

Knowledge-Portal



Inhaltsverzeichnis

1 Die wichtigsten Funktionen im Überblick	8
1.1 Information auf einen Blick	8
1.2 Visualisierung mit dem Net-Navigator	8
1.3 Intelligente Suchen	9
1.4 Gruppen- und rollenbasierte Personalisierung	10
1.5 Rechteverwaltung	10
1.6 DMS-Funktionalität	10
1.7 WYSIWYG-Editor	10
1.8 Feedback- und Kommentarfunktion	10
1.9 Abonnementfunktion	11
1.10 Volltextsuche mit statistischer Dokumentklassifikation	11
2 Systemvoraussetzungen	11
3 Überblick über die KP-Konfiguration	12
3.1 Basis-Konfiguration	12
3.2 Anwendungskonfiguration im Netz	14
3.3 Basisnetz	17
3.3.1 Allgemein	17
3.3.2 Änderungen im Basisnetz	18
3.3.3 Benutzer	19
3.4 KP-Wissensnetzkomponente (ab Version 3.7)	20
3.5 XML-Komponentenkonfiguration (ab KP 3.7)	21
3.6 XML-Entwicklungsumgebung	22
4 Login-Konfiguration	23
4.1 Grundkonfiguration Login	23
4.2 Konfigurierbare Parameter	23
4.3 Login-Art	26
4.3.1 Kein Login (type = 0)	26
4.3.2 Login-Box (type = 1)	26
4.3.3 Login-Seite ohne Gastbenutzer (type = 2)	27
4.3.4 Login-Seite mit Gastbenutzer (type = 3)	27
4.4 Startseite	28
4.5 Login-Komponenten	29



4.5.1	StandardLoginComponent	29
4.5.2	NtlmLoginComponent	30
4.5.3	SPNEGO-Login-Komponente (ab KP 3.8)	32
4.6	Logout-Button	39
5	Konfiguration der Suchen	39
5.1	Suchkonfiguration ab Version 3.5	39
5.1.1	Einführung	39
5.1.2	XML-Suchkonfiguration	40
5.1.3	Suchparameter	41
5.1.4	Suchformular	42
5.1.5	Standard-Suchkonfiguration	48
5.1.6	Front-End bezogene Suchkonfiguration	49
5.1.7	Treffererklärungen	52
5.2	Suchkonfiguration in älteren Versionen	54
5.2.1	Einführung	54
5.2.2	Konfiguration der Standard-Begriffssuche	55
5.2.3	Konfiguration der Meinten-Sie-Suche	56
5.2.4	Konfiguration der Dokumenten-Volltextsuche	56
5.2.5	Gruppierende Suchen	60
6	Layout-Konfiguration	60
6.1	Texte konfigurieren	61
6.2	Layout-Kategorien	63
6.3	Anzeigenamen der Layoutkategorien	63
6.4	Darstellung von Boxen	64
6.4.1	Navigation	65
6.4.2	Verlauf	67
6.4.3	Schnellanlage	68
6.4.4	Schnellsuche	69
6.5	Style-Anpassungen	72
6.5.1	CSS-Anpassungen	72
6.5.2	Eigene Dateien in das Webapps-Verzeichnis laden	72
6.5.3	Konfiguration von Icons	73
6.5.4	Themes	74
6.6	Angepasste Net-Navigator-Konfiguration	76
6.7	Konfiguration des Seitenkopfes	76



6.8	Cross-Site-Scripting Filterung (ab KP 3.8.8)	77
7	Detailseite für Begriffe und Individuen	77
7.1	Detailseiten konfigurieren	78
7.1.1	Erstellen einer View-Konfiguration	78
7.1.2	Verfügbare Konfigurationselemente	81
7.2	Ausblenden bestimmter Attribute und Relationen	121
7.2.1	Konfiguration von versteckten Attributen und Relationen (bis KP 3.7)	121
7.2.2	Konfiguration von versteckten Attributen und Relationen (ab KP 3.8)	123
7.3	Trainingsdokumente und Ähnlichkeitssuche (Brainbot)	124
7.3.1	Beschreibung	124
7.3.2	Schwellwert für Anzahl der Trainingsdokumente	125
7.3.3	Konfiguration der Ähnlichkeitssuche	126
7.3.4	Konfiguration der Volltextsuche	126
7.4	Kontextsuchen / Kontextboxen	126
7.4.1	Expertensuchergebnisse in Kontextboxen anzeigen	126
7.4.2	Ergebnisse von KPath-Ausdrücken in Kontextboxen anzeigen	128
7.4.3	Micro-Edit-Boxen (ab Version 3.7.0)	129
7.4.4	Ergebnisse einer Textsuche in einer Kontextbox anzeigen	134
7.4.5	Die Ähnliche-Themen-Kontextbox (Matchings)	135
7.5	HTML-Editor	135
7.5.1	Überblick	135
7.5.2	Bilder-Upload konfigurieren	136
7.6	Konfiguration der Kommentar-Funktion	137
7.6.1	Überblick	137
7.6.2	Konfiguration der Kommentar-Funktion (bis KP 3.7)	138
7.6.3	Konfiguration der Kommentarfunktion ab KP 3.8	139
7.7	Tooltips	140
7.8	Collections - Wissensnetzelemente sammeln und teilen	140
8	Pflegeseiten für Begriffe und Individuen	144
8.1	Erstellen einer Edit-Konfiguration	144
8.1.1	Allgemeines Verhalten	145
8.1.2	Allgemeines Aussehen	146
8.1.3	Attribute	148
8.1.4	Relation	152
8.1.5	RelationDropdown (ab Version 3.8.0)	156



8.1.6	Relation Wizard (ab Version 3.9)	160
8.1.7	AllExtensions	163
8.2	Sperren	165
8.3	"find or create" Funktionalität	166
8.3.1	Netzkonfiguration	166
8.3.2	Anwendungskonfiguration	167
8.4	Hochladen von Dateien mit dem Multi-Upload-Applet	168
8.5	Bearbeiten von Dateien mit dem Change-File-Applet	168
8.6	Tooltips	169
9	Die Collections	169
9.1	Das Sammeln von Objekten	169
9.2	Die Verwaltung von Collections	171
9.3	Kontextinformationen zu Collections	171
10	Ereignisprotokollierung	172
10.1	Überblick	172
10.2	Protokollierung in CSV-Dateien	173
10.3	Protokollierung in eine Datenbank	174
10.3.1	Einführung	174
10.3.2	Aktivieren der Datenbank-Protokollierung (bis KP 3.5)	174
10.3.3	Aktivieren der Datenbank-Protokollierung (ab KP 3.6)	175
10.3.4	Datenbank-Schema	176
11	Problemlösung	178
11.1	Quick Search	178
11.2	Topic-Detailseite	179
11.2.1	Es werden keine Super-/Subkonzepte und/oder Individuen angezeigt	179
11.2.2	Es werden keine Attribute angezeigt	180
12	Referenz	180
12.1	XML-Suchkonfiguration (searchConfig.dtd)	180
12.1.1	Element "SearchConfiguration"	180
12.1.2	Element "SearchDescriptions"	181
12.1.3	Element "Search"	181
12.1.4	Element "QueryParams"	183
12.1.5	Element "ComputedQueryParam"	183
12.1.6	Element "QueryParam"	184



12.1.7 Element "SortOption"	185
12.1.8 Element "UserQueryParams"	186
12.1.9 Element "Property"	186
12.2 XML Suchergebnisseiten-Konfiguration	187
12.2.1 Element "Search"	187
12.2.2 Element ResultViewConfig	188
12.2.3 Element TabGroup	188
12.2.4 Element Tab	189
12.2.5 Element TwoColumnContainer	189
12.2.6 Element ColumnRow	190
12.2.7 Element TwoColumnRow	190
12.2.8 Element LeftColumn	191
12.2.9 Element RightColumn	191
12.2.10Element SearchPartResult	192
12.2.11Element ResultHeader	193
12.2.12Elemente "Actions" / "GroupActions" / "HitActions"	193
12.2.13Element "SortOption"	194
12.2.14Element "SortCriteria"	194
12.3 Suchadapter	195
12.3.1 K-Infinity-Suchadapter (kinfinity)	195
12.4 Übersetzungen	201
12.4.1 Allgemein	201
12.4.2 msg_de.properties	201
12.5 Actions und Action-Transformer	211
12.5.1 Relation erzeugen (CreateRelationAction)	211
12.5.2 Relation mit neuem Relationsziel erzeugen (CreateNewTopicAction)	211
12.5.3 Objekt löschen (DeleteTopicAction)	212
12.5.4 KScript-Report ausführen (DownloadReportAction)	212
12.5.5 Report ausführen und Blob herunterladen (AjaxReportAndDownloadAction / ab KP 3.8)	213
12.5.6 Blob herunterladen (DownloadBlobAction / ab KP 3.8)	214
12.5.7 KPath-Ausdruck ausführen (ab 3.8.5)	215
12.5.8 KPath-Ausdruck ausführen und Session offenhalten (ab 3.8.6)	216
12.5.9 Mailto-Links erzeugen - EmailLinkIdActionTransformer (ab KP 3.8.8)	219
12.5.10Objektliste als Excel-Datei herunterladen	220
12.5.11Action Render Modes	220



13 Administration	222
13.1 Rechtekonfiguration	222
13.2 Portalstatus	222
14 Kochbuch	222
14.1 Zeitpunkt des letzten Logins in einem Attribut merken	222
15 Knowledge-Portal 4.0 - Versionshinweise	224
15.1 Was ist neu?	224
15.1.1 Box-Konfiguration	224
15.1.2 LESS statt CSS	226
15.1.3 Performance-Verbesserungen	228
15.2 Referenz	228
15.2.1 Bereitgestellte LESS-Variablen	228
16 Knowledge-Portal 5.0 - Versionshinweise	230
16.1 Was ist neu?	230
16.1.1 Änderungen beim BLOB-Streaming	230
17 Tomcat-Konfiguration	231
17.1 Anforderungen	232
17.1.1 Version	232
17.2 Encoding	232
17.3 Tomcat Connector für IIS	233
17.3.1 Tomcat Konfiguration	233
17.3.2 IIS-Website erstellen	233
17.3.3 Tomcat-Connector Konfigurieren	234
17.3.4 Troubleshooting	240



1 Die wichtigsten Funktionen im Überblick

1.1 Information auf einen Blick

Das Knowledge-Portal bietet Information integriert und um die Themen des Geschäfts herum gruppiert - z.B. alle Information zu einem Projekt auf einen Blick. Inhaltlich ähnliche oder thematisch relevante Information befindet sich immer auch in der unmittelbaren Umgebung und kann über wenige Mausklicks erreicht werden.

The screenshot shows the KnowledgePortal interface. The main content area displays the profile of Kerstin Kühling, a Dr. rer. nat. with a photo and various details. The left sidebar contains search and navigation options. The right sidebar lists related information such as travel experience, colleagues, and competencies.

Mitarbeiter	
Kerstin Kühling	
Abbildung	
Akademischer Titel	Dr. rer. nat.
e-mail-Adresse (German)	kerstin.kuehling@universal.com
Geburtsdatum	19.04.1973
Telefon	069 - 777 392
hat Expertise in	Kühlung
ist beteiligt an	Branntwein-Destillation Almeria PEP-Anlage Lima
ist verantwortlich für	Kühlkreislauf (Jug Rusi) WP-Schmelzereislauf
Person arbeitet bei Organisation	Konsum+Convenience

Suche
kühlung
Semantische Suche

Schnellsuche

Navigation

- Universal Ag
 - Chemie+Pharma
 - Energie+Kraftwerke
 - Konsum+Convenience
 - Abteilung Fördertechnik und Transport
 - Abteilung Produktionsplanung und digitale Anlagenbau
 - Anlagenplanung
 - Process-Engineering
 - Wasser und Abwasser
- Projekt
- Thema

Verlauf

- Kerstin Kühling
- Sterling Isollined 206
- Destillation
- Willkommen im Knowledge Portal der Universal AG

intelligent views

- Auslandserfahrung(1)
 - Spanien
- Kollegen in der Abteilung(3)
 - Anton Anlage
 - Franz Förder
 - Werner Wohlgeruch
- Kompetenzbereiche(4)
 - Branntwein
 - Destillation
 - Maximator-A45
 - Polyesterschmelze
- Ähnliche Themen(18)

1.2 Visualisierung mit dem Net-Navigator

Die Vernetzung wird mit dem Net-Navigator visualisiert. Nutzer können sich in einer grafisch-interaktiven Sicht die Zusammenhänge Schritt für Schritt klarmachen und müssen erst für die Details in die einzelnen Dokumente „abtauchen“.



1.3 Intelligente Suchen

Einfache Suchen:

z.B. alle Kollegen, die von einer Änderung an einem bestimmten Produkt betroffen sein können oder alle Zulieferer dafür - sind vorformuliert und stehen als Informationsangebote direkt bei den Objekten zur Verfügung. So kann ein Teil der Fragen direkt aus dem Wissensnetz heraus beantwortet werden, das gilt vor allem für Fragen nach Experten und organisatorischen Zuständigkeiten.

Semantische Suche:

Das Knowledge-Portal enthält eine Suchmöglichkeit, die sich in der Bedienung so einfach präsentiert wie z.B. die Suche in Google, die aber eine umfassende semantische Unterstützung bietet. Wenn beispielsweise in Projekten gesucht wird, werden automatisch Unterprojekte mit einbezogen.

Auf die Frage nach „Milch“ und "Pumpe" wird ein passendes Produkt zurückgeliefert, auch wenn die Suchbegriffe nicht direkt in der Produktbeschreibung vorkommen.



