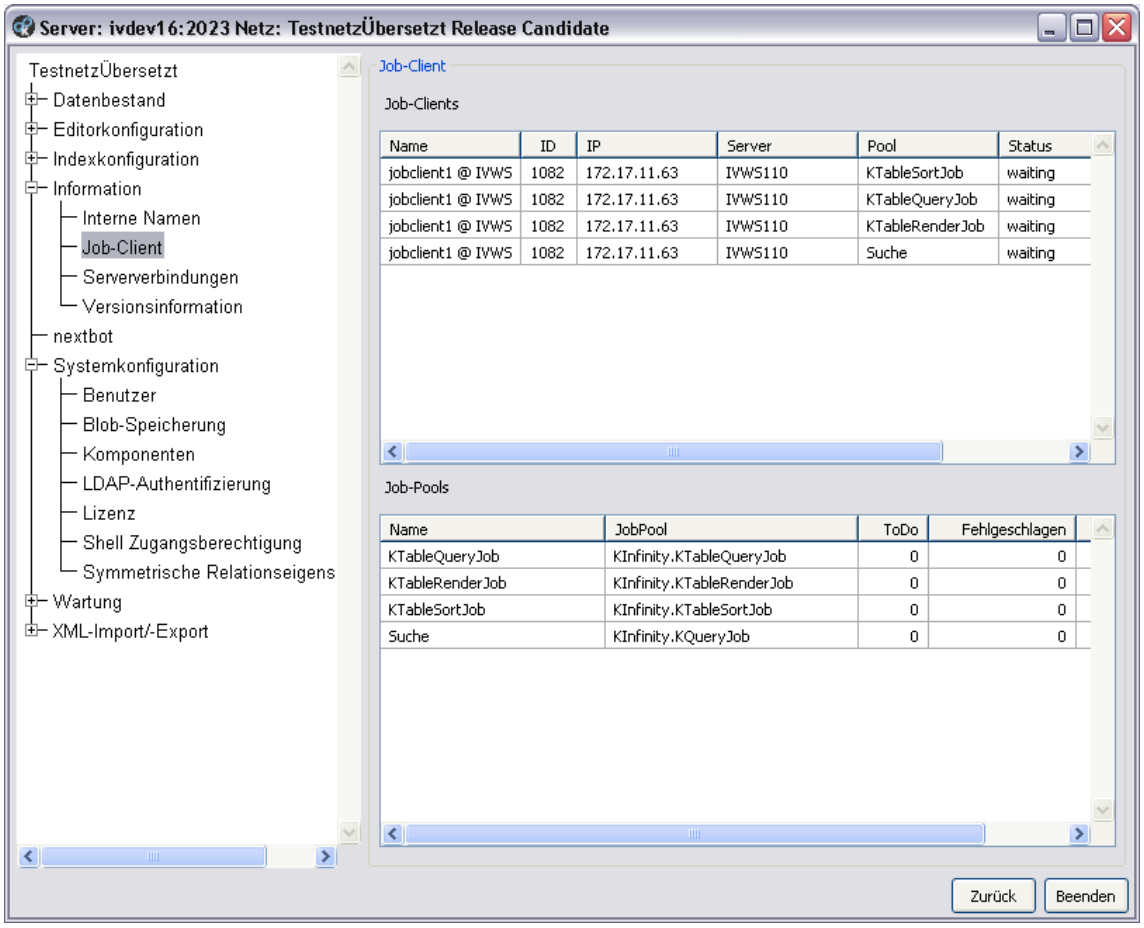
Interner Name	Name	Typ	Attributtyp	Position
bifabMarkupEditorNichtLoes	Nicht löschen (MarkupEditor)	Attribut	Boolesch	hat Verweis aus (abstrakt) (Rela
BIFABName	Name-Term	Begriff	---	
BIFABPragmatik	Pragmatik	Attribut	Formatierte Zeichenkette	Zus (Formatierte Zeichenkette)
BIFABPragmatik	Wortverwendungsklasse	Begriff	---	
BIFABPragmatik	Pragmatik	Attribut	Formatierte Zeichenkette	Sinnv (Formatierte Zeichenkette)
BIFABRedensartLemma	Redensart-Lemma	Begriff	---	
BIFABSprichwortLemma	Sprichwort-Lemma	Begriff	---	
BIFABTerm	Benennung	Begriff	---	
BIFABTopicBehavior	Topic	Begriff	---	
bifabWerkKuerzel	Kürzel	Attribut	Zeichenkette	Werk (Individuum)
BIFABWortgruppenLemma	Wortgruppen-Lemma	Begriff	---	
conceptArtikel	Artikel	Attribut	Formatierte Zeichenkette	Subst. (Begriff)
conceptEditorConfiguration	Begriffseditorkonfiguration	Attribut	Zeichenkette	T (Begriff)
darstellung	Darstellung	Attribut	Formatierte Zeichenkette	Bezeichnung (Individuum)
darstellungVereinfacht	Darstellung, vereinfacht	Attribut	Formatierte Zeichenkette	Bezeichnung (Individuum)
datumLetzteAenderung	Letzte Änderung	Attribut	Datum	Bearbeitungszustand (Auswahl)
defiInformation	def-i Information	Attribut	Formatierte Zeichenkette	TermKontext (Individuum)
definiertDurchSynonym	definiert durch Synonym	Relationsbegriff	---	
definiertSynonym	definiert Synonym	Relationsbegriff	---	
displayName	Anzeigename	Attribut	Zeichenkette	T (Individuum)
dudenempfehlung	Dudenempfehlung	Attribut	Boolesch	Bezeichnung (Individuum)
editorConfiguration	Editorkonfiguration	Attribut	Zeichenkette	T (Begriff)
ergaenzungDef	Ergänzung (Definitionstext)	Attribut	Formatierte Zeichenkette	Be{zeich|nen}nung (Individuum)

Liste der internen Namen

Diese Liste ist identisch mit der Liste, die auch im Knowledge-Builder angezeigt werden kann.

5.2 Job-Client

Unter dem Aspekt "Job-Client" erhält man einen Überblick über die aktiven Job-Clients und die vorhandenen Jobs. Die Jobs sind in Pools eingeteilt: Ein Pool enthält alle Jobs eines bestimmten Typs (z.B. Suchen).



Job-Client

In der oberen Tabelle sind die momentan aktiven Job-Clients und in der unteren Tabelle die Job-Pools zu sehen. In beiden Tabellen können durch die rechte Maustaste Kontextmenüs geöffnet werden.

5.2.1 Liste der Job-Clients

In der oberen Tabelle wird pro Job-Client und Job-Pool eine Zeile angezeigt. „ID“ gibt eine eindeutige ID des Job-Clients an. Falls also ein Job-Client für mehrere Job-Pools zuständig ist, werden mehrere Zeilen mit der gleichen ID angezeigt, die alle denselben Job-Client betreffen.

Im Kontextmenü werden folgende Aktionen angeboten:

- „Job-Client sicher herunterfahren“ sendet an den Job-Client die Aufforderung, sich zu beenden. Momentan durchgeführte Jobs werden noch fertig abgearbeitet.
- „Abgestürzten Job-Client entfernen“ sorgt dafür, dass ein Job-Client, der fälschlicherweise noch in der Liste auftaucht, entfernt wird. Dies kann z.B. der Fall sein, wenn der Job-Client-Prozess „hart“ beendet (Absturz des Rechners o.ä.) wurde und somit nicht mehr aus der Liste ausgetragen werden konnte.
- „Informationen anzeigen“ zeigt einige (interne) Statuswerte an.

5.2.2 Liste der Job-Pools

In der unteren Tabelle sind die Job-Pools zu sehen. „To Do“ zeigt die Anzahl der noch abzuarbeitenden Jobs an, „Fehlgeschlagen“ zeigt die Anzahl der fehlgeschlagenen Jobs an. „Job-Clients“ gibt an, wie viele der momentan aktiven Job-Clients für den Job-Pool zuständig sind.

Im Kontextmenü der Job-Pools stehen die folgenden Aktionen zur Verfügung:

1. Job-Pool leeren: Alle ausstehenden und fehlerhaften Jobs dieses Typs werden entfernt und nicht mehr ausgeführt. Dies ist nur möglich, wenn keine Job-Clients aktiv sind.
2. Zu ignorierende Fehlermeldungen konfigurieren: Diese Aktion ist nur möglich, wenn schon Jobs dieses Typs existieren oder ausgeführt wurden. In einem Dialog können Fehlermeldungen als Zeichenketten eingegeben werden, die im Falle ihres Auftretens ignoriert werden sollen. Tritt ein Fehler mit einer in dieser Liste aufgeführten Fehlermeldung auf, dann wird der Job nicht in die Liste der fehlgeschlagenen Jobs eingetragen. Das ist zum Beispiel dann interessant, wenn der Fehler bei einem externen Dienst aufgetreten ist und somit kein Bedarf daran besteht zu wissen, warum die Bearbeitung des Jobs fehlgeschlagen ist.

5.3 Serververbindungen

An dieser Stelle wird dargestellt, welche Clients sich auf dem Server in welches Netz verbunden haben. Es wird die ClientID, die vom Server bei der Anmeldung vergeben wird, und die IP-Adresse angezeigt.

Der kursive Eintrag ist das Admin-Tool selbst.

5.4 Versionsinformation

An dieser Stelle lassen sich die Versionsinformationen betrachten. Der Inhalt ist identisch mit dem Aufruf des Menü-Punktes Hilfe -> Info im Knowledge-Builder.

Im Einzelnen werden folgende Inhalte angezeigt:

- Build
  Nummer des Builds. Die Nummer schlüsselt sich wie folgt auf: YYMMDDNN. Wobei die einzelnen Buchstaben folgendes bedeuten: Y: Year, M: Month, D: Day, N: Number (fortlaufende Nummer für alle Builds an einem Tag)
- Release State
  Aktueller Release-Zustand der Software.
- Netzversion
  Aktuelle Netzversion
- Netz-Information
  <Netzname> @ <Servername>:<Port>
- Speicherbegrenzung
  Speichergrenze in Bytes, die vom System oder per Parameter vorgegeben ist
- VM Version
  Aktuelle VM-Versionsnummer
  und Array (Zahlenfolge) für die Angabe weiterer Parameter:
  bytes 1 - 4 contain this platform's id where:
  1 = platform version,

2 = platform number,
3 = platform release major number,
4 = platform release minor number (bits 7-4) and flags (bits 3-0)
bytes 5 - 8 contain this image's id where:
5 = image major version,
6 = image minor version,
7 = image release major number,
8 = image release minor number
bytes 9 - 12 contain the platform id that made this image

- Locale
  Aktive Locale-Einstellung
- Komponenten
  Aktive Komponenten im Netz
- Pakete
  Auflistung aller Pakete mit deren Versionsnummern

Mit der Schaltfläche "RSA-Key kopieren" wird die buildID und der zugehörige öffentliche Schlüssel des Admintools in die Zwischenablage kopiert. Dieser wird benötigt, um die Zugangskontrolle für Anwendungen zu konfigurieren.

6 nextbot

Administration und Betrieb der nextbot Erweiterung sind in einem eigenen Dokument beschrieben, sodass hier auf die Replikation der Inhalte verzichtet wird.

7 Systemkonfiguration

7.1 Benutzer

Hier werden die verfügbaren Benutzerkonten aufgelistet. Wenn im Rechtesystem ein Personenbegriff konfiguriert wurde (Menü "Datei -> Einstellungen -> Rechte" im Knowledge-Builder), kann ein Benutzerkonto auch mit einem Individuum aus dem Wissensnetz verknüpft werden. Auf diese Weise können Rechte auf spezifische Benutzer ausgeweitet oder eingeschränkt werden.

Bei momentan mit dem Knowledge-Builder angemeldeten Benutzern wird der Status als "aktiv" gekennzeichnet.

- Mit *Erstellen* wird ein neues Benutzerkonto angelegt. Man wird nach dem Namen und dem Passwort des neuen Benutzers gefragt.
- Über *Verknüpfen* kann man den ausgewählten Benutzer mit einem Benutzer-Individuum des Wissensnetzes verknüpfen. Zur Verfügung stehen alle Individuen des Personenbegriffs, die noch nicht verknüpft wurden.
- Mit *Verknüpfung aufheben* wird die Zuordnung wieder entfernt.
- Mit *Passwort ändern* kann dem Benutzer ein neues Passwort gegeben werden.
- Falls es für Administrationsaufgaben o.ä. notwendig ist, dass Benutzer vom Wissensnetz

abgemeldet sind, können aktive Benutzer mit Abmelden ferngesteuert abgemeldet werden. Falls ein angemeldeter Benutzer auf diese Weise abgemeldet wird, erscheint bei diesem Benutzer ein entsprechende Meldung.