Organisation	
👤 Jug Rusi	■■■■
Produkt	
🔧 Sterling Isolined 206	■■■■
🔧 TKP-A45	■■■■
🔧 KBG-Pumpe	■■■■
🔧 Linn MXH	■■■■
🔧 Maximator-A45	■■■■
🔧 Strahl-Vakuumpumpe	■■■■
🔧 Dampfstrahlpumpe	■■■■
🔧 Exzenterpumpe	■■■■
🔧 Pumpe	■■■■
🔧 Thermpumpe	■■■■
Lebensmittel	
🥛 Milch	■■■■
Flüssigkeit	
💧 Schmelze	■■■■
Meinten Sie...	
Produkt	
🔧 Humpen	■■■■

1.4 Gruppen- und rollenbasierte Personalisierung

Alle Nutzer sind auch als Knoten im Wissensnetz eingetragen und stehen mit den Themen des Netzes in Beziehung. So können sehr bequem Regeln zu Informationsverteilung und -zugriff formuliert werden: „neue Information zu einem Kunden sollten alle Mitarbeiter aller Projekte, die für diesen Kunden durchgeführt werden, automatisch vorgelegt bekommen“

1.5 Rechteverwaltung

1.6 DMS-Funktionalität

Das Knowledge-Portal verfügt über grundlegende DMS-Funktionen wie das Einstellen und Versionieren von Dokumenten und die rollenbasierte Zuweisung von Berechtigungen.

In Projekten wurde zudem das Erstellen, Strukturieren, verteilte Bearbeiten und die Generierung von PDF-Versionen von Dokumenten umgesetzt.

1.7 WYSIWYG-Editor

Durch die flexiblen Konfigurationsmöglichkeiten bei der Pflege des Wissensnetzes über das Knowledge-Portal kann das Einstellen und Verwalten der Informationen individuell angepasst werden.

So können vollständige Artikel mit Hilfe eines WYSIWYG-Editors erstellt und verschlagwortet und parallel alle dazugehörenden strukturierten Informationen mit Erfassungsmasken eingegeben werden.

1.8 Feedback- und Kommentarfunktion

Feedback- und Kommentarfunktionen sowie Diskussionsthreads für alle Informationen im Knowledge-Portal geben allen Benutzern die Möglichkeit, aktiv an der Bewertung und Verbesserung



der Informationen teilzunehmen.

1.9 Abonnementfunktion

Alle Informationen können von den Benutzern individuell getaggt werden. Durch eine Abonnementfunktion kann jeder Nutzer seinen individuellen Zugriff auf die Informationen gestalten.

1.10 Volltextsuche mit statistischer Dokumentklassifikation

Das Knowledge-Portal kann standardmäßig mit der klassifizierenden Volltextsuchmaschine nextbot unseres Partner der *brainbot technologies AG* oder der Open Source Volltextengine Lucene betrieben werden.

Für andere Suchmaschinen wie OmniFind von IBM oder FAST von Microsoft existieren generische Adapter, die eine schnelle Integration ermöglichen.

2 Systemvoraussetzungen

Tomcat-Version

Knowledge-Portal-Version	Benötigte Tomcat-Version
2.x.x	5.5.25 oder höher
3.x.x	6.0.x

Tipp: Wird eine ältere Version verwendet, kann es beim Anzeigen von Detailseiten zu JSP-Fehlern kommen. Ist dies der Fall, sollte die Version des Tomcat überprüft werden. In manchen Fällen hilft es auch, den Inhalt des Work-Verzeichnisses von Tomcat zu löschen (das Verzeichnis "*work*" unterhalb des Tomcat-/Layoutengine-Installationsverzeichnisses).

JAVA-Version

Knowledge-Portal-Version	Java-Version
2.x.x	Java JDK 1.5 (Java 5)
3.x.x	Java JDK 1.6 (Java 6)

K-Infinity-Version

Knowledge-Portal-Version	K-Infinity-Bridge
--------------------------	-------------------



2.x.x	K-Infinity 2.5 (KEM-Version \geq 1.20)
3.0.x	K-Infinity 3.1
3.7.x	K-Infinity 3.2 (KEM-Version \geq 3.11)

WICHTIG: Genaue Versionsangaben zur benötigten K-Infinity-Bridge sind in den Release-Notes der von Ihnen eingesetzten Knowledge-Portal-Version enthalten. Die obige Tabelle dient nur als Anhaltspunkt.

3 Überblick über die KP-Konfiguration

3.1 Basis-Konfiguration

Über die Datei `WEB-INF/kem.properties` in der WAR-Datei wird konfiguriert, die nötig sind, um die weitere Konfiguration aus dem Wissensnetz zu laden. Die Datei ist eine Text-Datei im ISO8859-1 Format, wobei wobei UTF-8 Zeichen als `\uxxxx` dargestellt werden müssen.

Folgende Konfigurations-Optionen können angepasst werden (Default-Werte sind hier aufgeführt):

```
## Bridge Locations  
bridges=//localhost:4713/mixed,//localhost:4714/streaming  
nnUrl=http://localhost:3030/nn
```

Mit den Optionen "bridges" wird festgelegt, welche Bridges das Knowledge-Portal verwenden soll um sich zum Wissensnetz zu verbinden. Hier können mehrere URLs der folgenden Art angegeben werden:

1. `//hostname:port/read` konfiguriert eine Bridge die nur zum Lesen verwendet wird
2. `//hostname:port/edit` konfiguriert eine Bridge die nur zum Schreiben verwendet wird
3. `//hostname:port/mixed` konfiguriert eine Bridge die zum Lesen und zum Schreiben verwendet wird
4. `//hostname:port/streaming` konfiguriert eine Bridge, die für den Upload/Download von Blobs verwendet wird.

Alternativ kann eine einzelne URL der Form `//hostname:port/kloadbalancer` angegeben werden. Dann verwendet das Knowledge-Portal einen KLoadBalancer der auf dieser Adresse erreichbar ist. Dieser muss so konfiguriert sein, dass es Gruppen **KEM** und **STREAMING** gibt. Bridges aus diesen Gruppen werden dann vom Knowledge-Portal zum Lesen und Schreiben (KEM), und für Blobs (STREAMING) verwendet.

Die Option "nnUrl" enthält eine einzelne URL, unter der eine KMultiBridge erreichbar sein muss. Diese wird vom NetNavigator verwendet.



```
# Reconnect zur Bridge alle zehn Minnute  
watchDogLatency = 10000
```

Ist eine Bridge nicht mehr erreichbar, versucht das Knowledge-Portal in festen Abständen, diese Bridge wieder zu erreichen. Die Option "watchDogLatency" enthält den Zeitabstand in Millisekunden zwischen zwei Versuchen. Wird ein KLoadBalancer verwendet, wird diese Option nur noch verwendet um lokale Caches zu aktualisieren. Der KLoadBalancer selbst kümmert sich schon darum, dass Bridges korrekt gestartet und verwaltet werden.

```
## Volume and system user credentials  
## (trustedLogin, if password is empty. Anonymous login, if login is empty)  
volume=KP38-Testnetz  
systemLogin=system  
systemPassword=
```

Die Option "volume" enthält den Namen des Wissensnetzes, auf dem das Knowledge-Portal laufen soll.

"systemLogin" und "systemPassword" sind die Zugangsdaten, mit denen sich der Systembenutzer am Netz anmeldet. Wenn das Systempasswort nicht im Klartext in der Datei abgelegt werden soll, kann über die Seite "admin.skat" eine verschlüsselte Form generiert werden.

```
# Internal name of the Volume-Configuration  
volumeConfigRoot=kpConfig
```

Die Option "volumeConfigRoot" enthält den internen Namen des Begriffs der die Konfiguration des KnowledgePortals enthält. Diese Option muss nur angepasst werden, wenn mehrere KnowledgePortals auf einem einzigen Wissensnetz laufen sollen.

```
## "checkVolumeComponent=false" disables the check of the Knowledge-Portal  
## volume component on startup.  
checkVolumeComponent=true
```

Die Option "checkVolumeComponent" wird die Versionprüfung der Knowledge-Portal-Komponente aktiviert. Die Komponente wird mit dem Admin-Tool zum Netz hinzugefügt werden. Ist die Option angeschaltet, so startet das Knowledge-Portal nur, wenn die Version der Komponente mit der Version des Knowledge-Portals (z.B. 3.8) übereinstimmt.

```
## Languages  
activeLanguages=en,de  
initialLanguage=
```

Der Parameter "activeLanguages" enthält eine Liste der Sprachen, die im Knowledge-Portal verwendet werden sollen, als ISO639-1 Codes. "initialLanguage" ist die Fallback-Sprache, die verwendet wird, wenn die aktuelle Browsersprache oder die im Wissensnetz hinterlegte bevorzugte Benutzersprache (siehe Parameter "userPreferredLanguageKPath - Verweis rechts) nicht in der Liste der "activeLanguages" enthalten ist.

```
# Seit KP 3.9.7  
restBridges=http://localhost:8060
```



Seit Version 3.9.7 können über den Parameter "restBridges" mehrere Restbridges in einer kommaseparierten Liste konfiguriert werden. Dies ist für bestimmte Features notwendig. In der Bridge-Konfiguration muss "trustedLogin" den Wert "true" haben.

3.2 Anwendungskonfiguration im Netz

Das Knowledge-Portal bietet die Möglichkeit, alle Konfigurationseinstellungen im Wissensnetz zu speichern. Dies vereinfacht die Installation neuer Versionen der Software, da keine Konfigurationsdateien im Installationsverzeichnis mehr angepasst werden müssen. Außerdem erleichtert dies den Transfer einer Knowledge-Portal-Installation auf ein anderes System oder den Austausch von Konfigurationseinstellungen mit der Kundenbetreuung von intelligent views.

Konfigurationseinstellungen im Knowledge-Portal werden immer mittels sog. Konfigurationsschlüssel definiert, die sich aus einem Namensraum und einem Eigenschaftsnamen zusammensetzen. Bsp:

```
<login>  
  <type>1</type>  
</login>
```

Im Beispiel werden ein Namensraum "login" und ein Eigenschaftsname "type" definiert. Dem so definierten Konfigurationsschlüssel "login.type" wird dann der Wert "1" zugewiesen.

Das Knowledge-Portal liest Standardwerte für die Konfiguration beim Start aus XML-Dateien, die im Installationsverzeichnis gespeichert sind ("settings.xml", "components.xml" etc.). Danach wird die Konfiguration aus dem Wissensnetz gelesen und mit der Standardkonfiguration überlagert. Im Wissensnetz können also Konfigurationsschlüssel aus der Standardkonfiguration überschrieben oder zusätzliche Konfigurationsschlüssel definiert werden.

Die Definition von Konfigurationsschlüsseln im Wissensnetz erfolgt mit dem Knowledge-Builder.

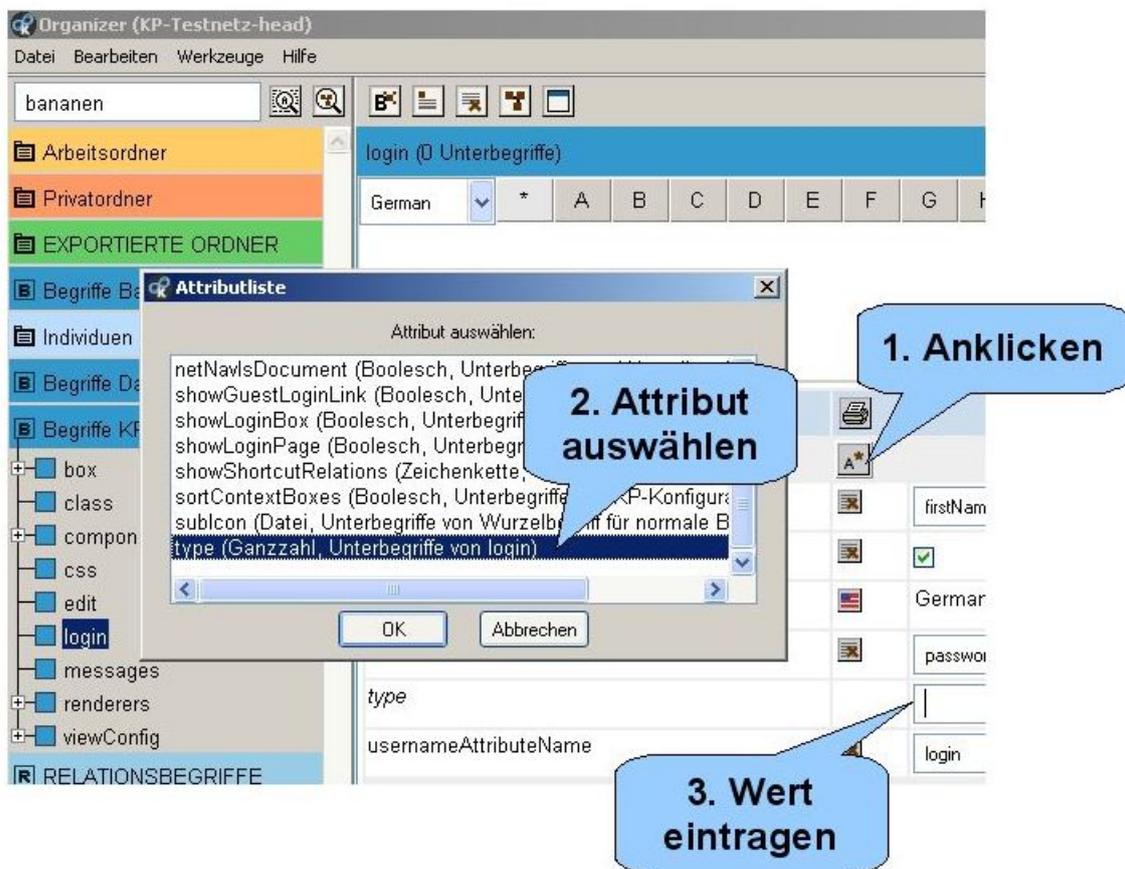
Um Konfigurationsschlüssel definieren zu können, muss zunächst ein Hauptbegriff mit dem Namen "KP-Konfiguration" und dem internen Namen "kpConfig" angelegt werden. Namensräume können definiert werden, indem Unterbegriffe von "KP-Konfiguration" mit den entsprechenden Namen angelegt werden (also z.B. "login", "search" oder "icons"). Anders als beim Hauptbegriff muss für die Unterbegriffe kein interner Name gesetzt werden.