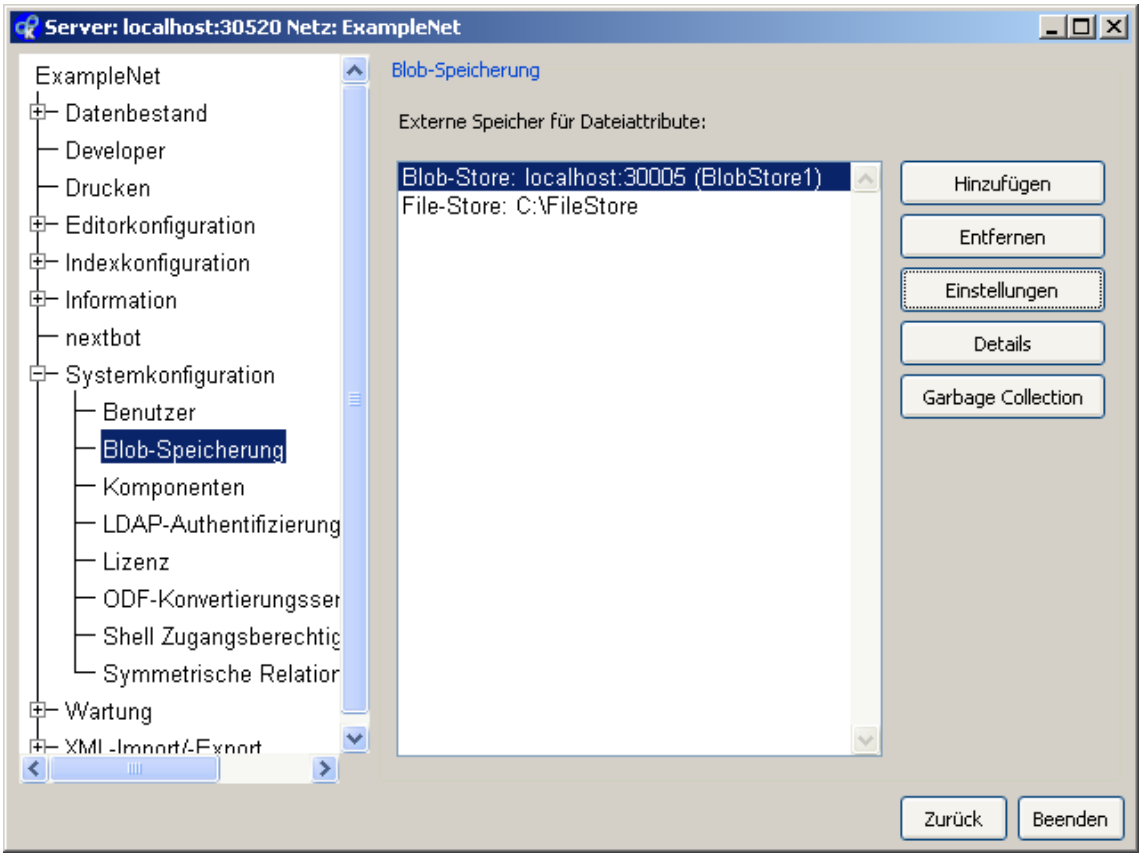
- *Löschen* entfernt ein Benutzerkonto. Falls der Benutzer mit einem Individuum verknüpft wurde, wird die Verknüpfung aufgehoben, das Individuum wird aber nicht mitgelöscht.
- Mit dem Ankreuzfeld *Administrator* wird festgelegt, ob der Benutzer Administratorrechte besitzt.

Falls nur ein Benutzerkonto vorhanden ist, erhält dieser Benutzer automatisch Administratorrechte.

Zur Information, die z.B. für Lizenzfragen wichtig ist, werden noch statistische Werte zu Anzahlen aktuell angelegter Benutzer und ihrer Funktion angezeigt.

7.2 Blob-Speicherung

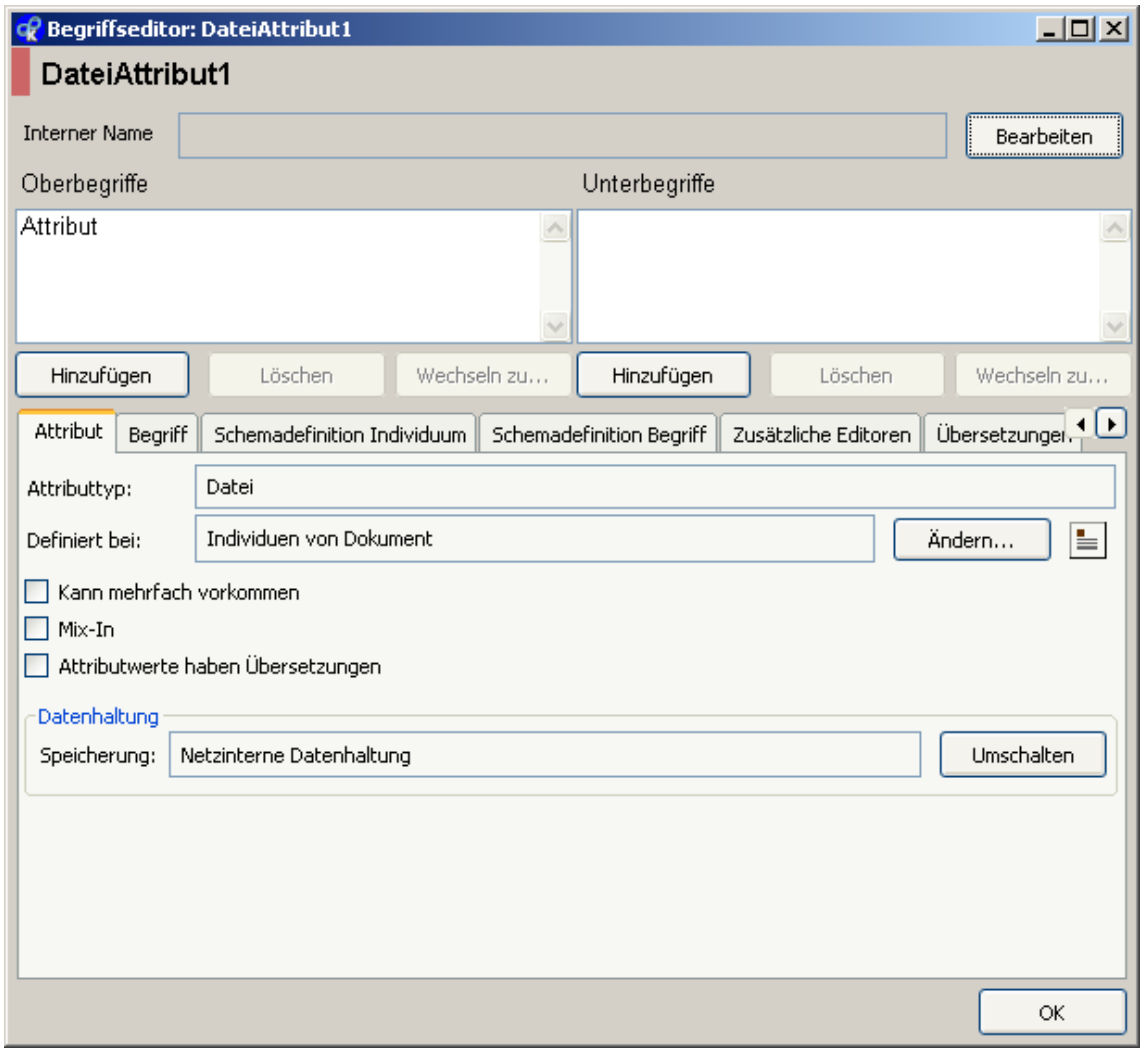
Dateiattribute (Blobs) werden per Default im Wissensnetz gespeichert. Da die Blob Inhalte in diesem Fall das Volumen des Wissensnetzes vergrößern, kann es je nach geplantem Blob-Volumen sinnvoll sein, eine externe Speicherung außerhalb des Netzes zu verwenden. Dann werden dafür konfigurierte Attribute vom Typ Datei außerhalb des Netzes in den externen Speichern verwaltet. In diesem Konfigurationsabschnitt des Admin-Tools wird die Verwaltung der externen Speicher beschrieben.



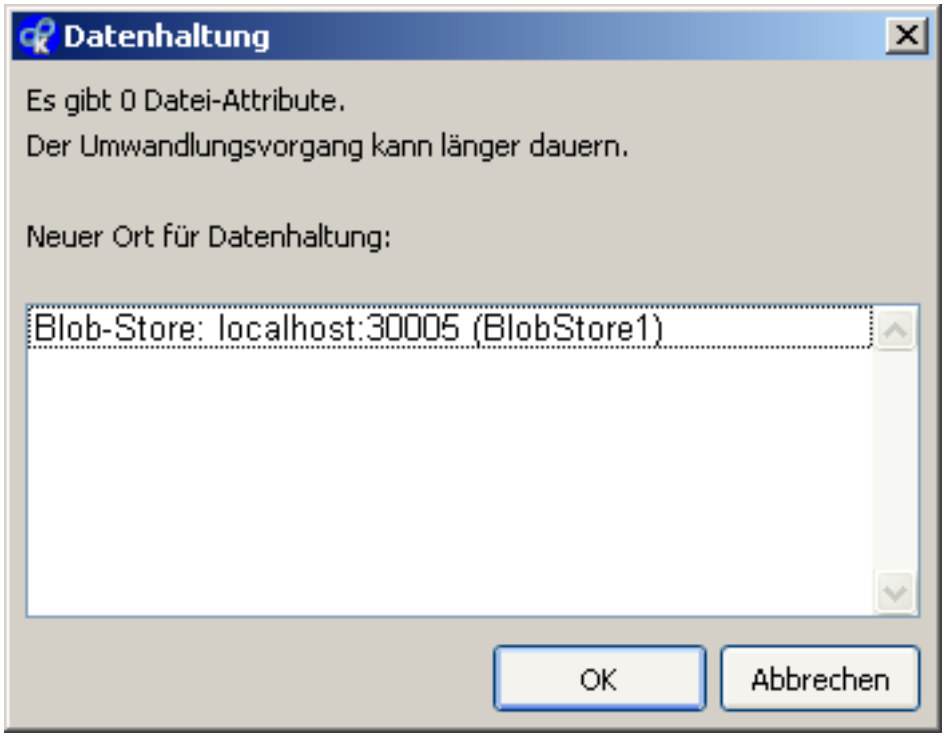
Wie in der obigen Abbildung zu sehen, gibt es zwei Arten externer Speicher: File-Store und Blob-Store. Diese werden in den folgenden Unterkapiteln näher beschrieben, hier folgt noch die Beschreibung der oben abgebildeten Schnittstelle.

- Mit "Hinzufügen" kann ein neuer externer Speicher angelegt. Man wird zuerst nach dem Typ des externen Speichers (File-Store oder Blob-Store) gefragt. Anschließend wird der externe Speicher angelegt und das Konfigurationsfenster geöffnet.
- Mit "Entfernen" wird der externe Speicher aus dem Netz entfernt. Die gespeicherten Daten bleiben dabei aber erhalten, diese muss man bei Bedarf von Hand löschen. Externe Speicher können nur entfernt werden, wenn sie nicht mehr von Dateiattributen oder im Falle eines File-Stores nicht von einem Blob-Store verwendet werden.
- "Einstellungen" öffnet das spezifische Konfigurationsfenster (siehe Unterkapitel 10.1 und 10.2).
- "Details" zeigt den aktuellen Status des externen Speichers an sowie alle Objekte, die den externen Speicher verwenden.
- "Garbage Collection" startet eine Garbage Collection, bei der alle nicht mehr vom Wissensnetz referenzierten Dateien endgültig aus dem jeweiligen Store gelöscht werden. Im Falle des Blob-Stores wird die Garbage Collection vom Blob-Service-Dienst durchgeführt.

Die Verwendung der externen Speicherung wird an den Attributen selbst konfiguriert, also innerhalb des Knowledge Builders. Hierzu kann man im unteren Bereich des Reiters "Attribut" beim Editor für einen Attributbegriff die Datenhaltung einstellen.



Dies ist auch jederzeit möglich, nachdem bereits Blobs erfasst wurden und die Datenhaltung geändert werden soll. Nach Auswahl des "Umschalten" Knopfs wird ein Dialog zur Wahl der neuen Speicherart angezeigt, der verfügbare Speicherarten auflistet.



Die Dauer der Migration der Blobs von einer Speicherart zur anderen hängt vom Datenvolumen und der Netzwerkverbindung ab.

7.2.1 File-Store

File-Store bedeutet, dass alle Dateien im anzugebenen Verzeichnis und dessen Unterverzeichnissen gespeichert werden. Dieses Verzeichnis muss für alle Clients, die mit Dateien arbeiten, erreichbar und beschreibbar sein. In diesem Verzeichnis wird zur Verwaltung automatisch eine Konfigurationsdatei KFilesystemStore-<id>.ini angelegt. Die Blobs werden in durchnummerierten Unterverzeichnissen gespeichert. Die Konfigurationsdatei enthält zwei Einträge:

```
lastDir=2
volumeID=jhkibimgggmoamin
```

- **lastDir** enthält den Namen des Unterverzeichnisses, das die zuletzt gespeicherte Datei enthält. Dieser Eintrag dient nur zur Performanceoptimierung.
- **volumeID** enthält die ID des Wissensnetzes. Dies soll verhindern, dass ein File-Store versehentlich mit einem falschen Netz betrieben wird. Dies betrifft auch insbesondere eine Kopie (oder neu hochgeladene Version) eines Netzes, für das der File-Store ursprünglich angelegt wurde. Netzkopien benötigen eigene Kopien aller enthaltenen File-Stores.

Will man ein Netz mit einem oder mehreren File-Stores kopieren, muss man folgende zusätzliche Schritte nach dem Kopieren des Wissensnetzes durchführen:

- Verzeichnis des File-Stores kopieren
- In dieser Kopie die Konfigurationsdatei KFileSystemStore-<id>.ini löschen. Dadurch verliert die Kopie die Verbindung zum alten Wissensnetz. Beim ersten Zugriff aus der neuen Kopie des Wissensnetzes auf diese Kopie des File-Stores wird eine neue Konfigurationsdatei angelegt.
- Im kopierten Netz ist in den Einstellungen des File-Stores das neue Dateiverzeichnis anzugeben.