Um eine Eigenschaft in einem Namensraum zu definieren, muss am zugehörigen Begriff zunächst ein neues Attribut angelegt werden, das den Namen der Eigenschaft trägt (also z.B. "type"). Danach kann das neue Attribut am Begriff ausgewählt und der Wert gesetzt werden.



The screenshot shows a software interface for defining a schema for 'login'. The interface includes a header with the title 'login', a text input field for 'Interner Name', and a list of 'Oberbegriffe' (superconcepts) with 'KP-Konfiguration' selected. Below this are buttons for 'Hinzufügen', 'Löschen', and 'Wechseln zu...'. The main area is divided into tabs: 'Begriff', 'Schemadefinition Individuum', and 'Schemadefinition Begriff'. The 'Schemadefinition Begriff' tab is active, showing a list of attributes under the heading 'Attribute'. The attributes listed are: 'additionalFields', 'guestCanCreateUsers', 'guestLoginAllowed', 'passwordAttributeName', 'showGuestLoginLink', 'showLoginBox', 'showLoginPage', 'type', and 'usernameAttributeName'. Each attribute has a checked checkbox and an ellipsis button. The 'type' attribute is highlighted with a red rectangular box. Below the attributes is a section for 'Relationen' (relations) with an 'R*' icon.

Attribut in der Schemadefinition für Unterbegriffe definieren



Attribut anlegen

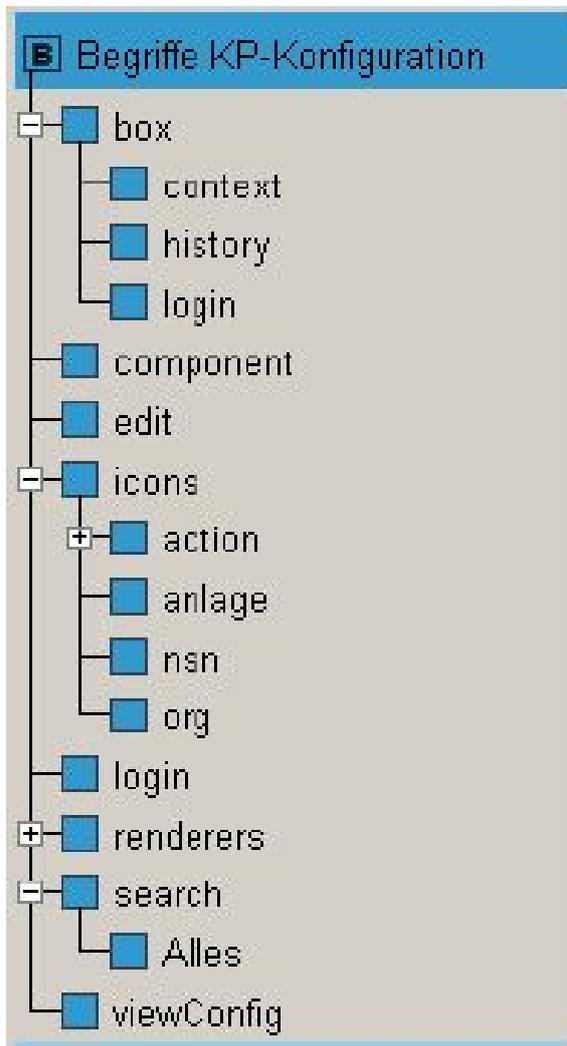
Um im Wissensnetz neu definierte oder geänderte Konfigurationsschlüssel in ein bereits gestartetes Knowledge-Portal zu übernehmen, muss entweder ein Neustart der Web-Anwendung oder ein manueller Reset der Konfiguration durchgeführt werden. Ein manueller Reset kann durch Eingabe der folgenden Adresse im Browser durchgeführt werden:

`http://[serveradresse:port]/[name-der-anwendung]/reloadVolumeConfiguration.skaf?password=[Reset-Kennwort]`

Änderung ab KP 3.8: Das Ausführen dieses Kommandos benötigt das Benutzerrecht "kpAdmin" für den gerade angemeldeten Benutzer. Benutzer, die dieses Recht im Rechtesystem nicht zugewiesen bekommen haben, können die Konfiguration nicht neu laden.

Die in eckigen Klammern angegebenen Platzhalter müssen durch die gültigen Werte für die jeweilige Installation ersetzt werden. Das Reset-Kennwort hat den Standardwert "volumeConfig" (definiert in der Datei "settings.xml" im Installationsverzeichnis) und kann über den Konfigurationsschlüssel "kpVolumeConfig.reset.password" geändert werden.

Einen Ausschnitt aus diesem Teilnetz zeigt die folgende Abbildung. Welche Begriffe und Attribute für die einzelnen Funktionen benötigt werden, ist im jeweiligen Abschnitt beschrieben.



Ausschnitt Teilnetz für Knowledge-Portal-Konfiguration

3.3 Basisnetz

3.3.1 Allgemein

Das K-Infinity Knowledge-Portal verwendet ein vorgegebenes "Basisnetz", bei dem Teile des Schemas schon modelliert sind, das aber dennoch erweitert und ergänzt werden kann. Dieser Abschnitt beschreibt diese vorgegebenen Teile des Wissensnetzes. Daher kann er auch verwendet werden, um ein bestehendes Wissensnetz so anzupassen, dass ein K-Infinity Knowledge-Portal darauf betrieben werden kann.

Die Änderungen betreffen unter anderem

- interne Namen von Systemrelationen und bestimmten Begriffen/Attributen
- das Vorhandensein bestimmter Individuen
- Operationen für die Aktivierung der Kommentarfunktionen
- das Teilnetz *kpConfig*, über das das Layout der Funktionsboxen, der Kontextboxen und einzelner Detailseiten konfiguriert werden kann



Zu beachten ist, dass bestimmte Modifikationen des Basisnetzes zwingend erforderlich sind (beispielsweise das Vorhandensein eines System-Users), da das Knowledge-Portal ansonsten nicht korrekt initialisiert werden kann. Bestimmte Funktionen des Knowledge-Portals stehen zwar nicht zur Verfügung, wenn die Änderungen am Basisnetz nicht durchgeführt worden sind (bspw. das Kommentieren von Topics), beeinträchtigen die Initialisierung oder die sonstigen Funktionen des Portals jedoch nicht.

3.3.2 Änderungen im Basisnetz

Die folgenden Elemente müssen zwingend in jedem Wissensnetz vorhanden sein, damit es mit einem Knowledge-Portal-Layout angezeigt werden kann. Weitere optionale Änderungen werden in den entsprechenden Kapiteln beschrieben.

Interne Namen

Topic/Attribut/Relation	interner Name
Konfigurationsbegriff	kpConfig
Systemrelation	systemRelation
Abkürzungsrelation	shortcutRelation
Benutzerrelation	userRelation
Systemrelation "erweitertIndividuum"	extendsInstance
Login-Attribut beim User-Topic	login

System-User

Da sich das Knowledge-Portal für die Initialisierung selbst am Wissensnetz anmeldet, muss für diese Anmeldung eine Instanz des Personenbegriffs vorhanden sein. Beim aktuellen Basisnetz ist dies ein Individuum von "*Web-Nutzer*" mit dem Namen "*system*" und dem Wert "*system*" für das Login-Attribut. Das Attribut password muss ebenfalls den Wert "*system*" enthalten.

Mehr Informationen zum Systembenutzer und zum Login finden Sie in den Abschnitten *Systembenutzer* und *Login-Konfiguration*.

Login-Suche

KEM-Suche, die zum Login verwendet wird. Mehr Informationen zum Login finden Sie im Abschnitt *Login-Konfiguration*.

KEM-Suche

Es muss mindestens eine KEM-Suche vorhanden sein, die als Suche im Knowledge-Portal verwendet wird.

3.3.3 Benutzer

Personenbegriff

Der Personenbegriff legt fest, welche Personen sich am Knowledge-Portal einloggen können. Nur Personen, die Individuen dieses Begriffs sind, werden beim Login berücksichtigt.

Sollen nicht alle gespeicherten Personen auch die Möglichkeit bekommen sich einzuloggen, wird ein Unterbegriff (z.B. "Webbenutzer") zum Begriff "Person" definiert. Wird der Personenbegriff nun auf "Webbenutzer" festgelegt, ist ein Login nur dieser Teilmenge an Personen gestattet.

Der Personenbegriff wird im Knowledge-Builder definiert. Wählen Sie dazu "Datei -> Einstellungen". Unter dem Reiter "Rechte" kann der gewünschte Personenbegriff gewählt werden.

In diesem Fenster kann auch festgelegt werden, welcher Benutzer verwendet werden soll, wenn ein Gast-Login am System möglich ist (vgl. Abschnitt Login-Konfiguration)

3.3.3.1 Systembenutzer

Zum Auslesen von Konfigurationsinformationen aus dem Wissensnetz benötigt das Knowledge-Portal ein spezielles Benutzerkonto im Wissensnetz, den sog. *System-Benutzer* (nicht zu verwechseln mit dem Gast-Benutzer).

Da sich das Knowledge-Portal für die Initialisierung selbst am Wissensnetz anmeldet, muss für diese Anmeldung eine Instanz des Personenbegriffs vorhanden sein. Beim aktuellen Basisnetz ist dies ein Individuum von "Web-Nutzer" mit dem Namen "**system**" und dem Wert "system" für das **Login-Attribut**.

Das Attribut **password** muss ebenfalls den Wert "system" enthalten.

Wenn im Wissensnetz das Rechtesystem aktiviert ist, so muss darauf geachtet werden, dass für den System-Benutzer keinerlei Rechteeinschränkungen gelten. Am einfachsten ist dies zu realisieren, indem man als erste Regel im Rechtesystem einen Benutzerfilter für den System-



Benutzer einfügt, der alle Operationen erlaubt (vgl. K-Infinity-Handbuch/Rechtesystem).

Mehr Informationen zum Systembenutzer und zum Login finden Sie in den Abschnitten *Systembenutzer* und *Login-Konfiguration*.

Der System-Benutzer kann nicht zum manuellen Login über die Login-Maske verwendet werden.

Benutzername und Passwort

Der Benutzername des Systembenutzers kann in der Konfigurationsdatei "*customized.xml*" festgelegt werden. der Defaultname ist "*system*". Zum Ändern des Benutzernamens wird folgender Schlüssel verwendet:

```
<kpConfig><systemUser>  
  <login>Benutzername</login>  
</systemUser></kpConfig>
```

Das Passwort des Benutzers wird unter diesem Schlüssel gespeichert:

```
<kpConfig><systemUser>  
  <password>SysUserPassword</password>  
</systemUser></kpConfig>
```

3.4 KP-Wissensnetzkomponente (ab Version 3.7)

Ab Knowledge-Portal-Version 3.7 und K-Infinity 3.2.3 wird das Knowledge-Portal-Konfigurationsmodell im Wissensnetz von einer Wissensnetzkomponente verwaltet, die mit dem Admin-Tool installiert werden muss. Diese Komponente kümmert sich beim Umstieg von einer älteren auf eine neuere Knowledge-Portal-Version um den Upgrade des Konfigurationsmodells im Wissensnetz und um die Erzeugung des Basisschemas für den Betrieb eines Knowledge-Portals (Vergabe von internen Namen, Erzeugung von Attributen, etc.). So wird z.B. die Konfiguration für MicroEdit-Boxen bei der erstmaligen Installation der Komponente aus dem alten KP-3.6-Modell in das neue KP 3.7-Modell (Box-Individuen im Konfigurationsnetz) übersetzt.

Das Knowledge-Portal ab Version 3.7 meldet einen Fehler beim Starten, wenn im Wissensnetz nicht die passende Knowledge-Portal-Komponente installiert ist. Dadurch soll sichergestellt werden, dass das Wissensnetz ein zum jeweiligen Knowledge-Portal passendes Konfigurationsmodell enthält.

Sollte es aus wichtigen Gründen notwendig sein, die Prüfung auf das Vorhandensein der Komponente beim Start im Knowledge-Portal zu deaktivieren (z.B. weil man eine alte und eine neue Version des Knowledge-Portals gleichzeitig auf dem selben Wissensnetz einsetzen möchte), so kann die Prüfung deaktiviert werden, indem man im KP-War-File in der Datei "kem.properties" die Eigenschaft "checkVolumeComponent" auf den Wert "true" setzt.

Installation der Wissensnetzkomponente

1) Admin-Tool starten und anmelden am Wissensnetz. Wichtig: Stellen Sie zuvor sicher, dass Sie die zur jeweiligen Knowledge-Portal-Version passende Version des Admin-Tools verwenden (bspw. für das KP 3.7.0 mit der Build-Nummer \geq 11020900).

2) Wählen Sie im Admin-Tool im Navigationsbaum links den Punkt "Systemkonfiguration > Komponenten" aus.



3) Auf der rechten Seite werden jetzt zwei Listen angezeigt: Oben die Liste der verfügbaren Komponenten (mit "Software" überschrieben) und unten die Liste der im Wissensnetz installierten Komponenten (mit "Wissensnetz" überschrieben). Sollte noch keine Komponente mit dem Namen "Knowledge-Portal" im Wissensnetz installiert sein, wählen Sie die Knowledge-Portal-Komponente in der Software-Liste aus. Vergewissern Sie sich, dass die Hauptversionsnummer der Komponente der Hauptversion des von Ihnen eingesetzten Knowledge-Portals entspricht (d.h. bei einem Knowledge-Portal 3.7.x muss auch die Komponente die Version 3.7.x haben).

4) Klicken Sie jetzt auf "Standardkomponente hinzufügen", um die Komponente im Wissensnetz zu installieren. Nach der Installation zeigt das Admin-Tool sofort an, dass eine Aktualisierung der Komponente notwendig ist, damit alte Konfigurationsmodelle auf jeden Fall in das aktuellste Format migriert werden. Dies wird im folgenden Abschnitt beschrieben.

Aktualisieren der Wissensnetzkomponente

Das Konfigurationsmodell im Wissensnetz kann an die neueste Knowledge-Portal-Version angepasst werden, indem die Knowledge-Portal-Komponente im Wissensnetz aktualisiert wird. Zur Aktualisierung der Komponente führen Sie bitte die Schritte 1-3 aus der obestehenden Installationsanleitung aus. Sollte eine Aktualisierung des Konfigurationsmodells notwendig sein, wird dies rot in der Liste "Wissensnetz" hervorgehoben. Wählen Sie zum Upgrade die rot markierte Knowledge-Portal-Komponente aus und klicken Sie den Button "Aktualisieren".

3.5 XML-Komponentenkonfiguration (ab KP 3.7)

Die Konfiguration der KP-Komponenten (z.B. Login-Komponenten, Event-Log-Komponenten) muss ab der KP-Version 3.7 in XML formuliert werden. Das XML muss in einem Zeichenketten-Attribut mit dem Namen "componentsXml" am KP-Konfigurationsbegriff hinterlegt werden. Eine leere XML-Komponentenkonfiguration sieht folgendermaßen aus:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>  
<Components xmlns="http://www.iviews.com/knowledgeportal/kpConfig" >  
</Components>
```

Das XML muss dem kpConfig-XML-Schema der jeweiligen KP-Version genügen. Bitte verwenden Sie zum Bearbeiten des XML die KP-XML-Entwicklungsumgebung für die jeweilige KP-Version (siehe Kapitel "KP-XML-Entwicklungsumgebung").

Zur Konvertierung einer bestehenden Komponentenkonfiguration auf Begriffsebene (Begriffe "applicationComponents", "component" und "searchComponents") nach XML steht eine Upgrade-Funktion auf der KP-Admin-Seite (/admin.skat) zur Verfügung. Diese wandelt die Konfigurationsbegriffe in ein XML um und speichert dieses im Attribut "componentsXml" ab. Nach der Konvertierung werden die Namen der alten Konfigurationsbegriffe mit dem Zusatz "(obsolete)" versehen.



3.6 XML-Entwicklungsumgebung

Innerhalb der KP-Konfiguration müssen bestimmte Teilkonfigurationen in XML formuliert werden. Hierzu zählen:

- Konfiguration von Detail- und Pflegeseiten (View-/EditConfig)
- Konfiguration von Actions (Action-Config-XML)
- Konfiguration der Suchadapter (SearchConfig-XML)
- Konfiguration der Suchergebnisseiten (ResultViewConfig)
- Konfiguration der KP-Komponenten (Components-XML)

Jedes XML-Format ist entweder in einer DTD oder einem XML-Schema beschrieben. Um das XML für die jeweiligen Teilkonfigurationen komfortabel eingeben zu können und das XML auf Konformität mit dem jeweiligen Schema bzw. der jeweiligen DTD überprüfen zu können, gibt es seit dem KP 3.7 eine Entwicklungsumgebung zur Eingabe des XML. Die Entwicklungsumgebung basiert auf Eclipse und ist mit allen XML-Schemata bzw. DTDs der jeweiligen KP-Version vorkonfiguriert. Außerdem sind in der Entwicklungsumgebung Beispiel-XML-Dateien für die jeweiligen XML-Formate enthalten.

Bitte gehen Sie zur Installation der Entwicklungsumgebung bitte folgendermaßen vor:

1. Download des ZIP-Archivs mit der Entwicklungsumgebung (Klick auf Büroklammer-Symbol in der untenstehenden Tabelle - funktioniert nur in der Online-Version dieses Dokuments in ODIN)
2. **Optional:** Falls noch keine Java-Laufzeitumgebung der Version 1.5 oder höher auf Ihrem Computer installiert ist, laden Sie bitte den Java-Installer herunter und führen Sie diesen aus.
3. Auspacken der ZIP-Datei in ein beliebiges Verzeichnis
4. Starten der Entwicklungsumgebung durch Ausführen der Datei "Eclipse.exe" im Installationsverzeichnis.

KP-Version	Betriebssystem	Entwicklungsumgebung (ZIP-Archiv)	Java-Installer
3.7	Windows (32 bit)	 KPConfigXMLEditor-3.7-win32.zip	 jre-6u35-windows-i586.exe
3.8	Windows (32 bit)	 KPConfigXMLEditor-3.8.1-win32.zip	 jre-6u35-windows-i586.exe



4 Login-Konfiguration

4.1 Grundkonfiguration Login

Login-Suche

Die Login-Suche dient zur Ermittlung der Personeninstanz des sich einloggenden Benutzers. Sie muss den Namen "*login*" tragen. Wird K-Infinity mit einem Knowledge-Portal verwendet, muss diese Suche auch bei den KEM-Suchen der Gruppe "*verwendete Konfigurationen*" zugeordnet werden.

In der Suche kann in verschiedenen Attributen gesucht werden, was ermöglicht, sich z.B. mit dem Login-Namen oder der E-Mailadresse anzumelden. Wichtig dabei ist, dass bei korrekter Angabe des Benutzernamens nur **ein** Individuum gefunden wird. Ist das Ergebnis der Suche **nicht eindeutig**, schlägt das Login fehl!

Die Konfiguration der Login-Suche wird im Knowledge-Builder vorgenommen. Wählen Sie dazu "*Datei > Einstellungen*". Unter dem Reiter "*KEM Suche*" werden links alle existierenden Suchen aufgelistet. Wählen Sie die Suche "*login*" und klicken Sie "*Bearbeiten*". Weitere Informationen zum Bearbeiten vom Suchen finden Sie im Handbuch des Knowledge-Builders.

Hinweis: Bei der Login-Suche sollte die Option "Suchbegriff zerlegen" **ausgeschaltet** sein. Andernfalls besteht die Gefahr, dass der Login-Name nicht gefunden wird, wenn er Leer- oder Satzzeichen enthält.

Speicherung des Passworts

Bei der Wahl des Personenbegriffs wird an diesem Begriff automatisch das Attribut "*password*" erzeugt, das das Passwort des jeweiligen Benutzers beinhaltet. Dieses wird verschlüsselt abgelegt und ist nicht einsehbar.

Anmerkung: Beim Wechseln des Personenbegriffs wird das alte Passwortattribut nicht gelöscht, da sonst die gespeicherten Daten ebenfalls gelöscht werden. Das Attribut muss unter Umständen manuell nachgezogen werden.

Wahl des Login-Namens

Der Login-Name des Benutzers kann in einem eigenen Attribut gespeichert werden und somit vom Namen des Individuums abweichen. In der Grundkonfiguration ist das Namensattribut der Person als Benutzername voreingestellt, kann jedoch problemlos geändert werden.

Zum Ändern des Login-Namens muss ein zusätzliches Attribut am Personenbegriff definiert werden. Alternativ kann auch ein bereits bestehendes Attribut gewählt werden, wenn z.B. die E-Mailadresse als Login-Name verwendet werden soll.

Bitte beachten: Diese Änderung muss auf jeden Fall in der Login-Suche und der Konfiguration des Systembenutzers berücksichtigt werden!

4.2 Konfigurierbare Parameter

Die folgende Liste zeigt alle Parameter, die zur Konfiguration der Login-Seite möglich sind. Sie sind frei wählbar, sollten jedoch mit Bedacht gewählt werden, um eventuelle Sicherheitslücken durch eine Fehlkonfiguration zu vermeiden.

Wird zusätzlich der Parameter "*type*" gewählt, überschreibt dieser evtl. manuell gesetzte Parameter. Welche Parameter vom Parameter "*type*" erzwungen werden, wird bei der entsprechenden Login-Art beschrieben.



Parameter	Typ	Default-wert	Beschreibung
type	Integer		Grundkonfiguration für verschiedene Login-Arten (überschreibt andere Parameter!).
userConcept	String		Interner Name des Benutzerbegriffs.
usernameAttribute	String	login	Interner Name des Benutzernamen-Attributs.
passwordAttribute	String	password	Interner Name des Passwort-Attributs.
guestLoginAllowed	Boolean	true	Gibt an, ob ein Login ohne Authentifizierung möglich ist.
showGuestLoginLink	Boolean	true	Gibt an, ob auf der Login-Seite ein Link angezeigt werden soll, um sich als Gastbenutzer einzuloggen (nur in Verbindung mit <i>showLoginPage=true</i> relevant).
showLoginPage	Boolean	false	Falls <i>true</i> , wird eine Login-Seite vor die Webapplikation geschaltet.
showLoginBox	Boolean	true	Zeigt eine Kontextbox mit Login-Formular bzw. Benutzerinformationen an.
guestCanCreateUsers	Boolean	false	Gestattet es einem Gastbenutzer, einen neuen Benutzer anzulegen. Dies wird benötigt, wenn man eine Registrierung an der Webapplikation ermöglichen möchte.
additionalFields	String		Interne Namen der Attribute, die bei der Registrierung zusätzlich abgefragt werden sollen; die Liste wird durch Leerzeichen getrennt.



afterUserLoggedIn	String		Die Zeichenkette, die hier gesetzt wird, wird bei jedem Login als K-Script ausgeführt. Damit lassen sich z.B. Attribute am Benutzer setzen, die den Zeitpunkt des letzten Logins speichern.
userPreferredLanguageKPath (ab KP 3.8.6)	String		<p>KPath, mit dem die bevorzugte Sprache für das Web-Frontend aus dem Wissensnetz ausgelesen werden kann. Der KPath wird beim Login mit dem Benutzer-Topic als Startobjekt ausgeführt wird. Der KPath-Ausdruck muss einen String zurückliefern, der in ein Java-Locale-Objekt umgewandelt werden kann. Details zu der Zeichenketten-Repräsentation der Java-Locales sind in der Javadoc der Java-Klasse "Locale" zu finden (siehe WWW-Link rechts). Das erzeugte Locale wird mit den im KP aktivierten Sprachen (activeLanguages in kem.properties) abgeglichen und nur dann als initiale Sprache gesetzt, wenn es mit dieser Liste vereinbar ist.</p> <p>Beispiel 1: Ist als bevorzugte Sprache im Wissensnetz "cn" gesetzt, in der kem.properties aber nur die Sprachen "de" und "en" aktiviert, so wird "cn" nicht als initiale Sprache gesetzt und es greifen die Einstellungen aus dem Web-Browser oder die in der kem.properties konfigurierten Standardsprache (initialLanguage).</p> <p>Beispiel 2: Ist als bevorzugte Sprache am Benutzer "de_AT" gesetzt und im KP die Sprache "de" aktiviert, so wird "de_AT" als initiale Sprache gesetzt.</p>



4.3 Login-Art

Das Knowledge-Portal ermöglicht unterschiedliche Vorgehensweisen beim Login. Grundle-
gend wird dabei unterschieden, ob es ein Login gibt, ein Gast-Login möglich ist oder sich nur
registrierte Benutzer einloggen können.

Eine Grundkonfiguration kann über den Parameter "*type*" erreicht werden. Dieser erwartet
einen Integer-Wert, der für eine der folgenden Login-Konfigurationen steht. Eine detaillierte
Konfiguration kann über einzelne Parameter erreicht werden.

Hinweis: Manuell gesetzte Parameter, die nicht mit dem Parameter "*type*" vereinbar sind,
werden überschrieben! Die Beschreibungen der einzelnen Login-Konfigurationen beinhalten
zusätzlich eine Liste der Parameter, die mit dem Parameter "*type*" überschrieben werden.

4.3.1 Kein Login (type = 0)

Diese Login-Art lässt kein Login zu, d.h. alle Benutzer arbeiten mit den Rechten eines Gast-
benutzers.

Die Einstellung setzt voraus, dass ein Gastbenutzer definiert ist. Näheres hierzu finden Sie im
Kapitel *Systembenutzer*. Die Wahl dieser Login-Art erzwingt außerdem folgende Attributwerte
beim "*login*"-Begriff:

login	
Attribute	A *
guestLoginAllowed	<input checked="" type="checkbox"/>
Name (Deutsch)	German <input type="checkbox"/> univ. login
showLoginBox	<input type="checkbox"/>
showLoginPage	<input type="checkbox"/>
type	0

4.3.2 Login-Box (type = 1)

Ermöglicht das Login in einem Kontextmenü. Alle Benutzer sind beim Aufrufen der Webapp-
likation als Gastbenutzer eingeloggt. Über eine Login-Box in der linken oberen Ecke können
sie sich dann mit ihren Benutzerdaten einloggen.

Die Wahl dieser Login-Art setzt voraus, dass ein Gastbenutzer definiert ist. Näheres hierzu
finden Sie im Kapitel *Systembenutzer*. Die Wahl dieser Login-Art erzwingt zusätzlich folgende
Attribute beim "*login*"-Begriff des kpConfig-Teilnetzes:



login	
Attribute	A*
guestLoginAllowed	<input checked="" type="checkbox"/>
Name (Deutsch)	German <input type="checkbox"/> univ. login
showLoginBox	<input checked="" type="checkbox"/>
showLoginPage	<input type="checkbox"/>
type	1

4.3.3 Login-Seite ohne Gastbenutzer (type = 2)

Bei dieser Login-Art wird der Webapplikation eine Login-Seite vorangestellt; das Einloggen als Gastbenutzer ist nicht möglich. Erst nach erfolgreicher Authentifizierung mit Benutzername und Passwort kann auf das Frontend zugegriffen werden. Diese Login-Art erzwingt zusätzlich folgende Attributwerte beim "login"-Begriff:

login	
Attribute	A*
guestLoginAllowed	<input type="checkbox"/>
Name (Deutsch)	German <input type="checkbox"/> univ. login
showLoginBox	<input checked="" type="checkbox"/>
showLoginPage	<input checked="" type="checkbox"/>
type	2

Parametereinstellung für Gastbenutzer Typ 2

4.3.4 Login-Seite mit Gastbenutzer (type = 3)

Diese Einstellung verhält sich wie die Login-Art "Login-Seite ohne Gastbenutzer", jedoch ist ein Gast-Login möglich. Dazu wird auf der Login-Seite zusätzlich zum Login-Formular ein Link eingeblendet, über den sich jeder als Gast einloggen kann.



Hier ist Voraussetzung, dass ein Gastbenutzer konfiguriert ist. Dazu ist die folgende Parametereinstellung zwingend notwendig:

login	
Attribute	A*
guestLoginAllowed	<input checked="" type="checkbox"/>
Name (Deutsch)	<input checked="" type="checkbox"/> German <input type="checkbox"/> univ. <input type="text" value="login"/>
showLoginBox	<input checked="" type="checkbox"/>
showLoginPage	<input checked="" type="checkbox"/>
type	<input type="text" value="3"/>

Parametereinstellung für Gastbenutzer Typ 3

4.4 Startseite

Das Knowledge-Portal bietet verschiedene Möglichkeiten die Startseite eines Benutzers festzulegen. Diese Seite wird unmittelbar nach dem Login aufgerufen und dient als Einstieg für den Benutzer.

Benutzerseite als Startseite

Per Default wird der Benutzer nach dem Login auf sein eigenes Benutzertopic weitergeleitet. Loggt sich der Benutzer als Gast ein, landet er entsprechend auf dem Topic des Gastbenutzers.

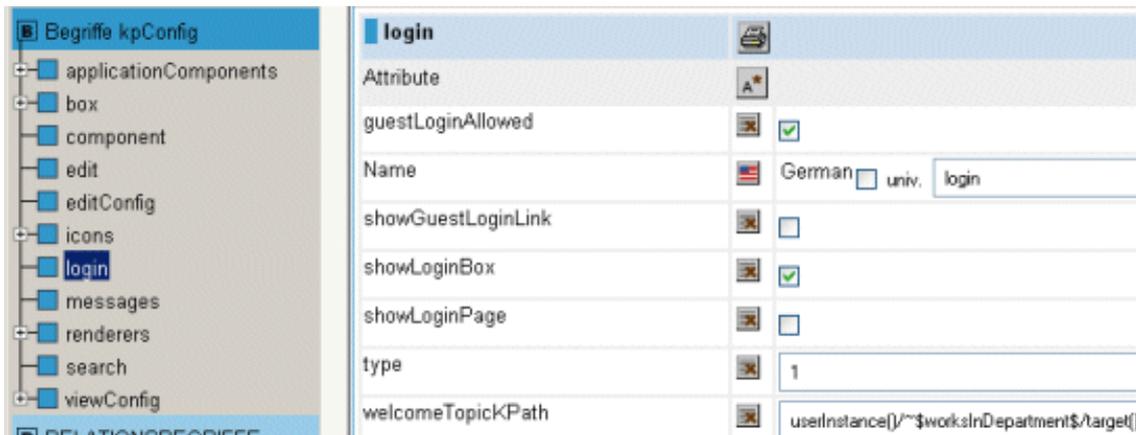
"Willkommen"-Seite

Die Konfiguration lässt es zu, ein bestimmtes Topic als Startobjekt zu definieren. Allen Benutzern wird diese Seite als Startseite angezeigt. Dazu muss dem gewählten Topic der interne Name "welcome" gegeben werden. Verwenden Sie dazu den Knowledge-Builder.

"Willkommen"-Seite über KPath

Das Knowledge-Portal ermöglicht die Angabe eines KPath zum Ermitteln der Startseite. So kann auf beliebige Topics im Wissensnetz verwiesen werden.

Dieser KPath wird unter folgendem Schlüssel definiert:



Soll die Startseite benutzerbezogen sein, kann im KPath die Funktion "*userInstance()*" verwendet werden. Sie repräsentiert das Benutzer-Individuum, von dem aus beliebige Relationen abgelaufen werden können.

Der KPath zur Abteilung des eingeloggten Benutzers würde wie folgt aussehen:

```
userInstance()/~$worksInDepartment$/target()
```

4.5 Login-Komponenten

Im Knowledge-Portal können unterschiedliche Login-Komponenten konfiguriert werden, um unterschiedliche Wege für die Anmeldung festzulegen. Wird nichts anderes konfiguriert wird die *StandardLoginComponent* verwendet.

Konfiguration der Login-Komponente bis KP 3.6

Um die Konfiguration zu ändern ist der Konfigurationsbegriff

```
kpConfig > applicationComponents > login
```

zu definieren. Dieser besitzt das Attribut *class*, das spezifiziert, welche Login-Komponente verwendet wird. Je nach Login-Komponente können bzw. müssen zusätzliche Attribute definiert werden. Welchen Wert *class* besitzen muss und welche zusätzlich Attribute verwendet werden, entnehmen Sie bitte den folgenden Kapiteln.

Konfiguration der Login-Komponente ab KP 3.7

Ab der Knowledge-Portal-Version 3.7 erfolgt die Komponentenkonfiguration über XML. Details zur Komponentenkonfiguration sind im Kapitel "XML-Komponentenkonfiguration" (siehe Verknüpfung rechts) beschrieben.

4.5.1 StandardLoginComponent

Diese Login-Komponente wird per default verwendet, ist also aktiv wenn keine Netzkonfiguration existiert. In diesem Fall wird folgendes Attribut gesetzt:

```
class = com.iviews.knowledgeportal.login.StandardLoginComponent
```



Bei dieser Komponente wird die Benutzerauthentifizierung gegen die im Netz gespeicherten *Web-User* mit dem im Netz gespeicherten Passwort durchgeführt.

Die Definition weiterer Attribute ist nicht nötig.

4.5.2 NtlmLoginComponent

Die NtlmLoginComponent ermöglicht die Authentifizierung der Benutzer gegen einen ActiveDirectoryServer. Bei der Verwendung des InternetExplorers authentifiziert sich der Benutzer automatisch mit seinem Windows-Login-Namen und wird durch ein Challenge-Response-Verfahren gegen den DomainController authentifiziert.

Zusätzlich ermöglicht diese Login-Komponente ein SingleSignOn. Das bedeutet, dass der Benutzer beim Aufruf der Website ohne Einwirken eingeloggt wird. Dieser Vorgang geschieht im Dialog zwischen Browser und Webapplikation und ist für den Benutzer nicht sichtbar. Durch das Challenge-Response-Verfahren ist auch keine Übermittlung des Passwortes nötig.

Unterstützte Konfigurationen:

- NTLMv1: JA
- NTLMv2: NEIN, diese Version wird aktuell (KP 3.7.x) nicht unterstützt.

Voraussetzungen

Um diese Login-Komponente zu verwenden ist zwingend erforderlich, dass die Benutzer in einer Domäne verwaltet werden. Die Authentifizierung geschieht dann zwischen Tomcat und Domaincontroller.

Personen die nicht in der Domäne existieren, können sich trotzdem einloggen. Diese werden dann wie in der *StandardLoginComponent* gegen das Netz authentifiziert. Dies erfordert jedoch, dass der Parameter *allowNonSSOUsers* auf *true* gesetzt wird.

Da bei der Verwendung der NTLM-Authentifizierung keine Passwörter im Netz abgelegt werden, müssen Benutzer im Netz, die nicht über den Domaincontroller authentifiziert werden können, zwingend über ein Passwort verfügen. Ansonsten würde man sich ohne Passwort als NTLM-Benutzer authentifizieren können, wenn der DomainController nicht erreichbar ist und das Flag *allowNonSSOUsers* gesetzt ist.

NTLM-Benutzer hingegen dürfen im Netz kein Passwort gesetzt haben!

Der Login ohne Passwort wird generell verweigert!

Voraussetzungen Bridge

Da bei der Nutzung von SSO kein Passwort im Netz gespeichert wird, muss der Benutzer ohne Passwort an der Bridge angemeldet werden können. Daher muss die Bridge entsprechend konfiguriert sein. Die Konfigurationsdatei (*bridge.ini*) muss folgenden Parameter beinhalten:

```
[KEMBridge]
trustedLoginEnabled=true
```

Aktivierung der Komponente bis Knowledge-Portal Version 3.6

Um die NtlmLoginComponent zu aktivieren muss das class-Attribut am Konfigurationsbegriff `kpConfig > applicationComponents > login` auf folgenden Wert gesetzt werden:



com.iviews.knowledgeportal.login.NtlmLoginComponent

Aktivierung der Komponente ab Knowledge-Portal Version 3.7

Im Attribut "componentsXml" am KP-Konfigurationsbegriff muss das Element "<StandardLogin />" durch das folgende XML-Fragment ersetzt werden:

```
<NtlmLogin>
  <allowNonSSOUsers>true/false</allowNonSSOUsers>
  <domainName>Name der Windows-Domäne</domainName>
  <enableSSO>true/false</enableSSO>
  <domainController>AD-Servername</domainController>
  <userName>Benutzername für AD-Authentifizierung (preauthentication)</userName>
  <password>Kennwort für AD-Authentifizierung</password>
</NtlmLogin>
```

Die möglichen Parameter der Komponente werden folgenden Abschnitt beschrieben.

Bitte verwenden Sie zum Bearbeiten der XML-Konfiguration die KPConfig-XML-Entwicklungsumgebung (siehe Link rechts).

Konfiguration der NtlmLoginComponent

Die folgenden Attribute werden unterstützt:

Parameter	Typ	Default	Beschreibung
domainController	String		Name oder IP-Adresse des DomainControllers. Wird dieser Parameter gesetzt, wird nur dieser eine Server verwendet. Bleibt der Parameter leer, wird ein DomainController über den Domain-Namen ermittelt (Load-Balancing)
domainName	String	!REQUIRED!	Name der Domäne
enableSSO	Boolean	true	Gibt an, ob Benutzer beim Aufruf der Applikation automatisch eingeloggt werden sollen.
allowNonSSOUsers	Boolean	false	Gibt an, ob bei fehlgeschlagener Authentifizierung gegen den DomainController eine Authentifizierung gegen das Netz durchgeführt werden soll.



userName	String		Name des Domänen-Accounts der zur Authentifizierung der Verbindung vom Knowledge-Portal zum Domänencontroller verwendet werden soll (ohne Domäne, d.h. "myUser" und nicht "myDomain\myUser"). Dieser Parameter sollte gesetzt werden, wenn die NTLM-Authentifizierung bei Domänencontrollern ab Windows Version 2003 nicht funktioniert. Wenn dieser Parameter gesetzt wird, muss auch immer der Parameter "password" gesetzt werden. Dieser Parameter ist ab KP-Version 2.6.2 verfügbar.
password	String		Das Kennwort zu dem in "userName" spezifizierten Domänen-Accounts Dieser Parameter ist ab KP-Version 2.6.2 verfügbar.

4.5.3 SPNEGO-Login-Komponente (ab KP 3.8)

4.5.3.1 Einführung

Die SPNEGO-Login-Komponente ermöglicht innerhalb von Microsoft-Windows-Netzwerken die automatische Anmeldung eines Benutzers am Knowledge-Portal (d.h. Anmeldung ohne Login-Formular), sofern dieser Benutzer an einer Windows-Domäne angemeldet ist.

Der Unterschied zur NTLM-Login-Komponente liegt darin, dass die SPNEGO-Login-Komponente die Authentifizierung des Benutzers mittels des Kerberos-Protokolls durchführt. Das Kerberos-Protokoll ist seit Windows 2000 das Standardverfahren für die Authentifizierung von Benutzern und Anwendungen innerhalb von Windows-Domänen und hat gegenüber NTLM eine Reihe von Vorteilen (siehe "Externe Ressourcen" rechts).

4.5.3.2 Systemvoraussetzungen

Für den Einsatz der SPNEGO-Login-Komponente gelten die folgenden Systemvoraussetzungen:

Client

- Workstation-Betriebssystem-Version: Windows 2000 Professional SP2 oder höher
- Benutzer ist mit einem Windows-Domänen-Benutzerkonto angemeldet
- Benutzer verwendet als Browser den Microsoft Internet Explorer 6.01 oder höher (empfohlen; es können auch andere Browser verwendet werden. Details hierzu sind im Kapitel "Browserkonfiguration" zu finden).

Server



- Server-Betriebssystem-Version: Windows 2000 Professional SP2 oder höher
- Server-System gehört der gleichen Windows-Domäne an, der auch die Knowledge-Portal-Benutzer angehören.

4.5.3.3 Kerberos-Konfiguration

4.5.3.3.1 Anlegen der Datei "krb5.ini"

Zunächst muss eine Kerberos-Konfigurationsdatei für den Java Authentication and Authorization Service (JAAS) angelegt werden. In dieser Datei werden die Parameter für den Zugriff auf den Kerberos Key Distribution Center (KDC) und für die Abbildung von Windows-Domänen auf Kerberos-Realms definiert. Die Parameter richten sich nach der Kerberos-Konfiguration innerhalb der jeweiligen Windows-Domäne und können daher von Domäne zu Domäne völlig unterschiedlich sein (eine Beispielkonfiguration, die in vielen Fällen funktionieren sollten, ist unten zu finden). Details zu dieser Konfigurationsdatei können der JAAS-Dokumentation und der Kerberos-Referenz für die Datei "krb5.conf" entnommen werden (siehe externe Ressourcen rechts).

Java sucht die Kerberos-Konfigurationsdatei unter Windows standardmäßig im Pfad "%WINDIR%\krb5.ini". Da der Pfad zu dieser Datei nicht in jedem Fall umkonfiguriert werden kann (z.B. beim Aufruf der Java-Kerberos-Dienstprogramme "kinit" und "klist"), sollte man die Datei genau an dieser Stelle anlegen.

Hier eine Standardkonfiguration für die krb5.ini:

```
[libdefaults]
default_realm = MYDOM.COM (Identifies the default realm)
default_tkt_etypes = aes128-cts rc4-hmac des3-cbc-sha1 des-cbc-md5 des-cbc-crc
default_tgs_etypes = aes128-cts rc4-hmac des3-cbc-sha1 des-cbc-md5 des-cbc-crc
permitted_etypes = aes128-cts rc4-hmac des3-cbc-sha1 des-cbc-md5 des-cbc-crc
ticket_lifetime = 600

[realms]
MYDOM.COM = {
kdc = MachineA.mydom.com (host running the KDC/AD-Controller)
default_domain = mydom.com
}

[domain_realm]
.mydom.com = MYDOM.COM

[appdefaults]
autologin = true
forward = true
forwardable = true
encrypt = true
```

Wichtig bei der Kerberos-Konfiguration ist, dass man Domain- und Realm-Name nicht durcheinanderbringt. Der Realm-Name ist normalerweise identisch mit dem Domain-Namen, muss jedoch immer in Großbuchstaben geschrieben werden.

Wenn als KDC ein Server mit dem Betriebssystem Windows Server 2008 oder Windows 7 verwendet wird, so müssen zudem die zu verwendenden Verschlüsselungstypen in der Rubrik "[libdefaults]" angepasst werden:



```
default_tkt_encypes = AES256-CTS-HMAC-SHA1-96 AES128-CTS-HMAC-SHA1-96 RC4-HMAC  
default_tgs_encypes = AES256-CTS-HMAC-SHA1-96 AES128-CTS-HMAC-SHA1-96 RC4-HMAC  
permitted_encypes = AES256-CTS-HMAC-SHA1-96 AES128-CTS-HMAC-SHA1-96 RC4-HMAC
```

4.5.3.3.2 Anlegen der Datei "login.conf"

Die Datei "login.conf" wird so wie die Datei "krb5.ini" von JAAS benötigt. Eine detaillierte Beschreibung des Dateiformats ist in der JAAS-Dokumentation von Oracle zu finden (siehe externe Ressourcen rechts).

Für die Knowledge-Portal-SPNEGO-Komponente müssen in der "login.conf" auf jeden Fall die beiden Konfigurationen "spnego-client" und "spnego-server" definiert sein:

```
spnego-client {  
    com.sun.security.auth.module.Krb5LoginModule required;  
};  
  
spnego-server {  
    com.sun.security.auth.module.Krb5LoginModule required  
    storeKey=true  
    isInitiator=false;  
};
```

Im Knowledge-Portal ist bereits eine "login.conf" mit einer Standardkonfiguration enthalten. Die Datei ist nach dem Deploy des KP-WAR-Files im Webapp-Verzeichnis unter "WEB-INF\login.conf" zu finden und muss daher in der Regel nicht per Hand angelegt werden.

4.5.3.3.3 Anlegen eines Benutzerkontos für die Kerberos-Authentifizierung

Für die SPNEGO-Authentifizierung wird ein spezielles Windows-Benutzerkonto in der Domäne benötigt. In den Kontooptionen dieses Benutzerkontos sollten nur die folgenden Optionen angehakt sein:

- Kennwort läuft nie ab
- Konto wird für Delegierungszwecke vertraut

Die Vertrauenswürdigkeit für Delegierungszwecke muss je nach Domänenfunktionsebene (Modus, in dem die Windows-Domäne betrieben wird, bspw. "Windows 2000 kompatibel", "Windows Server 2003", "Windows Server 2008") an unterschiedlichen Stellen konfiguriert werden. Im Modus "Windows 2000" muss eine Checkbox in den Kontoeinstellungen (Reiter Account) gesetzt werden, im Modus "Windows Server 2003/2008" gibt es einen eigenen Reiter dafür (siehe Screenshot). Sollte der Reiter in den Kontoeinstellungen nicht angezeigt werden, so läuft die Domäne entweder noch im Modus "Windows 2000" oder es muss dem Benutzerkonto wie unten beschrieben erst ein SPN mittels des SetSPN-Kommandos zugeordnet werden. Danach sollte der Reiter "Delegation" angezeigt werden.



General | Address | Account | Profile | Telephones | Delegation | Organization

Delegation is a security-sensitive operation, which allows services to act on behalf of another user.

Do not trust this user for delegation

Trust this user for delegation to any service (Kerberos only)

Trust this user for delegation to specified services only

Use Kerberos only

Use any authentication protocol

Services to which this account can present delegated credentials:

Service Type	User or Computer	Port	Service N
--------------	------------------	------	-----------

Expanded

Add... Remove

OK Cancel Apply Help

Konfigurationsdialog von Windows Server 2008

Das Benutzerkonto muss weiterhin mit einem Kerberos Service Principal Name (SPN) verknüpft werden. Ein SPN ist ein eindeutiger Bezeichner für einen Dienst innerhalb der Kerberos Realm und muss folgendermaßen aufgebaut sein:

HTTP/[Hostname.zum.Zugriff.auf.die.Website]

Der SPN muss auf dem Server, der als Kerberos-KDC (Key Distribution Center; normalerweise identisch mit dem Active-Directory-Server) fungiert mittels des folgenden Kommandos mit dem Benutzerkonto verknüpft werden:

```
setspn -A [SPN] [accountname]
```

Beispiel:



```
setspn -A HTTP/host.mydom.com myaccount
```

Hierbei ist noch zu beachten, dass der SPN keine Port- oder Pfadangaben auf dem Server enthalten darf, d.h. auch wenn das KP im Browser über die Adresse "http://host.mydom.com:8080/xyz" aufgerufen wird, muss der SPN so wie im oben gezeigten Beispiel gesetzt werden. Dies bedeutet, dass pro Host immer nur ein SPN an einen Account gebunden werden kann.

Um sicherzustellen, dass derselbe SPN nicht mit mehreren Benutzerkonten verknüpft ist (wichtig, da die Browser ansonsten keine Tickets an den Tomcat senden), sollte danach das folgende Kommando aufgerufen werden:

```
setspn -L [accountname]
```

4.5.3.3.4 Überprüfen der Kerberos-Konfiguration

Wenn das Benutzerkonto für die Kerberos-Authentifizierung in der Domäne und die Konfigurationsdatei "krb5.ini" auf dem Rechner, auf dem auch der Tomcat-Webserver installiert ist, angelegt ist, so kann die Kerberos-Konfiguration mit dem folgenden Kommando überprüft werden:

```
kinit <Name des Benutzerkontos>@<Name der Kerberos-Realm in Großbuchstaben>
```

Beispiel:

```
kinit myaccount@MYDOM.COM
```

Das "kinit" Kommando ist unter Windows Bestandteil des JDK und genauso wie die Datei "java.exe" im Bin-Verzeichnis des JDK-Installationsverzeichnisses zu finden.

Wenn das Kommando ohne Fehler bzw. Exception ausgeführt wird und in der Ausgabe etwas von "New ticket is stored in cache file C:\...\krb5cc ..." steht, so ist die Kerberos-Konfiguration in Ordnung und es kann mit der Aktivierung der SPNEGO-Login-Komponente im Knowledge-Portal fortgefahren werden.

4.5.3.4 Bridge-Konfiguration

Vor der Aktivierung der SPNEGO-Login-Komponente muss in den Konfigurationsdateien der K-Infinity-KEM-Bridge-Dienste in der Rubrik "[KEMBridge]" die Eigenschaft "trustedLoginEnabled" auf den Wert "true" gesetzt werden:

```
[KEMBridge]
```

```
...
```

```
trustedLoginEnabled=true
```

```
..
```

Danach müssen die Bridge-Dienste, deren INI-Dateien geändert wurden, durchgestartet werden.

4.5.3.5 Aktivieren der SPNEGO-Login-Komponente

Um die SPNEGO-Login-Komponente zu aktivieren muss zunächst die Datei "spnego.properties" im WAR-File des Knowledge-Portals angepasst werden (die genaue Beschreibung der referenzierten Dateien und des Benutzerkontos ist im vorangehenden Kapitel beschrieben):

```
spnego.path.krb5conf=<Pfad zur Datei "krb5.ini">
```

```
spnego.path.loginconf=<Pfad zur Datei "login.conf">
```



```
spnego.preauth.username=<Name des Benutzerkontos für die Authentifizierung>  
spnego.preauth.password=<Kennwort des Benutzerkontos für die Authentifizierung>  
#Loglevel für SPNEGO-Filter: 900 - warn / 1 - debug  
spnego.loglevel=1
```

Das Kennwort sollte nicht im Klartext in die Properties-Datei eingetragne werden. Statt dessen sollte mittels der Verschlüsselungsfunktion auf der KP-Admin-Seite (/admin.skat) ein verschlüsselter Property-Wert erzeugt und dieser anstelle des Kennworts eingetragen werden (verschlüsselte Property-Werte beginnen mit der Zeichenkette "decrypt:").

Außerdem muss noch das XML für die KP-Komponentenkonfiguration im Wissensnetz (Attribut "componentsXml" am KP-Konfigurationsbegriff) bearbeitet werden. Darin muss das XML-Element "<StandardLogin/>" durch das folgende XML-Element ersetzt werden:

```
<SPNEGOLogin  
  showLoginBox="true"  
  showLoginPage="true"  
  showGuestLoginLink="false"  
  guestLoginAllowed="false"  
  nonSSOFallback="false"  
  type="2" />
```

Damit ist die Konfiguration der SPNEGO-Login-Komponente abgeschlossen. Die Änderung der Datei "spnego.properties" im WAR-File wird erst mit dem Deploy des geänderten WAR-Files aktiv, d.h. für bereits aktive KP-Webapps muss erneut ein Deploy im Tomcat durchgeführt werden.

Die SPNEGO-Login-Komponente kann die Benutzer nur dann automatisch im KP anmelden, wenn in dem dazugehörigen Wissensnetz ein Benutzerobjekt existiert, dessen Benutzername identisch mit dem Namen des Windows-Benutzerkontos (ohne Domänenangabe) ist. Kann der Benutzer nicht automatisch angemeldet werden, so wird der Benutzer auf das Login-Formular weitergeleitet.

4.5.3.6 Browser-Konfiguration

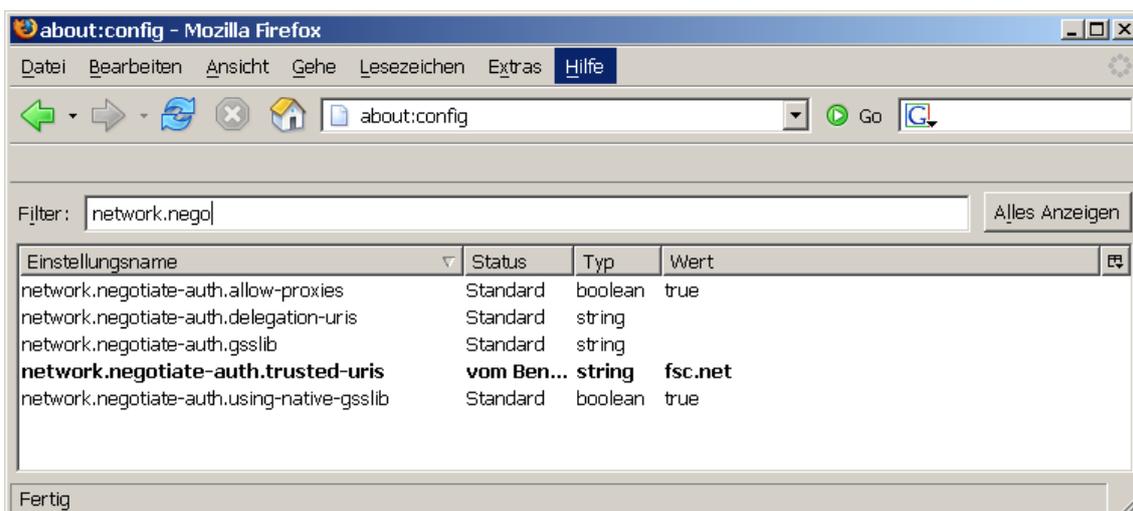
4.5.3.6.1 Internet-Explorer

Die Authentifizierung über SPNEGO mittels Kerberos wird vom Internet-Explorer seit Version 6.01 unterstützt. Bei der Browser-Konfiguration ist auf folgendes zu achten:

- Die HTTP-Adresse, über die das Knowledge-Portal aufgerufen wird, muss vom IE der **Intranetzone** zugeordnet werden, da die IE-Sicherheitsrichtlinien standardmäßig nur in dieser Zone die automatische Authentifizierung erlauben. Normalerweise sollte dies automatisch geschehen. Falls der IE die Adresse nicht automatisch der Intranetzone zuordnen kann, so kann die Adresse manuell hinzugefügt werden (Internetoptionen > Reiter "Sicherheit" > Lokales Intranet > Sites > Erweitert)
- In den IE-Einstellungen (Internetoptionen > Reiter "Erweitert" > Sicherheit) muss die Option "integrierte Windows-Authentifizierung aktivieren" angehakt sein. Diese Option ist in der IE-Standardkonfiguration aktiviert.

4.5.3.6.2 Firefox / Mozilla

Die automatische Authentifizierung über SPNEGO/Kerberos ist in Firefox standardmäßig nicht aktiviert und muss durch Hinzufügen eines Rechnernamens bzw. eines Windows-Domänennamens zu einer Whitelist aktiviert werden. Hierzu muss der Domain-/Hostname in den Einstellungen (URL "about:config") unter dem Schlüssel "**network.negotiate-auth.trusted-uris**" eingetragen werden. Außerdem gibt es noch einige weitere Konfigurationsschlüssel, welche die SPNEGO/Kerberos-Authentifizierung betreffen. Das folgende Bild zeigt einige relevanten Konfigurationsschlüssel (wir übernehmen keine Gewähr für die Vollständigkeit dieser Liste):



Die SPNEGO-Authentifizierung funktioniert in Firefox sowohl unter Windows als auch unter Linux. Unter Linux muss eine funktionierende Kerberos-Konfiguration für die Integration mit der Windows-Domäne eingerichtet ist.

Wichtiger Hinweis: Wenn man bpsw. die Domain "mydom.com" in der Firefox-Konfig eingetragen hat, so funktioniert die SPNEGO-Authentifizierung nur dann, wenn das KP mit dem vollqualifizierten Hostnamen aufgerufen wird, also z.B. "http://machineA.mydom.com:8080/kp". Bei einem Aufruf nur über den Rechnernamen ("http://machineA:8080/kp") führt der Firefox keine Authentifizierung durch.

4.5.3.6.3 Google Chrome

Google Chrome unterstützt die automatische Authentifizierung via SPNEGO/Kerberos sowohl unter Windows als auch unter Linux.

Unter Linux müssen ggf. zusätzliche Bibliotheken installiert werden (GSSAPI) und ggf. auch noch zusätzliche Parameter beim Start des Browsers angegeben werden. Genaue Infos hierzu sind in einem Design Dokument von Chrome beschrieben (siehe Link in der rechten Spalte).

4.5.3.7 Bekannte Probleme

4.5.3.7.1 Verbindungsprobleme / HTTP 400 Bad Request

Bei großen Domains mit vielen Benutzern und Gruppen kann die Größe des Kerberos-Tickets stark wachsen. Da das Ticket im HTTP-Header übertragen wird, kann es passieren, dass die maximale Größe HTTP-Header überschritten wird.



Der Internet-Explorer zeigt hierbei Verbindungsprobleme an. Klappt man die Details auf, wird ein "HTTP 400"-Fehler angezeigt.

Mit Firefox und Firebug kann man im Netzwerk-Reiter zunächst eine 401-Antwort und danach eine 400-Antwort beobachten. Schaut man sich den Request-Header der 400-Antwort genau an, so sieht man eine sehr großes Kerberos-Ticket im

Der Standardwert für diese Größe ist 4096 Bytes. Er kann im Tomcat über den Connector in der Datei *server.xml* angepasst werden

Beispiel:

```
<Connector connectionTimeout="20000" port="8080" protocol="HTTP/1.1"
    redirectPort="8443" maxHttpHeaderSize="53464"/>
```

4.5.3.7.2 Fallback auf Standard-Login mit Firefox und Chrome

In der KP-Konfiguration der SPNEGO-Komponente kann das Attribut "nonSSOFallback" auf "true" gesetzt werden. Dies bewirkt, dass das normale Login-Formular angezeigt wird, wenn der Benutzer nicht per SSO authentifiziert werden konnte.

Aufgrund unterschiedlicher Verhaltensweisen der verschiedenen Browser funktioniert dieses Fallback momentan nur mit dem Internet Explorer.

Dies liegt daran, dass das Protocol zum Einloggen aus verschiedenen Schritten bestehen kann, in denen der Server ein HTTP-401 (Forbidden) zusammen mit anderen Informationen an den Browser zurückschickt. Der Internet Explorer reagiert hierauf mit weiteren Verbindungen, an deren Ende klar ist, dass eine Authentifizierung nicht möglich ist.

Firefox und Chrome geben allerdings zu früh auf und zeigen nur eine leere Seite an.

4.6 Logout-Button

Die Anzeige des Logout-Buttons wird über eine benutzerdefinierte Rechtesystemoperation namens "logout" gesteuert. Nachdem die Operation mit dem Operationssymbol "logout" im Knowledge-Builder definiert wurde, kann man z.B. eine Regel im Rechtesystem anlegen, welche die Operation bei allen Benutzern außer dem Gast-Benutzer erlaubt.

5 Konfiguration der Suchen

5.1 Suchkonfiguration ab Version 3.5

5.1.1 Einführung

Im Knowledge-Portal ab Version 3.5 ist die Standard-Begriffssuche in vielerlei Hinsicht nicht mehr so starr und "festzementiert", wie dies in früheren Versionen der Fall war. Viele Aspekte der Suche, wie z.B. die bei einer Suchanfrage ausgeführten Teilsuchen, die durchsuchten Datenquellen oder das Layout der Suchergebnisseite, sind jetzt konfigurierbar.

Zur Suche in unterschiedlichen Datenquellen werden im Knowledge-Portal sogen. Suchadapter



verwendet. Standardmäßig bringt das Knowledge-Portal Suchadapter für die Suche im Wissensnetz (K-Infinity- und NextBot-Suchen) und für die Suche im Index des Nutch-Webcrawlers mit. Für weitere Datenquellen können auf Anfrage Suchadapter erstellt werden.

Mittels des K-Infinity-Suchadapters lassen sich alle im Wissensnetz definierten Suchen ausführen. Details zur Konfiguration von Suchen im Wissensnetz sind im Handbuch des Knowledge-Builder 3.1, Kapitel 7 zu finden. Mittels des Suchadapters für den Nutch-Webcrawler lassen sich indizierte Webseiten aus einem Intra- oder Internet durchsuchen. Details zur Installation und Konfiguration des Nutch-Webcrawlers sind unter <http://lucene.apache.org/nutch/> zu finden.

Die Konfiguration der Teilsuchen, die bei der Ausführung einer Suchanfrage über das Suchformular im Knowledge-Portal ausgeführt werden sollen, kann im Wissensnetz in Form eines XML-Dokuments hinterlegt werden (siehe nächstes Kapitel). Im Knowledge-Portal ist aber auch eine Standardkonfiguration für die Suche hinterlegt, die dafür sorgt, dass sich die Suche weitestgehend so verhält, wie dies in früheren Knowledge-Portal-Versionen der Fall war. Ausnahmen hiervon werden im Kapitel "Migration der Suchkonfiguration" beschrieben.

Suchergebnisse
Suche nach „mustermann“

Semantische Suche	Dokumentensuche
Max Mustermann ■■■■	Keine Treffer
Meinten Sie...?	
Manchester ■■■■	

5.1.2 XML-Suchkonfiguration

Die Teilsuchen, die ausgeführt werden sollen, wenn von einem bestimmten Suchfeld aus eine Suchanfrage gestartet wird, können in einem XML-Dokument beschrieben werden. Das hierfür verwendete XML-Format wird in der folgenden DTD-Datei (XML Document Type Definition) beschrieben:



searchConfig.dtd

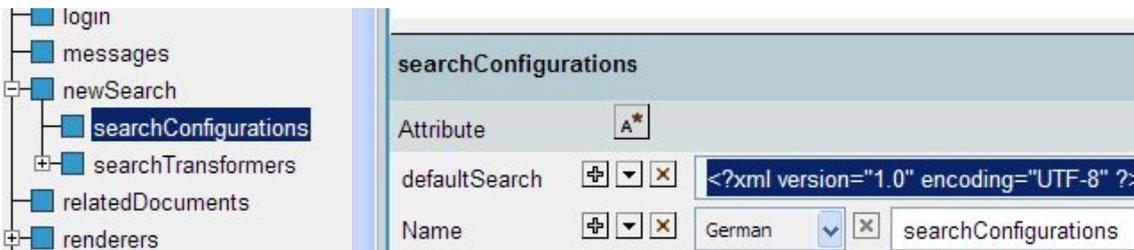
Ein XML-Dokument zur Beschreibung einer Suchkonfiguration sieht demnach in etwa so aus:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE EditConfig PUBLIC "-//intelligent views GmbH//DTD searchmanager
  searchConfig" "searchConfig.dtd">
<SearchConfiguration id="defaultSearch">
  <SearchDescriptions>
    <Search id="topicSearch" componentId="kinfinity">
      ...
    </Search>
    <Search id="intranetSearch" componentId="websearch">
      ...
    </Search>
  </SearchDescriptions>
</SearchConfiguration>
```



```
</Search>  
</SearchDescriptions>  
</SearchConfiguration>
```

Am Anfang des XML wird im Element SearchConfiguration im Attribut "id" der Name einer Suche angegeben, für welche diese Konfiguration gelten soll. Damit das Knowledge-Portal diese Suchkonfiguration berücksichtigt, muss das XML im Wissensnetz unter dem Begriff "kpConfig -> newSearch -> searchConfigurations" in einem Zeichenketten-Attribut abgelegt werden. Der Name des Zeichenketten-Attributs muss der im XML angegebenen SearchConfiguration-ID entsprechen (im Beispiel "defaultSearch"):



Derzeit ist im Knowledge-Portal nur der Wert "defaultSearch" als mögliche SearchConfiguration-ID verwendbar, der mit der Knowledge-Portal-Standardsuche (Box "Suche" in der linken Spalte des Web-Frontends) verknüpft ist.

Innerhalb eines SearchConfiguration-Elements werden dann die auszuführenden Teilsuchen in Form eines Search-Elements beschrieben. Zu jedem Search-Element muss ein ID-Attribut angegeben werden, damit die Teilsuche innerhalb der jeweiligen Suchkonfiguration eindeutig benannt ist. In dem Attribut "componentId" wird die ID des Suchadapters angegeben, welches zur Ausführung der jeweiligen Teilsuche verwendet werden soll. Als Wert für die componentId kann im Knowledge-Portal nur "kinfinity" angegeben werden, womit der Suchadapter für die K-Infinity- und NextBot-Suchen gemeint ist. Andere Werte können nur dann verwendet werden, wenn unter dem entsprechenden Wert ein Suchadapter konfiguriert ist. Dies wird in einem der folgenden Kapitel behandelt.

Eine weitere Beschreibung der in der Suchkonfiguration verwendbaren XML-Elemente und Suchadapter ist im Kapitel "Referenz" dieses Handbuchs zu finden.

5.1.3 Suchparameter

Wie schon in der Einführung angesprochen, können in der XML-Suchkonfiguration für jede Teilsuche die Suchparameter konfiguriert werden, die bei der Ausführung einer Teilsuchanfrage berücksichtigt werden sollen. Dies kann mittels der Unterelemente QueryParam, UserQueryParam und ComputedQueryParamComputer innerhalb einer Teilsuchkonfiguration definiert werden. Eine ausführliche Beschreibung dieser Elemente ist in der Referenz der Suchkonfiguration zu finden.

Die Elemente können in beliebiger Reihenfolge hintereinander angegeben werden. Im einfachsten Fall sollen einfach die vom Benutzer angegebenen Suchparameter (siehe Suchformular) in der Suchanfrage der Teilsuche berücksichtigt werden. In diesem Fall gibt man einfach das Element `<UserQueryParams />` in der Konfiguration an.

Sollen bestimmte Suchparameter manuell definiert werden (z.B. searchName), so können diese über das Element QueryParam angegeben werden. Beispiel:

```
<QueryParam name="searchName" value="trigramSearch" />
```



Die Elemente können auch kombiniert werden, wodurch es bspw. möglich ist, Suchparameter aus der Benutzeranfrage zu überschreiben. In dem folgenden Beispiel wird der Suchparameter "searchName", der auch vom Benutzer angegeben werden kann (Drop-Down-Feld im Suchformular) durch einen fest definierten Wert überschrieben:

```
<QueryParams>  
  <UserQueryParams />  
  <QueryParam name="searchName" value="trigramSearch" />  
</QueryParams>
```

Die Reihenfolge in der die Elemente angegeben werden, ist dabei natürlich wichtig. Würde man das QueryParam-Element vor das UserQueryParams-Element stellen, würde der Suchparameter "searchName" durch den vom Benutzer eingegebenen Suchparameter überschrieben.

Als drittes Element zur Konfiguration der Suchparameter steht noch `<ComputedQueryParam />` zur Verfügung. Mit diesem Element kann eine Java-Klasse angegeben werden, in der ein Programm zur Generierung von Suchparametern implementiert ist. Insbesondere für Entwickler eröffnet sich dadurch die Möglichkeit, Suchanfragen für Teilsuchen programmatisch zu erzeugen. Als Eingabe kann dabei auf die Suchanfrage des Benutzers, die Suchkonfiguration und die bereits berechneten Teilsuchergebnisse zugegriffen werden. Details hierzu finden sich wiederum in der Referenz.

5.1.4 Suchformular

5.1.4.1 Überblick

Die Suchanfrage für die Standard-Begriffssuche wird über das im Knowledge-Portal angezeigte Suchformular gestellt. Die Eingabefelder im Suchformular werden beim Anklicken des Starten-Buttons jeweils in bestimmte Suchparameter umgewandelt, die dann gesammelt als Suchanfrage vom Knowledge-Portal ausgewertet werden.

Die einzelnen Eingabefelder und die damit verknüpften Suchparameter werden in den folgenden Unterkapiteln erläutert.



Knowledge-Portal-Suchfeld

5.1.4.2 Texteingabefeld für Suchanfrage

Im Sucheingabefeld hat der Benutzer die Möglichkeit, nach einzelnen Termen (einzelne Wörter) oder nach Phrasen (Folgen von Wörtern) zu suchen.

Als Terme werden alle im Eingabefeld eingegebenen Wörter berücksichtigt, die durch Leerzeichen, Tab-Zeichen oder Newline-Zeichen voneinander getrennt sind. Ausgenommen davon sind die Wörter AND und OR, denen als Schlüsselwort bei der Interpretation der Suchanfrage eine besondere Bedeutung zukommt (dazu später mehr). Als Phrase wird eine Folge von Termen interpretiert, die in doppelte Anführungszeichen eingeschlossen ist. Beispiel:

```
wort1 wort 2 "und eine Phrase"
```

Die obenstehende Zeichenkette wird vom Knowledge-Portal in die drei Terme "wort1", "wort" und "2" sowie in die Phrase "und eine Phrase" zerlegt.