7.2.2 Blob-Store

Da beim File-Store die Vorraussetzung, daß alle Clients auf das Dateiverzeichnis zugreifen können müssen, nicht immer erfüllbar oder wünschenswert ist, kann der Zugriff auf den File-Store auch mit Hilfe des Blob-Stores über das Netzwerk erfolgen. Ein Blob-Store wird von einem eigenständigen Dienst (Blob-Service) bedient, der auf einem zu konfigurierenden Port je Blob-Store Anfragen der Clients (Hoch-/Herunterladen von Datein etc.) entgegennimmt und ausführt. Ein Blob-Store kapselt dabei einen File-Store der ebenso im Netz konfiguriert ist. Ein von einem Blob-Store gekapselter File-Store ist nicht mehr direkt von Attributen als Speicherort verwendbar sondern nur noch über den Blob-Store.

Die Dokumentation zur Einrichtung eines Blob-Store Dienstes findet sich in einem eigenen Dokument im Bereich Systemdokumentation.

Folgende Einstellungen müssen bei der Konfiguration vorgenommen werden:

- Hostname: Name des Rechners, auf dem später der Blob-Service-Dienst läuft. Dieser kann üblicherweise auf "localhost" belassen werden, da diese Einstellung beim Start des Blob-Services von diesem überschrieben wird und hier nicht geändert werden muss.
- Port: Die Nummer des TCP Ports, unter dem der Blob-Store Anfragen entgegennimmt. Dieser Port muss dann für alle Clients erreichbar sein.
- Interner Name: Eindeutige Bezeichnung des Blob-Stores, die in der Konfigurationsdatei des Blob-Services verwendet wird.
- Externer Speicher: Zeigt eine Liste aller vorhandenen externen Speicher an, in der der selektierte externer Speicher von diesem Blob-Store erfasst wird. Der Blob-Store leitet alle Anfragen der Clients an diesen externen Speicher weiter.

Innerhalb eines Netzes können somit verschiedene Datei-Attribute in verschiedenen Blob-Stores abgelegt werden. Die Blob-Stores können dann wiederum auf einen oder mehrere Blob-Services verteilt werden.

7.2.3 Umwandlung der Datenhaltung

Dieser Abschnitt erklärt die typische Anforderung die Daten von einer netz-internen Datenhaltung auf eine Speicherung durch einen externen Blob-Service umzustellen.

In der Planungsphase (die sehr kurz ausfallen kann) fallen folgende Aufgaben an:

- Identifizieren der zu migrierenden Dateiattribute
- Aufteilen der Dateiattribute und entsprechenden Datenvolumen auf einen oder mehrere Datenbereiche
- Reservieren der Betriebssystemressourcen: je ein Verzeichnis und TCP Port je Blob-Store, je ein Prozess (ggfs. auf verschiedenen Rechnern) je Blob-Service.

Im Admintool legt man daraufhin die notwendigen File-Stores und Blob-Stores im Bereich "Systemkonfiguration, Blob-Speicherung" an

- Anlegen eines File-Stores je Datenbereich, dazu je ein Verzeichnis spezifizieren.
- Anlegen eines Blob-Stores je File-Store. Jeder Blob-Store bekommt einen TCP Port zugewiesen, einen eindeutigen internen Namen und einen der eben angelegten File-Stores zugewiesen.

Mit diesen Konfigurationsdaten kann man die Blob-Services installieren und per ini-Datei konfigurieren, d.h. die entsprechenden Verzeichnisse anlegen, ini-Dateien schreiben und Di-

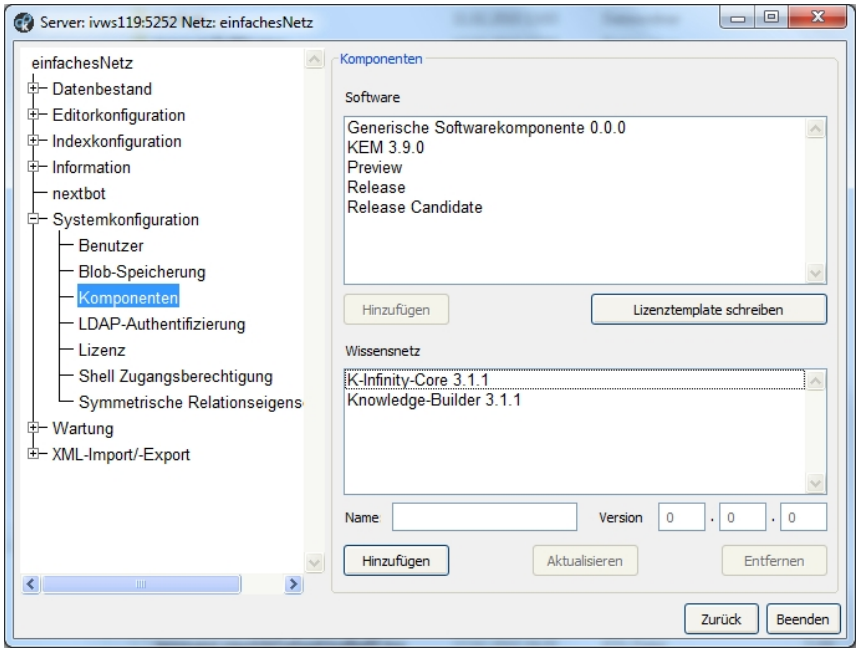
enste starten.

Danach kann man im Knowledge Builder die Datenhaltung der betroffenen Datei-Attribute umstellen (Reiter Attribut des Attributbegriffs). Hierbei wird der Knowledge Builder die Inhalte jedes Attribut aus dem Netz in den jeweilgen Blob-Store übertragen.

Nach der Umstellung ist es sinnvoll, den Mediator anzuweisen eine Garbage Collection auf dem Netz auszuführen, damit der ehemals von den Blobs belegte Platz freigegeben wird. Gleichzeitig sollte man daran denken, die verschiedenen Verzeichnisse mit in ein Dateisystem-Backup aufzunehmen, da diese Inhalte jetzt nicht mehr von Backup des Mediators erfasst werden.

7.3 Komponenten

Mit Hilfe der Komponenten-Konfiguration lassen sich Versionen von einzelnen Bausteinen in der Software und im Wissensnetz kontrollieren.



Komponenten-Ansicht

Im oberen Bereich auf der rechten Seite sind die Komponenten, die in der Software enthalten sind, aufgelistet. Diese lassen sich zum Netz hinzufügen. Beim Hinzufügen einer Komponente finden unter Umständen Anpassungen im Netz statt. Es finden Schemaergänzungen statt, damit die neue Komponente funktionieren kann. Mit dem Knopf *Lizenztemplate schreiben* wird eine Textdatei erzeugt, die verwendet wird, um eine Lizenzdatei von intelligent views oder einem Partner von intelligent views zu schreiben.

Im unteren Bereich sind die im Netz enthaltenen und konfigurierten Komponenten aufgelistet. Ist in der Software eine neue Version vorhanden, wird dies mit der Meldung *(Auf x.x.x aktualisieren)* angezeigt. Es wird ebenfalls mit der Meldung *(Unlizensierte Komponente)* anzeigt, wenn eine Komponente nicht ausreichend lizensiert ist.

Es stehen weitere Knöpfe zur Verfügung:

- *Hinzufügen*: Es gibt die Möglichkeit, generische Model-Komponenten hinzuzufügen. Diese können vom Administrator frei benannt und mit Versionnummern versehen werden. Sie dienen der eigenen Versionskontrolle und werden von der Software nicht direkt ange-

sprochen.

- *Aktualisieren*: Ist eine Aktualisierung einer Software-Komponente notwendig, kann diese mit diesem Knopf durchgeführt werden.
- *Entfernen*: Manche Komponenten können entfernt werden. Einige Komponenten sind zwingend erforderlich, damit die Software funktioniert. Diese lassen sich nicht entfernen.

7.3.1 Release State

In der Liste der Softwarekomponenten werden die möglichen Release-Zustände angezeigt, die eine Software haben kann.

- Preview
- Release Candidate
- Release

Der aktuelle Zustand kann in der Versionsinformation (Baumstruktur <Netzname> –> Information) unter dem Punkt *Release State* nachgesehen werden.

Mit dem Hinzufügen einer dieser Softwarekomponenten kann der Administrator festlegen mit welchem Release-Zustand der Software das Netz betreten werden darf.

7.3.2 Generische Softwarekomponente

Einstellungen in diesem Bereich werden nur von intelligent views bzw. auf Anweisung von intelligent views vorgenommen.

7.3.3 Generische Modellkomponente

Einstellungen in diesem Bereich werden nur von intelligent views bzw. auf Anweisung von intelligent views vorgenommen.

7.3.4 Druckkomponente

Mit Hilfe der Druckkomponente kann man Dokumentvorlagen (RTF- bzw. ODT-Dateien) mit kPath-Ausdrücken auf Objekten oder Objekt-Listen anwenden und daraus eine angepasste Ausgabe-Datei generieren, die entweder gedruckt oder gespeichert werden kann.

Hinweis: Im Folgenden wird OpenOffice synonym für "OpenOffice oder LibreOffice" verwendet.

7.3.4.1 Schema der Druckvorlagen

Nach dem hinzufügen der Druckkomponente wird im Teilnetz "Intern" der abstrakte Begriff "Druckvorlage (abstrakt)" erstellt. Darunter werden die individuenfähigen Begriffe "Druckvorlage" (interner Name: "Druckvorlage") und "Druckvorlage (Liste)" (interner Name: "DruckvorlageListe") erstellt. Die zwei Typen werden als Vorlage für Individuen bzw. Objektlisten verwendet.

Die Individuen dieser Begriffe speichern die Dokumentvorlagen im Dateiattribut "Dokument (Druckvorlage)" (interner Name: "printTemplateBlob").

Begriffe können eine Relation "hat Druckvorlage" (interner Name: hasPrintTemplateRelation) zu den Vorlage-Individuen eingehen. Ein Anwender kann dann eine Druckvorlage (oder "Druckvorlage(Liste)") auf einem Objekt (bzw. Objektliste) anwenden, wenn diese Relation zwischen dem zugehörigen Begriff und dem Vorlage-Individuum besteht und er Leserecht auf dieser Relation hat.

Somit können die Druckvorlagen je Anwender und Objekttypen konfiguriert werden.

Im Wissensnetz hinterlegte Druckvorlagen werden automatisch an der Dateiendung der enthaltenen Datei erkannt und zugeordnet.

7.3.4.2 Der Druckdialog

Im Druckdialog kann der Anwender zum angezeigten Druckobjekt (bzw. Liste) die Vorlage wählen. Als letzte Vorlage ist "Datei ..." angegeben - hierüber kann der Anwender eine Druckvorlage manuell wählen. Zudem kann er einen Drucker auswählen. Auch hier gibt es die Möglichkeit, eine Datei zu wählen, sodass die Ausgabe-Datei weiter bearbeitet werden kann.

7.3.4.3 RTF-Dokumente erstellen

Die RTF-Vorlagedateien können auswertbare kPath-Ausdrücke mit den Schlüsselworten **KPATH_EXPAND** und **KPATH_ROWS** sowie Aufrufe registrierter kSkripte mit den Schlüsselworten **KSCRIPT_EXPAND** und **KSCRIPT_ROWS** enthalten. Die Pfadausdrücke bzw. der Name des aufzurufenden Skriptes stehen immer zwischen spitzen Klammern und nach dem Schlüsselwort durch ein Leerzeichen getrennt.

KPATH_EXPAND

Der kPath-Ausdruck nach diesem Schlüsselwort sollte ein einzelnes semantisches Objekt oder einen einfachen Wert (Datum, Zeichenkette etc.) zurückliefern. Bei der Auswertung wird der ursprüngliche Ausdruck durch das Ergebnis ersetzt. Die Formatierung des Ausdrucks bleibt erhalten, Umbrüche des Wertes werden in Zeilenumbrüche umgesetzt.

Beispiel:
Die Vorlage sei:

```
Absender:
<KPATH_EXPAND @$adresse$/rawValue()>
```

Nach Auswertung steht in der Ausgabedatei:

```
Absender:
intelligent views gmbh
Julius-Reiber-Str. 17
64293 Darmstadt
```

KSCRIPT_EXPAND

Alternativ zum Pfadausdruck kann mit KSCRIPT_EXPAND ein registriertes KScript aufgerufen werden. Die Ausgabe dieses Skriptes (Skriptelemente mit <Output>) wird in das Dokument übernommen. Die Registrierung von Scripten erfolgt im KB im Systemordner 'Registrierte Scripte'.

Beispiel:

Die Vorlage sei:

```
<KSCRIPT_EXPAND einSkriptDas1bis9Ausgibt>
```

Nach Auswertung steht in der Ausgabedatei:

```
123.456.789
```

KPATH_ROWS

Dieser Ausdruck muss in einer Tabelle stehen. Der kPath-Ausdruck nach diesem Schlüsselwort muss eine Liste semantischer Objekte liefern. Bei der Auswertung wird die Tabellenzeile des KPATH_ROWS Ausdrucks für jedes Ergebnis des kPath-Ausdrucks einmal ausgewertet. Somit können Tabellen dynamisch ergänzt werden. Es spielt übrigens keine Rolle, in welcher Spalte der KPATH_ROWS Ausdruck steht.

Beispiel:
Die Vorlage sei:

Teile (<KPATH_EXPAND topic()/~$hatTeile$/size()> Stück)	**Bemerkung**
<KPATH_EXPAND topic()><KPATH_ROWS topic()/~$hatTeil$/target()/sort(@$name$, true)>	<KPATH_EXPAND topic()/@$bemerkung$>

Nach Auswertung in der Ausgabedatei:

Teile (3 Stück)	**Bemerkung**
RTF-Druck	
ODT-Druck	Ersetzt den RTF-Druck
Konvertierungsservice	Optionaler Dienst

KSCRIPT_ROWS

Bei KSCRIPT_ROWS werden die Objekte für die Tabellenzeilen durch ein registriertes Script ermittelt. Der Name des registrierten Skriptes wird direkt innerhalb des Scriptes werden Objekte, die als Tabellenzeile ausgegeben werden sollen, mit <AddToFolder/> zum Ergebnisorder hinzugefügt.

Beispiel:
Die Vorlage sei:

Spalte1	**Spalte2**

<KSCRIPT_ROWS allePersonen><KPATH_EXPAND @$nachname$>	<KPATH_EXPAND @$Vorname$>

Nach Auswertung in der Ausgabedatei:

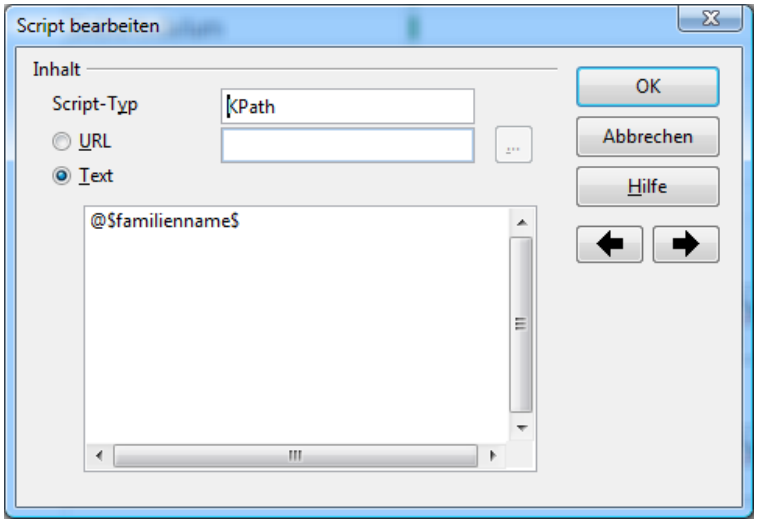
Spalte1	**Spalte2**
Meier	Peter
Schulze	Helmut

7.3.4.4 ODT-Dokumente (OpenOffice) erstellen

Der Druck über das ODT-Format (Open Document Text, offener Standard) hat viele Vorteile gegenüber dem RTF-Format:

- Die eingebetteten Skript-Anweisungen sind nicht Teil des Textes sondern werden in speziellen Script-Elementen abgelegt. Somit macht man sich seine Formatierung nicht durch längliche Skripte kaputt.
- Das ODT-Format unterstützt eine große Menge an Formatanweisungen (vergleichbar mit MS-Word), die RTF nicht kennt.
- RTF hat als Format keine einheitliche Normierung (MS-Word kann z.B. "mehr" RTF als der Standard).
- Die Bearbeitung der RTF-Vorlagen ist sehr fragil. Vor allem MS-Word neigt dazu, die Vorlagen mit Steuerelementen (wie z.B. die aktuelle Cursorposition bei der letzen Bearbeitung) zu ergänzen , sodass die Skripte nicht mehr verlässlich erkannt werden können.

ODT-Vorlagen können mit OpenOffice erstellt werden. Die Erstellung erfolgt analog zu der Erstellung von RTF-Vorlagen mit dem Unterschied, dass die Path-/Script-Anweisungen in Script-Elementen abgelegt werden, wie in der folgenden Abbildung gezeigt.



Script-Bearbeitung in OpenOffice

Als Script-Typen gibt es:

- **KPath** : analog zu **KPATH_EXPAND**
- **KScript** : analog zu **KSCRIPT_EXPAND**
- **KPathRows** : analog zu **KPATH_ROWS**
- **KPathImage** : zur Einbettung von Bildern

Zur Erstellung von ODT-Dateien müssen sich die Dateien "zip32.dll" und "unzip32.dll" aus dem InfoZip-Paket (http://www.info-zip.org) im Ausführungsverzeichnis befinden. (Ab K-Infinity 3.2.8 nicht mehr erforderlich)

Zur Einbettung von Bildern können Datei-Attribute oder URLs verwendet werden. Bei der Verwendung von URLs wird versucht, ein Bild von der angegebenen Adresse zu laden.

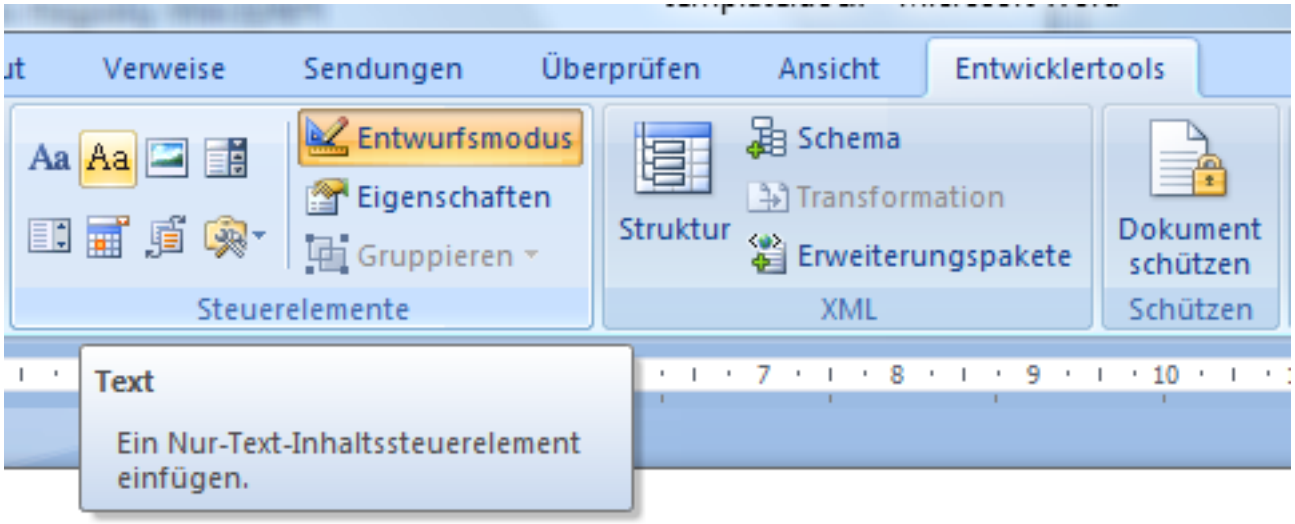
Eingebettete Bilder werden immer auf ihre Originalgröße (bei 96d dpi) gezogen. Möchte man im Ausdruck eine andere Größe erhalten, muss man einen Rahmen mit den gewünschten Ausmaßen (unbedingt absolute Maße in cm verwenden!) um das Script-Element bauen. Das resultierende eingebettete Bild wird dann so in den Rahmen eingepasst, dass das Rahmenmaß unter Beibehaltung der Bild-Seitenverhältnisse nicht überschritten wird.

7.3.4.5 DOCX-Dokumente (Microsoft Word) erstellen

DOCX-Vorlagen können mit Microsoft Word 2007 oder neuer erstellt werden.

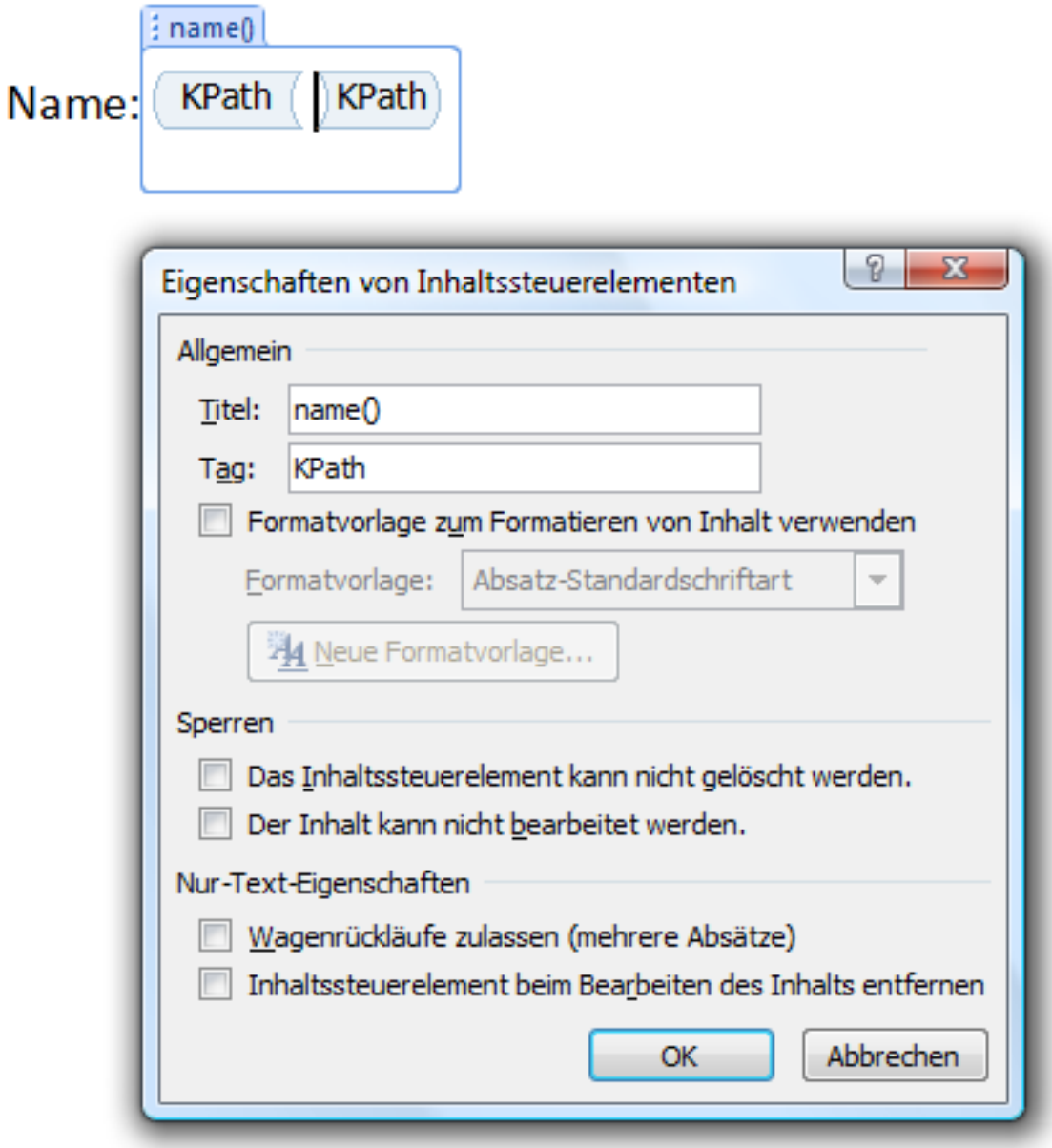
Die Erstellung erfolgt analog zu der Erstellung von RTF-Vorlagen mit dem Unterschied, dass die Path-/Script-Anweisungen in Text-Inhaltssteuerelementen abgelegt werden.

Zum Einfügen der Steuerelement müssen in Word zuerst die Entwicklertools aktiviert werden. Dazu im Office-Menü die **Word-Optionen** öffnen und in der Kategorie **Häufig verwendet** die Option **Entwicklerregisterkarte in der Multifunktionsleiste anzeigen** aktivieren. Nun aktiviert man auf der Registerkarte **Entwicklertools** den **Entwurfsmodus**.



Um KScript/KPath-Ausdrücke einzufügen, fügt man ein **Nur-Text-Inhaltssteuerelement** ein. Der Text des Steuerelements wird durch den berechneten Text ersetzt. Bei den Eigenschaften des Steuerelements (erreichbar über das Kontextmenü auf der schließenden Klammer) gibt man bei **Titel** das KScript bzw. den KPath an. Falls man den Titel leer lässt, wird stattdessen der Text des Steuerelements verwendet. Als **Tag** gibt man den Script-Typ an. Als Script-Typen gibt es:

- **KPath** : analog zu **KPATH_EXPAND**
- **KScript** : analog zu **KSCRIPT_EXPAND**
- **KPathRows** : analog zu **KPATH_ROWS**



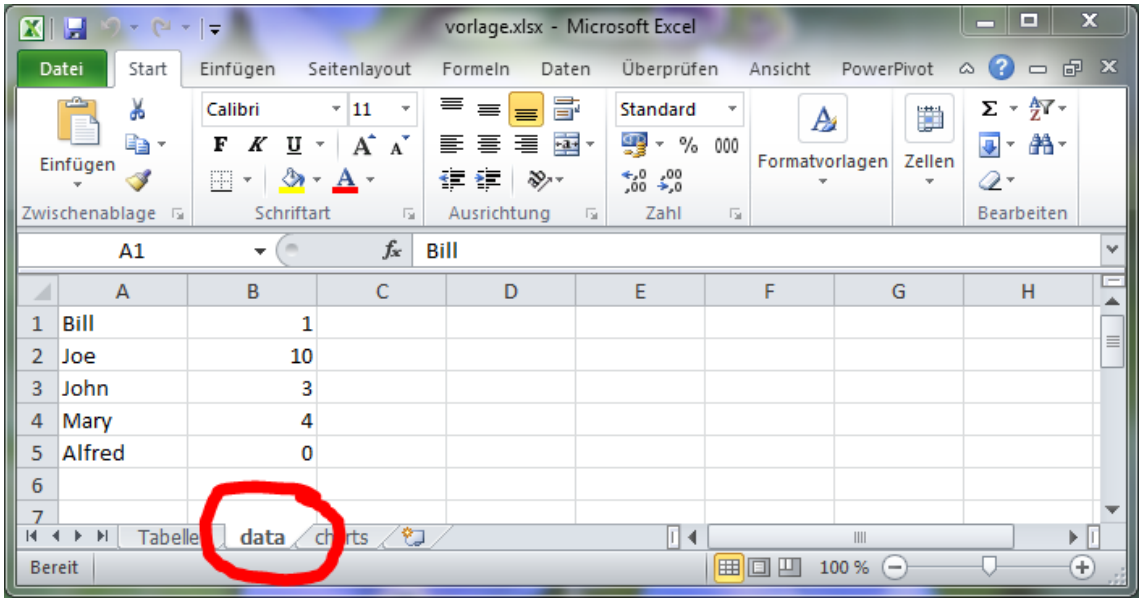
Steuerelement mit einem KPath-Ausdruck

7.3.4.6 XLSX-Dokumente (Microsoft Excel) erstellen

XLSX-Vorlagen können mit Microsoft Excel 2007 oder neuer erstellt werden. Diese Vorlagen funktionieren nur mit Objektlisten.

Erstellen der Excel-Datei

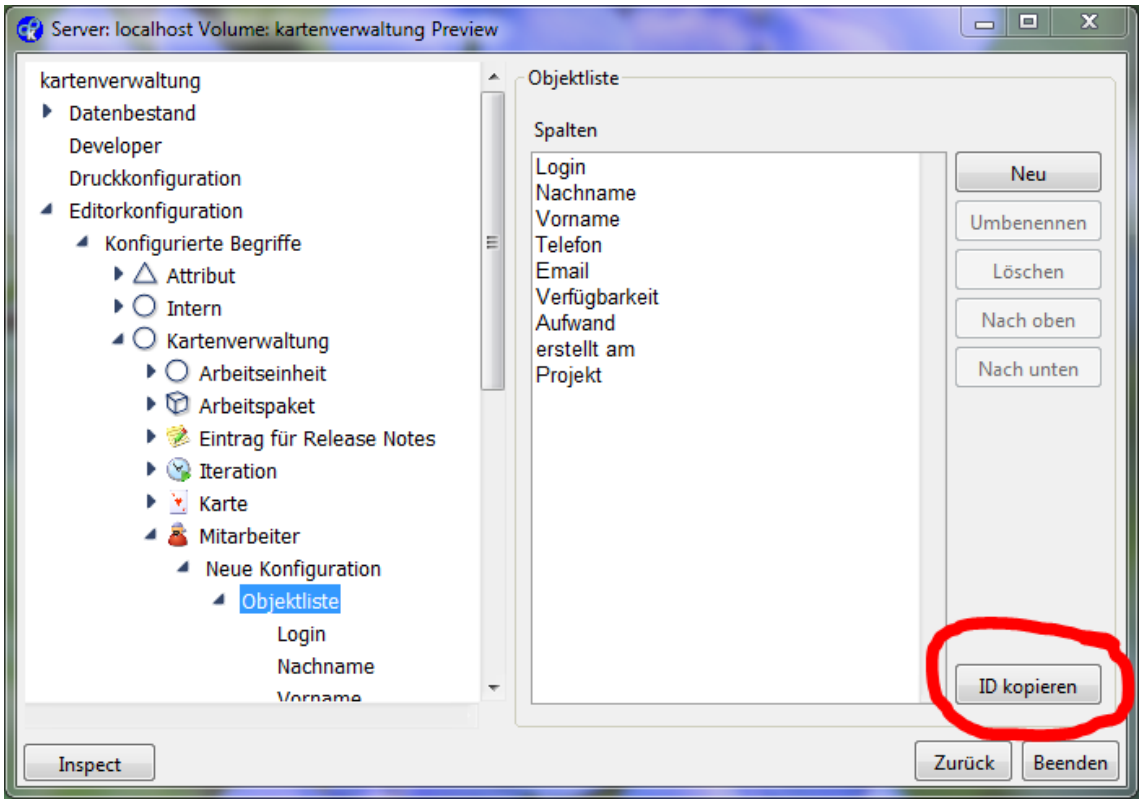
Als Vorlage dient eine gewöhnliche Excel-Datei, die ein zusätzliches Arbeitsblatt namens "data" enthalten muss. Die Objektlistendaten werden später dann in dieses Arbeitsblatt gefüllt und zwar ohne Überschriften und beginnend mit der Zelle A1.



Die anderen Arbeitsblätter können Daten aus dem Blatt "data" in Formeln referenzieren. K-Infinity sorgt dafür, dass alle Formeln neu berechnet werden, sobald die ausgefüllte Excel-Datei das nächste Mal mit Excel geöffnet wird.

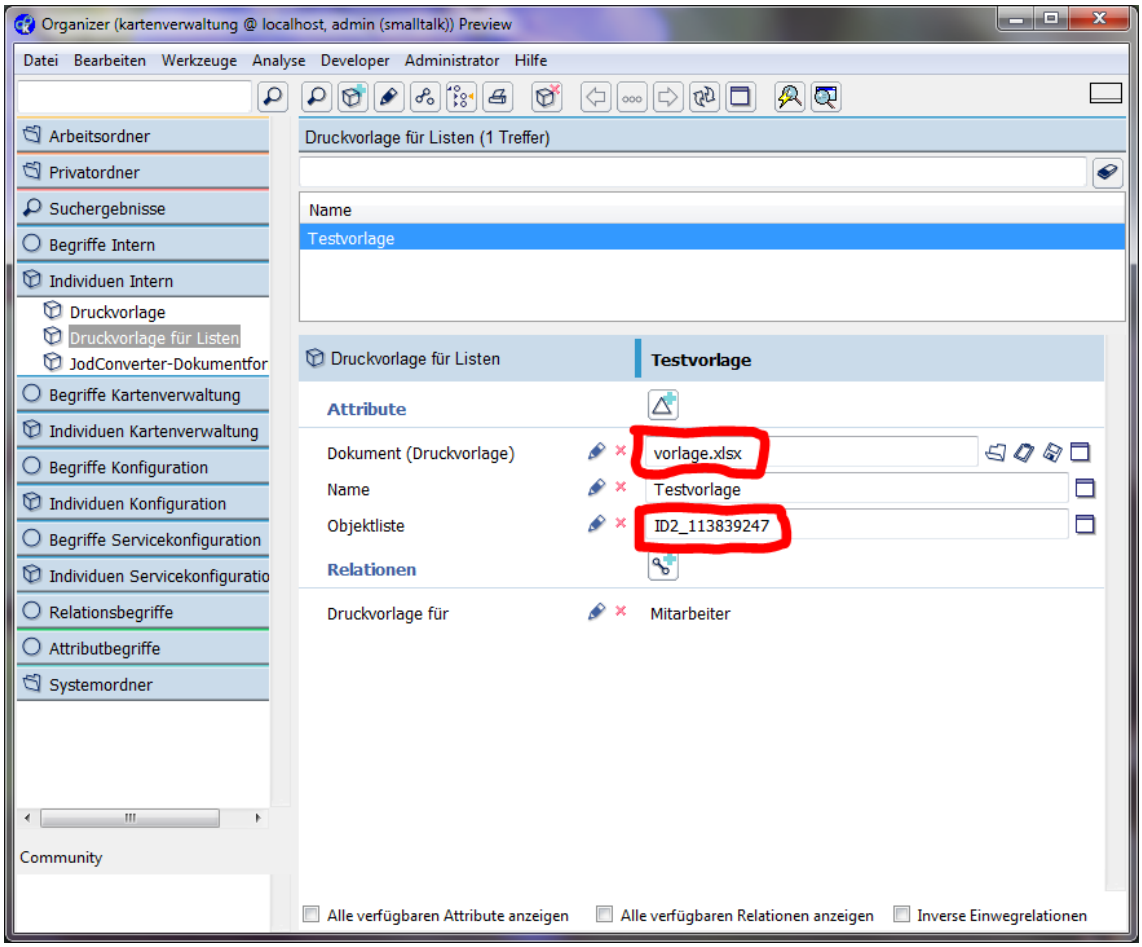
Erstellen der Konfiguration in K-Infinity

Zunächst wählt man im Admin-Tool eine geeignete Objektlistenkonfiguration aus oder erstellt eine neue. Über die Funktion "ID kopieren" überträgt man dann die ID der Konfiguration in die Zwischenablage (diese wird im nächsten Schritt benötigt).



Schließlich erstellt man im Knowledge-Builder eine neue "Druckvorlage für Listen" und be-

füllt das Attribut "Objektliste" mit der ID aus der Zwischenablage. In das Attribut "Dokument (Druckvorlage)" lädt man die in Schritt 1 erstellte Excel-Datei hoch. Der Name der Druckvorlage kann frei vergeben werden und über die Relation "Druckvorlage für" kann man der Benutzungsschnittstelle einen Hinweis geben, für welche Individuen die Vorlage angeboten wird.



Wenn man das Attribut "Dokument (Druckvorlage)" an dem Individuum nicht angelegt hat, so wird bei der Dokumentgenerierung eine Excel-Datei erzeugt, die ein Arbeitsblatt mit den Daten der Objektliste und den Spaltenüberschriften aus der Objektlistenkonfiguration enthält, d.h. man muss nicht zwangsläufig eine Excel-Datei als Druckvorlage angeben.

7.3.4.7 Dokumentformatkonvertierung mit Open/LibreOffice

Das Ausgabeformat des Druckvorgangs entspricht dem des verwendeten Templates. Möchte man ein anderes Ausgabeformat erhalten, muss man einen Konverter einrichten.

Dazu benötigt man eine Installation von Libre- oder OpenOffice ab Version 4.0 auf dem Rechner, der die Konvertierung durchführen soll - gewöhnlich dort, wo die Bridge oder der Jobclient läuft, der auch den Druckvorgang durchführt.

Zusätzlich muss in der Konfigurationsdatei (bridge.ini, jobclient.ini, etc.) der Pfad zum "soffice"-Programm angegeben werden, welches Teil der Libre/OpenOffice-Installation ist und sich dort im Unterverzeichnis "program" befindet.

```
[file-format-conversion]
```

```
sofficePath=C:\Program Files (x86)\LibreOffice 4.0\program\soffice.exe
```

7.3.4.8 JOD-Konvertierungsservice

Möchte man andere Ausgabe-Dokumentformate als die direkt unterstützten haben, so kann man dies in der Regel mit dem Konvertierungsservice JOD-Konverter erreichen. Dieser Dienst benötigt OpenOffice und ist als OpenSource-Software erhältlich unter

```
http://www.artofsolving.com/opensource/jodconverter
```

Hinweis: Getestet wurde die Version 2.2.2

Der JODConverter kann auf einem beliebigen Server installiert werden. Die Installation von OpenOffice auf allen Client-Rechnern ist nicht notwendig. Im Admin-Tool trägt man unter Systemkonfiguration/Komponenten/Konvertierungsservice ein, über welche Adresse der Konvertierungsservice erreichbar ist.

Beispiel

```
http://jodconverter:8080/jodconverter/service
```

Alternative: Der JOD-Konverter kann durch die K-Infinity-REST-Bride ersetzt werden (siehe Kapitel "Formatkonvertiertung").

Dokument-Formate

Damit die Ausgabeformate verfügbar sind, müssen entsprechend konfigurierte Individuen im Netz vorhanden sein. Die Schaltfläche "Schemaupgrade" des Abschnittes "Konvertierungsservice" im Admin-Tool richtet das hierfür nötige Schema ein. Es gibt den Begriffe "JodConverter-Dokumentformat" (interner Name: "jodConverterDocumentFormat"), dessen Individuen die Zeichenketten-Attribute "file-extension" und "mime-type" (interner Name "jodConverterFileExtension" bzw. "jodConverterMimeType") sowie das boolsche Attribut "Für Druckausgabe verwenden" (interner Name "jodConverterUseInPrinting") tragen können.

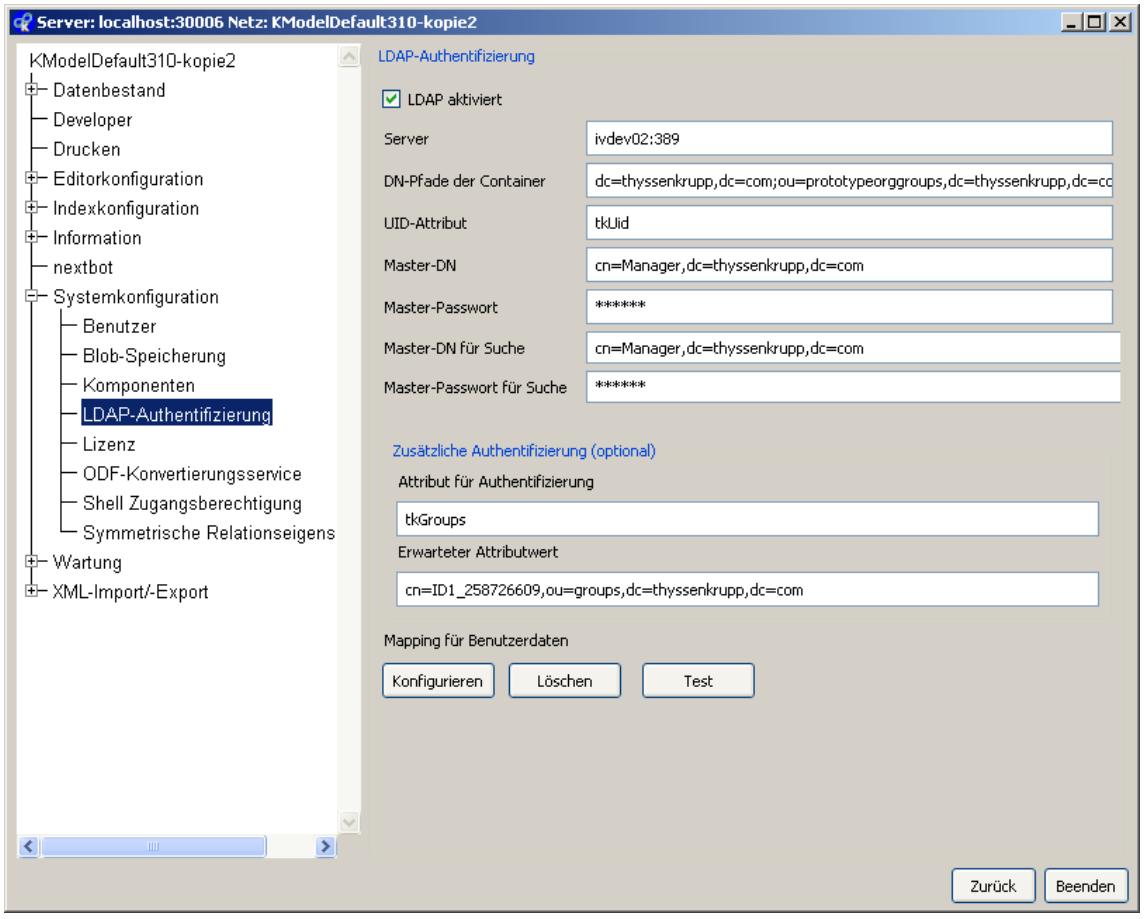
Die passenden Werte für file-extension und mime-type stehen in der XML-Datei "documentformats.xml", die im JodConverter enthalten ist. Wenn das Flag "Für Druckausgabe verwenden" gesetzt ist, wird in der Druckausgabe der Name des zugehörigen Dokumentformates als zusätzliches Ausgabeformat angeboten.

Wichtig ist, dass nicht alle Formate ineinander Konvertiert werden können. Die wichtigsten sind:

Name	Extension	Mime-Type
Portable Document Format	pdf	application/pdf
OpenDocument Text	odt	application/vnd.oasis.opendocument.text
Microsoft Word	doc	application/msword

7.4 LDAP-Authentifizierung

In K-Infinity kann ein LDAP-Server zur Authentifizierung eingebunden werden. Dort eventuell bereits vorliegende Kennungsdaten werden abgefragt, zusätzlich können in LDAP vorhandene Benutzerdaten (Gruppenzugehörigkeiten, Kontaktdaten etc.) bei jedem Login ins Wissensnetz üb werden.



LDAP-Konfigurationsreiter

Zuallererst muss der Reiter LDAP angewählt werden und der Haken im Kästchen LDAP aktiviert gesetzt werden. Damit ist ausgedrückt, dass die eingegebenen Kennungsdaten durch den konfigurierten LDAP-Server zu prüfen sind. Die Angaben zum Server trifft man im Eingabefeld Server , z.B. host:389 . Die durch Doppelpunkt abgetrennte Zahl ist die Portnummer, unter der der LDAP-Server abgefragt werden soll, für LDAP typischerweise 389.

Um einen Umstieg auf einen anderen Server zu erleichtern, ist die Pfad-Angabe zum Abfragen desjenigen Attributs der Benutzerdaten in der LDAP-Datenbank in zwei Teile aufgespalten, einen absoluten Suffix-Anteil und einen Anteil, der relativ dazu noch vorangefügt wird. Es sind mehrere absolute Suffix-Anteile angebbar, dies geschieht durch Semikola getrennt im Feld "DN-Pfade der Container".

Ein Beispiel:

Die Kennungsdaten von Benutzern werden mit der Anfrage

userid=username,ou=user,dc=company,dc=com

vom LDAP-Server für einen Benutzer namens username geholt. Dabei ist userid das At-

tribut, in welches der bei der Anmeldung eingegebene Benutzername einzusetzen ist, d.h. dieser Attributname, nämlich userid , wird in das Eingabefeld UID-Attribut eingetragen, und ou=user,dc=company,dc=com ist der Verzeichnispfad, über den man an Benutzerindividuen in der LDAP-Verzeichnisstruktur des LDAP-Servers gelangt, woraus folgt, dass dieser Pfad in das Eingabefeld DN-Pfad einzugeben ist. Zusätzlich wird das eingegebene Passwort übermittelt. So kann die Abfrage im Falle eines Server- oder Verzeichniswechsel flexibel angepasst werden.

Falls nur bestimmten im LDAP-Verzeichnis eingetragenen Benutzern der Zugriff auf den LDAP-Server (und damit der Login) erlaubt werden soll, kann man den Namen eines Benutzer-Attributs in das Eingabefeld Attribut für Authentifizierung eintragen und den für alle zulässigen Benutzer erwarteten Attributwert in das Eingabefeld Erwarteter Attributwert darunter. Hierbei ist zu beachten, in welcher Form der Attributwert von der LDAP-Abfrage geliefert wird.

Für die Benutzerauthentifizierung unter KEM steht noch ein weiterer Mechanismus zur Verfügung: die Suche nach dem passenden DN durch eine Instanz im LDAP-Verzeichnis, die Sucherlaubnis für die entsprechenden Bereiche besitzt. Der DN und das Passwort für diese Instanz sind in den Feldern "Master-DN für Suche" und "Masterpasswort für Suche" anzugeben.

Abschliessend gibt es noch die Möglichkeit, für den Zugriff auf LDAP einen "trusted user" zu verwenden. Dazu muss ein Benutzer mit entsprechenden Rechten im LDAP angelegt werden, der die Daten aller zu authentifizierenden Benutzer lesen kann. Login und Passwort für den "trusted user" sind in den Feldern "Master-DN" und "Master-Passwort" anzugeben.

Abgleich von Benutzerdaten mit Informationen aus LDAP

Nach erfolgter Authentifizierung wird das mit dem LDAP-Benutzer verknüpfte Benutzer-Individuum im Wissensnetz gemäß einer Import-Konfiguration aus den LDAP-Verzeichnisdaten aktualisiert. Diese Konfiguration muss zumindest die Abbildung des eingegebenen Kennungsnamens auf ein Benutzerindividuum ermöglichen, also z.B. eine Attributabbildung, die für die Identifizierung des Individuums herangezogen werden soll, z.B. für den Fall, dass nicht das Namensattribut verwendet werden soll. Darüber hinaus stehen alle Mapping-Funktionen des Tabellen-Imports zur Verfügung, so dass z.B. aus LDAP-Daten heraus Attribute und Beziehungen angelegt und aktualisiert werden können.

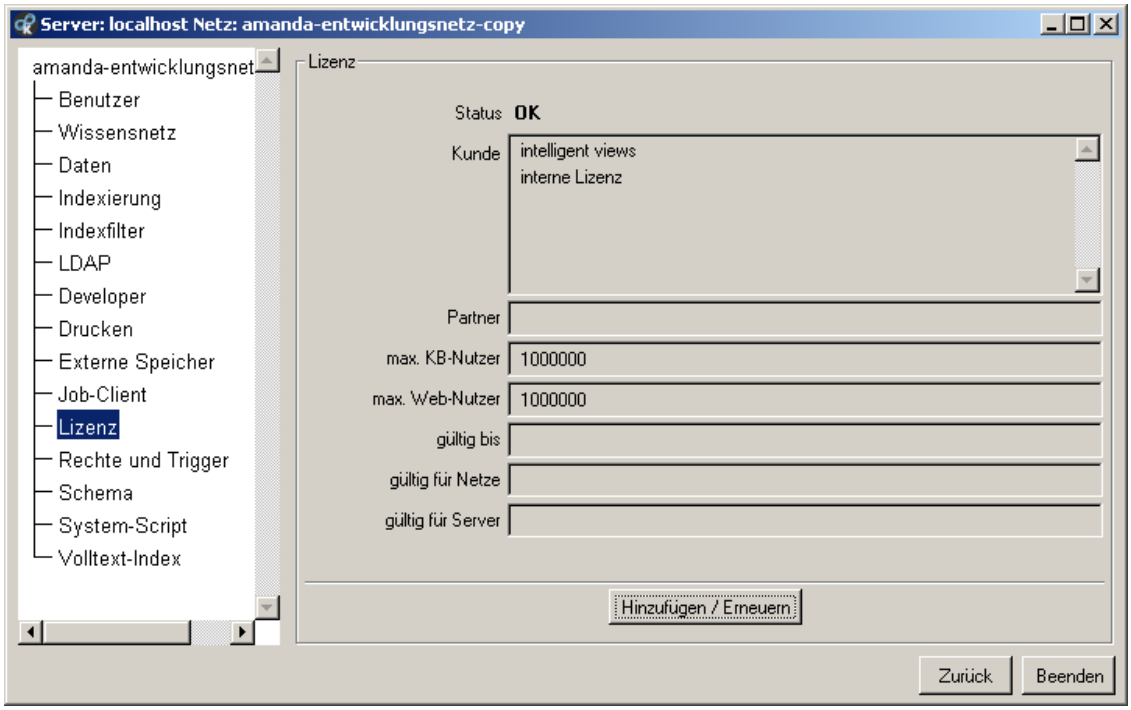
Ist die Konfiguration abgeschlossen, kann man ihre Funktion durch Anklicken des Knopfs Test überprüfen. Es erfolgt eine Abfrage von Kennung und Passwort und die Rückmeldung, ob eine Authentifizierung erfolgreich war oder nicht.

7.5 Lizenz

Zusammen mit einer Version von K-Infinity wurde Ihnen ein Lizenzierungsschlüssel ausgehändigt, mit dessen Hilfe Sie Ihr Wissensnetz lizenzieren können. Der Schlüssel besteht in Form einer Lizenzdatei mit der Endung .key. Diese Datei kann über das Admin-Tool in das Wissensnetz, das lizenziert werden soll, eingespielt werden. Alle betroffenen Werkzeuge, mit denen das Wissensnetz bearbeitet werden soll, überprüfen die im Netz gespeicherte Lizenz und geben entsprechende Fehlermeldungen aus. Netze ohne Lizenz sind nicht benutzbar.

- Stellen Sie sicher, dass Sie über eine aktuelle key-Datei verfügen
- Fügen Sie über den Button Hinzufügen/Entfernen diese Lizenz Ihrem Wissensnetz hinzu.
- Falls der Status »OK« angezeigt wird, war die Lizenzierung erfolgreich.

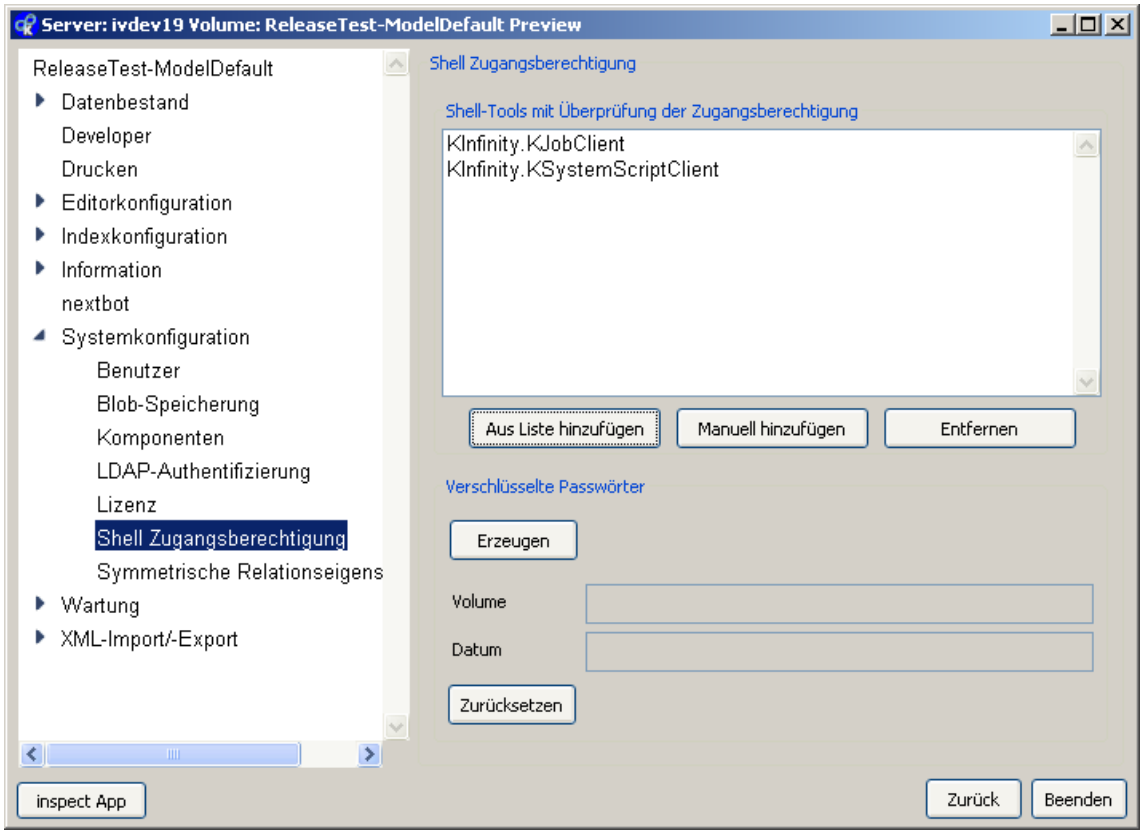
Die anderen Felder zeigen weitere Informationen zu Ihrer Lizenz an.



Gültige Lizenz

7.6 Shell Zugangsberechtigung

Diese Konfigurationsseite bietet die Möglichkeit die Zugangsberechtigung der verschiedenen Clients zu steuern und die Job-Clients im Speziellen zu überwachen.



Zugangsberechtigung zu den Clients

Die Liste *Mit Überprüfung der Zugangsberechtigung* zeigt die Klassennamen mit deren Namespace der Clients, die beim Starten eine Überprüfung der Zugangsberechtigung vornehmen. Die Clients müssen dann mit den Parametern *-user* und *-password* gestartet werden.

Erläuterung der Knöpfe:

- *Aus Liste hinzufügen:* In dieser Liste sind die Standard-Clients enthalten, die hinzugefügt werden können.
- *Manuell hinzufügen:* Mit diesem Knopf lassen sich kundenspezifische Clients hinzufügen. Syntax: <*Namespace*>.<*Klassennamen*> Anmerkung: Dem AdminTool sind die Klassennamen nicht bekannt. Somit findet an dieser Stelle auch keine Validierung statt. Es sollte beim Start des Clients überprüft werden, ob der Mechanismus der Überprüfung der Zugangsberechtigung funktioniert.
- *Entfernen:* Der selektierte Client wird aus der Liste entfernt.

Verschlüsselte Passwörter

Passwörter für die Clients können verschlüsselt in einer Datei abegspeichert werden. Auf diese Weise muss das Passwort nicht sichtbar beim Starten angegeben werden. Die Clients werden dann mit den Parametern *-user* und *-passwordFile* gestartet.

Erläuterung der Knöpfe:

- *Erzeugen:* Für einen auszuwählenden Benutzer wird eine Passwortdatei <*Dateiname*>.auth erzeugt.
- *Zurücksetzen:* Ein neuer Schlüssel wird erzeugt. Dadurch werden alle bisher erzeugten

Passwortdateien ungültig.

7.7 RSA Anwendungskontrolle

Mit Hilfe der RSA Anwendungskontrolle lässt sich konfigurieren, welcher Client mit welchem Mediator einen Verbindungsaufbau durchführen darf. Dabei lässt sich für beide Richtungen eine Einschränkung festlegen. Beim Meditor wird konfiguriert, welche Clients sich zu ihm verbinden dürfen und beim Client wird eingestellt mit welchem oder welchen Mediatoren ein Verbindungsaufbau erlaubt ist.

Jede Version jeder Anwendung hat ein einmaliges RSA-Schlüsselpaar. Der öffentliche Teil kann abgefragt werden. Bei Anwendungen ohne Benutzeroberfläche geschieht dies über den Kommandozeilenparameter "**-showBuildID**". Mit Benutzeroberfläche kann im Info-Fenster mit "RSA-Key kopieren" die zugehörige Ausgabe in die Zwischenablage kopiert werden.

Wenn keine Schlüssel konfiguriert werden, sind alle BuildIDs erlaubt. Ansonsten müssen diese in der zur Anwendung gehörenden Konfigurationsdatei eingetragen werden.

Verbindet sich ein Client mit dem Mediator, so führen diese einen Handshake aus. Dabei wird jeweils überprüft, ob die Gegenseite den zur BuildID passenden privaten Schlüssel kennt (RSA-signierte Antwort auf Einmal-Anfrage), zum anderen wird der Handshake nur ausgeführt, wenn die Gegenseite sich mit einer erlaubten buildID meldet. Bei falschen Schlüsseln wird die Verbindung abgebrochen.

Möchte man zum Beispiel, dass der KB sich mit keinem anderer Mediator als dem Build 09032308 verbindet, so kopiert man die Ausgabe des Mediators mit obiger Kommandozeile in die kb.ini:

```
[buildID.76102B7C2E9BE00C0834B94B30FD2993]
build=Build 09032308
rsa.n_1=56FA70A3AABFF20AD0A38D4FBA5CDD09
rsa.n_2=D49738B323DC29D9EFE34597261C3D2E
rsa.n_3=E639B40656389FBC2DA40621719C8189
rsa.n_4=5202281449A20010A2010000800FFFFF
rsa.n_5=FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
rsa.n_6=FFFFFFFFEE89164123960094A0C08269
rsa.n_7=80401101000001FFFFCA1362143E693F
rsa.n_8=19CB8EBF0B21A19EEA0B23AD25D50F1B
rsa.n_9=30915AEA636A424CA3CEA52790FDB04E
rsa.n_10=1389EC501CF7AC8CC363B730A6118128
rsa.n_11=EAA460061F551EC00000000000000000
rsa.n_12=00000000000000000000000000000000
rsa.n_13=00000410000011000040A000300004048
rsa.n_14=1003CF90E0000010441001CB171842AE
rsa.n_15=78900A2DAE330FCD0D4F596BBA0D5B8C
rsa.n_16=82DC94946D46104F1BFE4877056D20DD
rsa.e_1=010001
```

Dabei kann man mehrere dieser Schlüssel hinzufügen, die zusätzliche Information "**build=...**" dient nur dem leichteren Zuordnen und hat keinerlei technische Auswirkung, kann also beliebig angepasst werden, um unterschiedliche Anwendungen genauer zu beschreiben.
Analog kann man die Konfigurations-Dateien weiterer Clients (jobclient.ini u.s.w.) für diesen Mediator anpassen. Versucht sich nun ein Client mit einem anderen Mediator zu verbinden,

so bricht der Client mit einer Fehlermeldung (unknown fingerprint) ab.

Man kann am Mediator in der Datei "mediator.ini" auf die gleiche Weise die Clients eintragen, die sich beim Mediator anmelden dürfen. Andere Clients erhalten dann ebenfalls eine Fehlermeldung beim Anmeldeversuch.

Möchte man diese Funktion deaktivieren, ohne alle Schlüssel auszutragen, so fügt man folgenden Eintrag in die Konfigurationsdatei ein:

```
[buildID.00000000000000000000000000000000]
```

Die Zugangskontrolle für Anwendungen kann bei Clients und Server unabhängig voneinander aktiviert oder deaktiviert werden.

7.8 Symmetrische Relationseigenschaften

An dieser Stelle gibt es keine Konfigurationmöglichkeit. Es wird nur darüber informiert, ob Relationseigenschaften auf symmetrisch geschaltet sind oder nicht.

Beim Neuanlegen eines Netzes hat der Administrator die Möglichkeit diese Auswahl zu treffen.

Nachträglich lässt sich mit Hilfe des Wartungsskriptes *switchToSymmetricRelationProperties* dieser Zustand verändern. Dieser Vorgang kan je nach Netzgröße sehr lange dauern.

Symmetrische Relationseigenschaften: ja oder nein?

Bei Relationen wird zwischen Hin- und Rückrichtung unterschieden. An den beiden Richtungen können nun verschiedene Eigenschaften (Meta-Informationen) angebracht werden. Dann dürfen die Relationseigenschaften nicht auf symmetrisch geschaltet sein.

Wird allerdings auf bessere Performance bei der Suche Wert gelegt und es kommen unterschiedliche Relationseigenschaften auf Hin- und Gegenrichtung nicht vor, dann sollten die Relationseigenschaften auf symmetrisch geschaltet sein.

8 Wartung

8.1 Client-Caches

Mit *Client-Caches zurücksetzen* lassen sich alle BlockCaches bei allen Clients zurücksetzen. Dies ist z.B. nötig, um nach Änderungen an der Editor-Konfiguration die Clients anzuweisen, diese neu zu laden und so die Änderungen zu berücksichtigen.

8.2 Garbage Collection

Garbage Collection auf dem Server ausführen.

- **Start**
  Starten der Garbage Collection
  Ein Weiterarbeiten auf dem Netz ist praktisch möglich. Allerdings verlängert sich dadurch die Zeit bis dieser Vorgang abgeschlossen ist. Zu einer extremen Verängerung kann es kommen, wenn viele Schreibzugriffe vorgenommen werden.

- **Pause**
  Die Garbage Collection wird angehalten und mit Start wieder fortgesetzt.
  Diese Funktion sollte kurz die Möglichkeit geben, die Systembelastung durch die Garbage Collection auszuschalten.
- **Anhalten**
  Die Garbage Collection wird unterbrochen

- **Information - Aktualisieren**
  Anzeige des aktuellen Fortschritts

Tip:

Gelegentlich sollte die Garbage Collection auf dem Server ohne jegliche verbundenen Clients gestartet werden.

8.3 Leistung

Im Abschnitt **Client** können Leistungsdaten der angemeldeten Clients (Bridge, Jobclient, KBs u.s.w.) erfasst werden. Dies wird durch die Option "Client-Leistungsdaten aufzeichnen" aktiviert bzw. deaktiviert. Über das Intervall kann man festlegen, wie häufig die Clients ihre Leistungsdaten übermitteln.

Achtung: Die Leistungsaufzeichnung wird das Gesamtsystem belasten und sollte nicht längere Zeit aktiviert bleiben.

Im Abschnitt **Server** können Netzwerk-Leistungsdaten des Mediators gemessen werden.

8.4 Wartungsinformation

In den Wartungsinformationen landen zum Einen Meldungen über Systemänderungen, von der Software erzeugt, und zum Anderen kann der Administrator eigene Kommentare hinzufügen.

Die Einträge haben immer das gleiche Format:

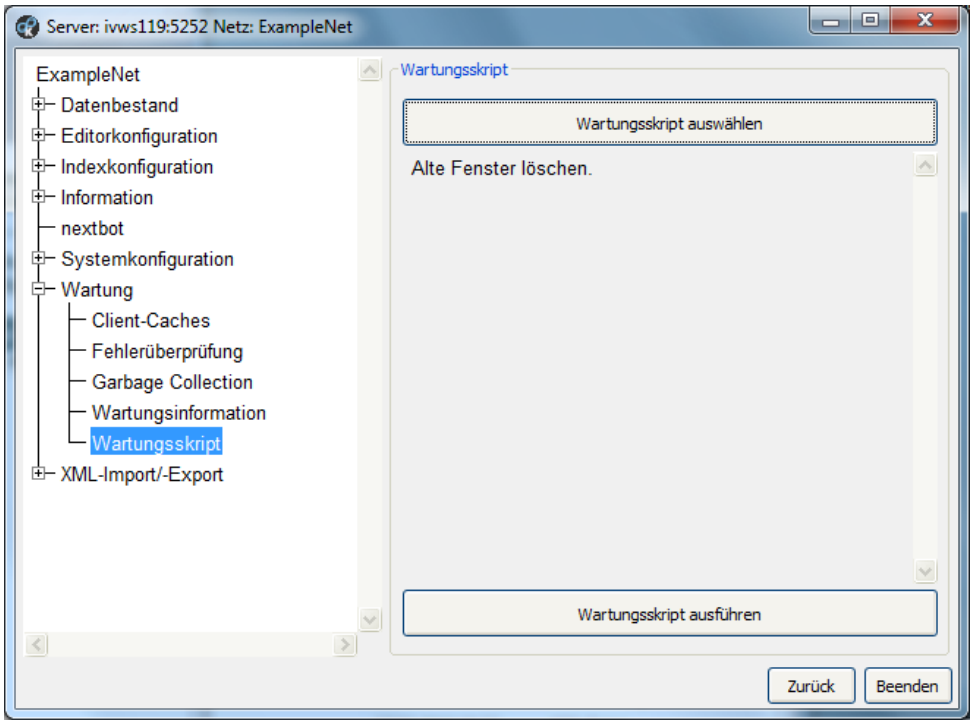
YYYY-MM-DD-HH-MM-SS >> <Kommentar>

Meldungen können sein:

- Updated from 3.0.18 to 3.1.0
- SystemScript performed:> #entitiesUebernehmen
- Add component: Release
- Remove component: Release
- K-Infinity-Core 3.1.0 (Auf 3.1.1 aktualisieren) - Update fertig
- Knowledge-Builder 3.1.0 (Auf 3.1.1 aktualisieren) - Fertig
- Updated from 3.1.0 to 3.2.0

8.5 Wartungsskript

Mit Hilfe von Wartungsskripten kann ein von intelligent views generierter Code-Baustein zur Wartung des Netzes ausgeführt werden. Mit „Wartungsskript auswählen“ wird ein Wartungsskript (Dateiendung kss) ausgewählt. Nach der Auswahl wird im Textfeld darunter eine Beschreibung des Skripts angezeigt. Mit „Wartungsskript ausführen“ wird das Skript dann ausgeführt.



Benutzerschnittstelle für das Wartungsskript

Anmerkung:

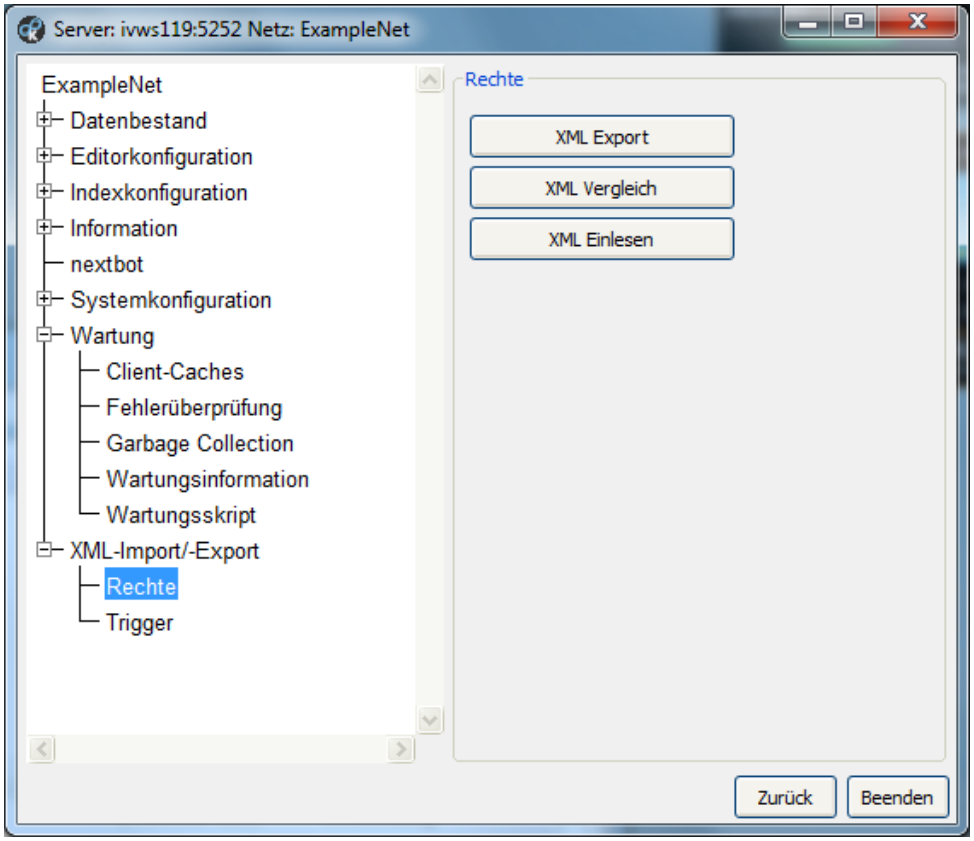
System-Skript wurde in Wartungsskript unbenannt. Die Dateiendung kss (für kinfinty system script) bleibt bestehen.

9 XML-Import/-Export

In dieser Rubrik befindet sich die Funktionalität, um Teile der Konfiguration zu speichern und wieder einzulesen.

9.1 Rechte und Trigger

Es können Rechte und Trigger per XML exportiert und importiert werden.



Benutzerschnittstelle für Im-/Export von Rechtekonfiguration

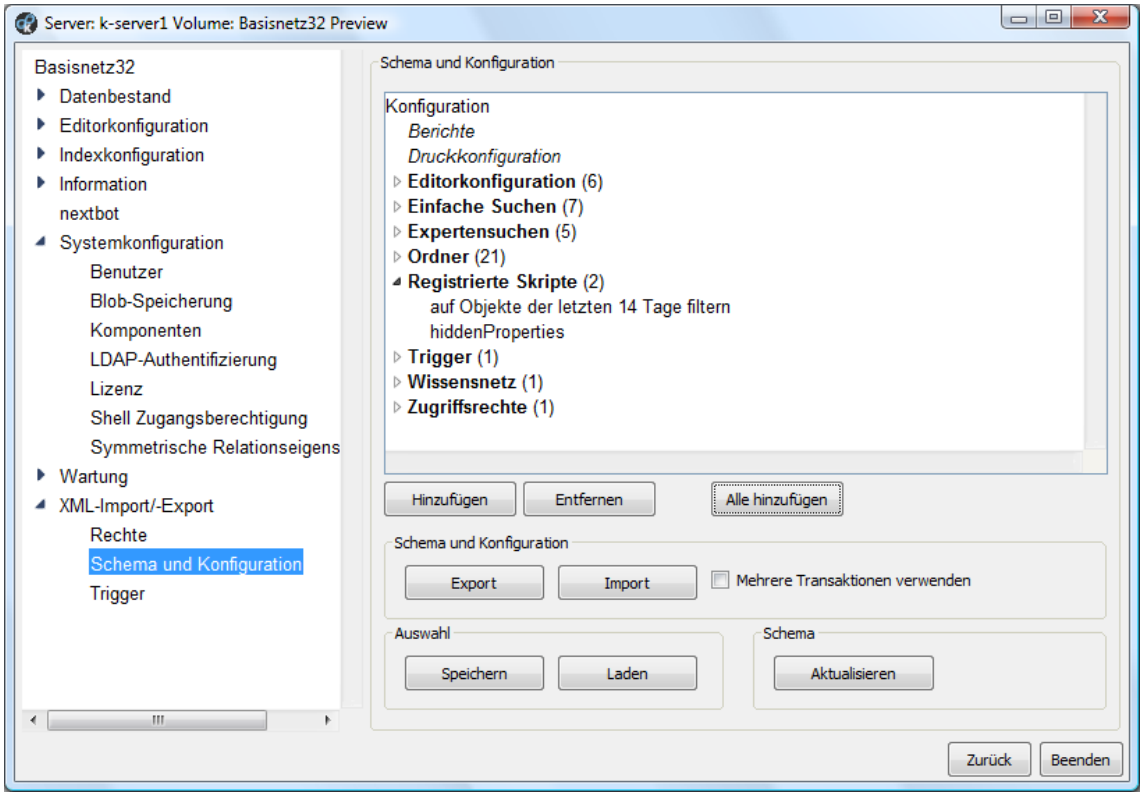
Der Funktionsumfang für Rechte und Trigger ist gleich.

Beim Export und Vergleich lassen sich Teilbäume auswählen.

9.2 Schema und Konfiguration

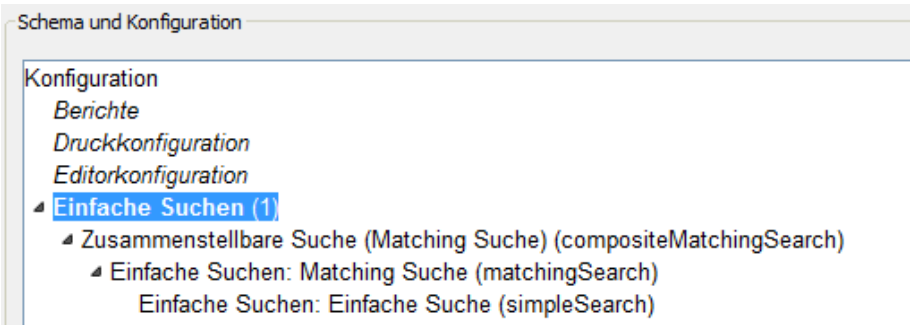
9.2.1 Export

Das Schema des Wissensnetzes und Konfigurationen können als XML-Dateien exportiert und importiert werden. Damit können diese auf andere Netze übertragen oder neue Netze aufgebaut werden.



Export von Schema und Konfiguration

In der Baumansicht **Konfiguration** werden die exportierbaren Objektklassen und die zu exportierenden Objekte angezeigt. Falls ein Objekte andere Objekte benötigt, werden diese unterhalb des Objektes angezeigt und automatisch mitexportiert. Im folgenden Beispiel verwendet eine zusammenstellbare Suche eine Matching-Suche, die eine einfache Suche verwendet.



Abhängigkeit einer Suchkonfiguration

Mit **Hinzufügen** kann man zu exportierende Objekte auswählen. **Alle hinzufügen** fügt automatisch alle exportierbaren Objekte hinzu.

Es gelten folgende Einschränkungen und Besonderheiten:

- **Berichte**: Der Berichtsordner muss eine externe ID haben und im öffentlichen Arbeitsordner liegen.
- **Druckkonfiguration**: Keine Einschränkungen.
- **Editorkonfiguration:** Keine Einschränkungen.

- **Einfache Suchen**: Fehlerhaft konfgurierte oder nicht eindeutig benannte Suchen können nicht exportiert werden. Falls eine Suche andere Suchen referenziert, werden diese automatisch mitexportiert.
- **Expertensuchen**: Expertensuche müssen eine externe ID haben und im öffentlichen Arbeitsordner liegen.
- **Ordner**: Es können nur "normale" Ordner exportiert werden. Ordner müssen eine externe ID haben und im öffentlichen Arbeitsordner liegen. Unterordner werden automatisch mitexportiert, auch wenn sie keine externe ID haben.
- **Registrierte Skripte**: Falls ein Skript Suchen aufruft oder andere Skripte referenziert, werden diese mitexportiert.
- **Trigger**: Es kann nur die gesamte Triggerdefinition exportiert werden.
- **Wissensnetz**: Siehe folgenden Abschnitt "Transfer des Wissensnetzschemas"
- **Zugriffsrechte**: Es kann nur die gesamte Rechtedefinition exportiert werden.

Mit **Export** können die ausgewählten Objekte in eine Archiv-Datei (TAR) exportiert werden. Dieses Archiv beinhaltet für jedes Objekt eine XML-Datei sowie eine zusätzliche Übersichtsdatei (instruction.xml). Das Archiv kann u.A. unter Unix mit tar und unter Windows mit 7-Zip entpackt werden.

Die **Auswahl** der Schema- und Konfigurationsdatei kann in eine Datei gespeichert und später wieder geladen werden.

9.2.2 Import

Mit **Import** kann eine Archivdatei eingelesen und die enthaltenen Schema- und Konfigurationsobjekte in das Wissensnetz importiert werden.

Vor dem Import sollte auf jeden Fall ein Backup des Netzes angelegt werden, da potentiell viele Änderungen vorgenommen werden.

Wenn **Mehrere Transaktion verwenden** angehakt ist, werden die Objekte in einzelnen Transaktionen importiert. Bei einem Abbruch des Imports bleiben bereits importierte Objekte dann bestehen. Dafür ist der Import bei größeren Konfigurationsdateien wesentlich schneller.

Bereits vorhandene Objekte werden durch die importierten Objekte ersetzt. Zur Identifizierungen werden je nach Objekt unterschiedliche identifizierende Eigenschaften verwendet:

- **Berichte**: Externe ID des Ordners + Name des Berichts.
- **Editorkonfiguration**: Name + der konfigurierte Begriff
- **Einfache Suchen**: Interner Name der Suche, oder Name falls eindeutig.
- **Expertensuchen** und **Ordner**: Externe ID.
- **Registrierte Skripte**: Registrierter Name des Skripts.
- **Wissensnetz**: RDF-ID, KPath und interne Objekt-ID.

Druckkonfiguration, Trigger und Zugriffsrechte werden global ersetzt.

9.2.3 Transfer des Wissensnetzschemas

Beim Export des Wissensnetzschemas werden alle Begriffe, aber keine Individuen exportiert. Im Admin-Tool kann man diese Auswahl nicht beeinflussen. Es ist aber möglich, im Knowledge-Builder die Auswahl einzugrenzen bzw. zu erweitern.

Dazu muss man im Admin-Tool mit **Schema aktualisieren** zusätzliche Boolsche Attribute definieren, die bei Begriffen angelegt werden können:

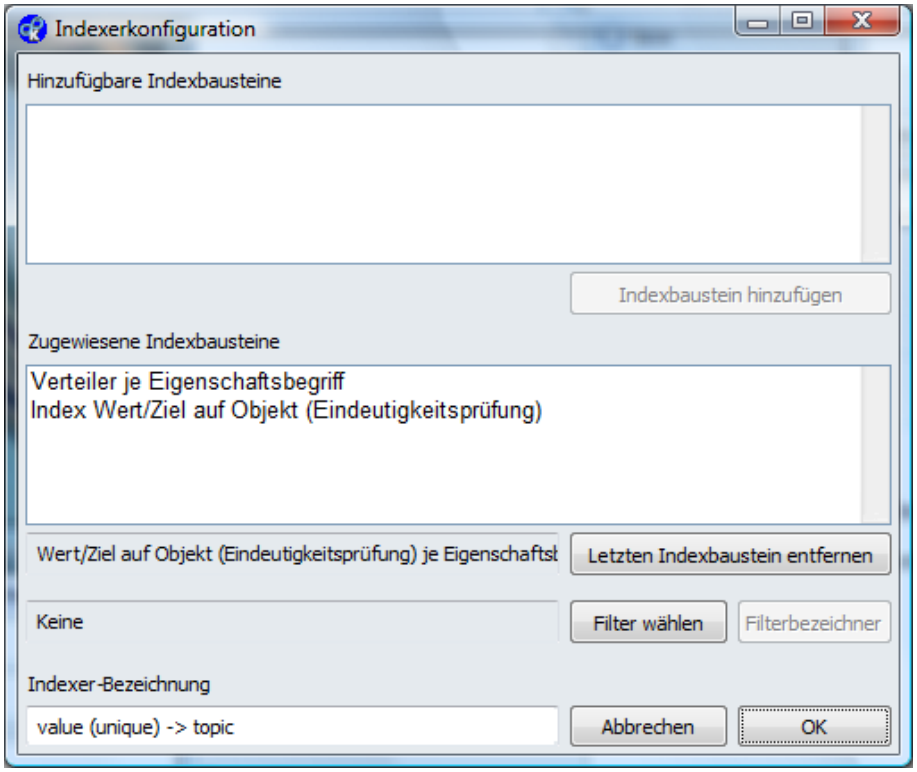
- **XML: Alle Individuen exportieren**: Es werden alle Individuen, inklusive der Individuen von allen Unterbegriffen, exportiert. Dies ist davon unabhängig, ob ein Begriff weitere Oberbegriffe hat, bei denen diese Einstellung nicht gesetzt ist.
- **XML: Direkte Individuen exportieren**: Direkte Individuen des Begriffs exportieren. Individuen von Unterbegriffen werden dadurch nicht beeinflusst. Die Einstellung hat Vorrang vor der Einstellung "Alle Individuen exportieren" von Oberbegriffen.
- **XML: Begriff und alle Unterbegriffe nicht exportieren**: Der Begriff und sämtliche Unterbegriffe werden nicht exportiert. Dies ist davon unabhängig, ob ein Begriff weitere Oberbegriffe hat, bei denen diese Einstellung nicht gesetzt ist.

Für den Transfer gelten folgende Einschränkungen:

- Passwort-Attribute werden grundsätzlich nicht exportiert.
- Datei-Attribute werden zur Zeit nicht exportiert, dies kann sich aber in zukünftigen Versionen noch ändern.

Im exportierten Netz werden rdf:id-Attribute angelegt. Dadurch wird sichergestellt, dass bei wiederholten Exporten die Zuordnung zu semantischen Objekten konsistent zu früheren Exporten ist.

Falls im exportierten Netz rdf:id- oder rdf:about-Attribute definiert sind, müssen diese mit einem Index versehen sein, der die Eindeutigkeit sicherstellt. Dies wird vor dem Export überprüft, der Export ist ohne einen solchen Index nicht möglich. Im Admin-Tool kann man einen solchen Index erstellen, falls noch nicht vorhanden:



Konfigurations des Eindeutigkeits-Index