Terme und Phrase können durch die Operatoren AND und OR miteinander verknüpft werden. Dabei sind auch Ausdrücke erlaubt, in denen sowohl AND- als auch OR-Operatoren vorkommen, allerdings ist dabei keine freie Klammerung erlaubt. Statt dessen hat AND immer Vorrang vor OR, d.h. mit AND verknüpften nebeneinander stehende Terme und Phrasen werden automatisch zu einer Gruppe zusammengefasst. OR-Operatoren wirken dann nur noch zwischen diesen AND-Gruppen und den verbleibenden Termen und Phrasen. Zum Beispiel wird der Ausdruck

```
wort1 OR wort2 AND wort3 AND wort4 OR wort5"
```



folgendermaßen ausgewertet:

wort1 OR (wort2 AND wort3 AND wort4) OR wort5

Termen oder Phrasen kann außerdem ein Plus- ("+") oder Minus-Zeichen ("-") vorangestellt werden, um anzuzeigen, dass ein Objekt den Term oder die Phrase enthalten muss (bzw. nicht enthalten darf), um in das Suchergebnis aufgenommen zu werden. Hierbei gilt, dass ich das Plus oder Minus immer nur auf den darauffolgenden Term bzw. die darauffolgende Phrase und niemals auf eine mit AND oder OR verknüpfte Gruppe von Termen bezieht.

Die Sucheingabe des Benutzers wird vom Knowledge-Portal analysiert, zerlegt und in zerlegter Form (Parsing-Tree) als Parameter "searchTree" in die Suchanfrage des Benutzers aufgenommen.

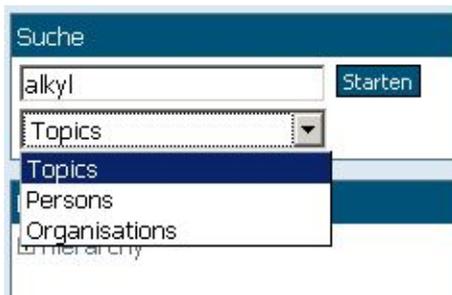
WICHTIG: Trotz der einfachen Syntax kann nicht garantiert werden, dass sich alle hier beschriebenen Operatoren auf jede mit dem Knowledge-Portal durchsuchbare Datenquelle abbilden lassen. Die genaue Abbildung der Syntax auf die Suche in einer Datenquelle ist daher der Beschreibung des jeweiligen Suchadapters zu entnehmen.

5.1.4.3 Auswahlfeld für Suchname

Damit der Benutzer auswählen kann, welche Suche für die Beantwortung seiner Suchanfrage ausgeführt werden soll, gibt es die Möglichkeit, eine Drop-Down-Feld mit einer Liste von Suchen im Suchformular anzeigen zu lassen.

In dem Drop-Down-Feld werden automatisch alle K-Infinity-Suchkonfiguration angezeigt, deren interner Name mit dem Präfix "textSearch." beginnt. Der interne Name der vom Benutzer ausgewählten Suchkonfiguration wird als Suchparameter "searchName" in die Suchanfrage des Benutzers aufgenommen. Wird nur eine Suchkonfiguration mit dem Präfix "textSearch." gefunden, so wird das Dropdown-Feld nicht angezeigt und automatisch bei jeder Suche der interne Name dieser K-Infinity-Suchkonfiguration als Suchparameter "searchName" in die Suchanfrage des Benutzers aufgenommen.

Das Präfix ist in der Konfigurationsdatei "names.xml" der Web-Anwendung unter der Eigenschaft "kpConfig.search.textSearchPrefix" festgelegt. Um das Präfix zu ändern, überschreiben Sie es in der Datei "customized.xml".



Dropdown-Feld zur Auswahl der Suchkonfiguration

Falls nichts anderes konfiguriert wird, ergibt sich das Label der Suchkonfiguration, das in der Dropdown-Liste angezeigt wird, aus dem Namen der Suchkonfiguration im Wissensnetz ab dem Ende des Präfixes.

Für eine Suchkonfiguration mit dem Namen "textSearch.Persons" wird in der Dropdown-Liste "Persons" angezeigt. Dieser Name wird sowohl im englisch- als auch im deutschsprachigen Modus angezeigt. Um völlig unabhängig vom Namen der Suchkonfiguration im Wissensnetz ein deutsch- und englischsprachiges Label für die Suchkonfiguration festzulegen, kann am Konfigurationsobjekt "kpConfig > messages" im Attribut "msg" bzw. "msg_de" ein Mapping



zwischen dem Namen der Suchkonfiguration im Wissensnetz und dem gewünschten Label für die jeweilige Frontend-Sprache eingetragen werden. Um beispielsweise den deutschen Namen der Suche "*textSearch.persons*" zu definieren, muss in das Attribut "*msg_de*" das folgende Mapping eingetragen werden:

```
textSearch.Persons=Personen
```

... wobei natürlich "*textSearch.Persons*" durch den Namen der jeweiligen Suchkonfiguration im Wissensnetz und "*Personen*" durch den jeweiligen Namen für die Suche im Knowledge-Portal ersetzt werden muss. Analog dazu kann im Attribut "*msg*" das Label der Suche für den englischsprachigen Modus des Web-Frontends festgelegt werden.

5.1.4.4 Auswahl der Sortierkriterien

Unter der Rubrik "Dokumente sortieren nach" werden im Suchformular zwei Radio-Buttons angezeigt, mit denen der Benutzer auswählen kann, ob die Suchtreffer bei der Anzeige der Suchergebnisse nach Trefferqualität oder nach Datum sortiert sein sollen.

Anhand der Auswahl des Benutzers wird der Suchparameter "sortOrder" mit bestimmten Sortierkriterien und Sortierreihenfolgen in die Suchanfrage aufgenommen.



Der Button "Qualität" ist mit den folgenden Sortierkriterien belegt:

1. Sortierkriterium: "quality" (Sortierreihenfolge absteigend)
2. Sortierkriterium: "label" (Sortierreihenfolge aufsteigend)

Der Button "Datum" ist mit den folgenden Sortierkriterien belegt:

1. Sortierkriterium: "date" (Sortierreihenfolge absteigend)
2. Sortierkriterium: "label" (Sortierreihenfolge aufsteigend)

Ob die Ergebnisse einer Teilsuche nach diesen Kriterien sortiert werden können, hängt davon ab, ob die Sortierkriterien in der jeweilige Teilsuche bzw. in dem jeweiligen Suchadapter definiert sind. Die Sortierkriterien "quality" und "label" sind im K-Infinity-Suchadapter standardmäßig definiert. Das Sortierkriterium "date" hingegen muss als Extra-Sortierkriterium in der Suchkonfiguration definiert werden. In der Standard-Suchkonfiguration des Knowledge-Portals ist das Sortierkriterium für die Dokumentensuche selbstverständlich definiert. Wird die Suchkonfiguration angepasst, muss man dies jedoch berücksichtigen.

Details zum sortOrder-Parameter, zu den standardmäßig verwendbaren Sortierkriterien und zur Konfiguration zusätzlicher Sortierkriterien sind in der Referenz der Suchadapter sowie in der Referenz der XML-Suchkonfiguration zu finden.

5.1.4.5 Spezielle Eingabefelder für Dokumentensuchen

5.1.4.5.1 Überblick

Wenn in der Suchkonfiguration des Knowledge-Portals auch Dokumentensuchen konfiguriert sind (d.h. Suche, die Dateien finden), so können im Suchformular zusätzliche Eingabefelder für



die speziellen Suchparameter dieser Suchen eingeblendet werden (z.B. Änderungsdatum und Dateityp). Diese Eingabefelder sind standardmäßig ausgeblendet, da Dokumentsuchen erst nach der Installation einer speziellen Suchmaschine (z.B. NextBot) ausgeführt werden können.

Damit die Auswahlfelder angezeigt werden, müssen mit dem Knowledge-Builder die folgenden Einstellungen in der Knowledge-Portal-Konfiguration im Wissensnetz vorgenommen werden:

1. Legen Sie am Begriff "KP-Konfiguration" den Unterbegriff "applicationComponents" an
2. Erzeugen Sie in der Schemadefinition für Unterbegriffe von applicationComponents ein Attribut vom Typ "Zeichenkette" mit dem Namen "class"
3. Legen Sie am Begriff "applicationComponents" den Unterbegriff "search" an
4. Legen Sie am Begriff "search" das Attribut "class" an und geben Sie den folgenden Klassennamen als Wert ein:
com.iviews.knowledgeportal.search.extensionfilter.FEFSearchComponent

Danach ist ein Neustart des Knowledge-Portals oder ein Reload der Wissensnetz-Konfiguration mit Beenden der aktuellen Session (Klick auf den Header) notwendig.

5.1.4.5.2 Dateityp-Auswahlfelder

Die Auswahlfelder für Dateitypen ermöglichen die Einschränkung von Dokumentsuchen (z.B. NextBot-Volltextsuche) auf bestimmte Dateiendungen.

<input type="checkbox"/>	Adobe PDF
<input type="checkbox"/>	Mircrosoft Excel Dateien
<input type="checkbox"/>	HTML-Dateien
<input type="checkbox"/>	Text-Dateien
<input type="checkbox"/>	Microsoft Word Dateien

Mit einem Auswahlfeld können mehrere Dateiendungen verknüpft sein (z.B. "doc", "docx" und "dot" mit Microsoft Word Dateien). Die Menge der Erweiterungen aller ausgewählten Dateitypen wird als Suchparameter namens "fileExtensionFilter" in die Suchanfrage des Benutzers aufgenommen. Es ist jedoch zu beachten, dass dieser Suchparameter nur bei Dokumentensuchen (z.B. NextBot-Volltextsuche) ausgewertet werden kann. Details zu dem Suchparameter sind in der Referenz der K-Infinity-Suchkomponente zu finden.

Die angezeigten Auswahlfelder werden ebenfalls am Begriff "*kpConfig* > *applicationComponents* > *search*" definiert. Hierfür muss in der Schemadefinition für den Begriff "search" ein Zeichenkettenattribut namens "*extensionFilters*" angelegt werden.

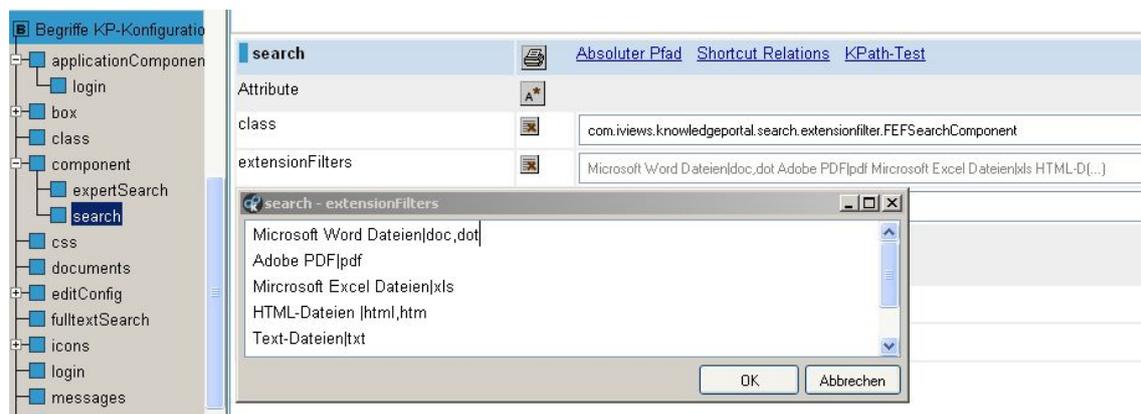


Als Wert der Eigenschaft muss ein Text in der folgenden Form eingegeben werden:

```
Microsoft Word Dokumente|doc,docx  
Text-Dateien|txt  
Source-Code|java,c,cc,h  
HTML-Dateien|html,htm  
Mails|mail
```

Pro Zeile wird ein Auswahlfeld definiert. Zunächst wird der Name des Auswahlfeldes angegeben. Durch "|" getrennt folgen darauf die mit dem Auswahlfeld verknüpften Dateiendungen. Die Dateiendungen selbst werden durch "," getrennt. Nach einem Zeilenumbruch beginnt die Definition des nächsten Auswahlfeldes.

Hinweis: Beim zu setzenden Wert des Schlüssels handelt es sich um einen mehrzeiligen Text, weshalb empfohlen wird, die Konfiguration im Netz vorzunehmen (siehe dazu "Konfiguration im Netz"). In diesem Fall kann ein mehrzeiliges Textfeld verwendet werden. Sollte die Konfiguration über die Datei "*customized.xml*" erfolgen, müssen die Zeilenumbrüche durch "\n" angegeben werden!



Konfiguration für Dateiendungsfilter im Wissensnetz

5.1.4.5.3 Texteingabefelder für Änderungsdatum von Dokumenten

Unter der Rubrik "Dokumente mit Änderungsdatum" werden im Suchformular zwei Eingabefelder angezeigt, in denen der Benutzer jeweils ein Datum eingeben kann. Das bei der Eingabe zu verwendende Datumsformat wird in Klammern unter den Eingabefeldern angezeigt. Der Wert im Eingabefeld "Von" wird als Suchparameter "fromDateFilter", der Wert im Eingabefeld "Bis" als Suchparameter "toDateFilter" in die Suchanfrage aufgenommen. Eine Eingabe in diesen Feldern führt dazu, dass Dokumentsuchen nur noch solche Dokumente bei der Suche berücksichtigen, deren Änderungsdatum im angegebenen Zeitraum liegt. Details zu den Suchparameter "fromDateFilter" und "toDateFilter" sind in der Beschreibung des Suchadapters zu finden, der zur Ausführung der Dokumentsuche verwendet wird.

Dokumente mit Änderungsdatum:

Von: Bis:

(Datumsformat TT.MM.JJJJ)



5.1.5 Standard-Suchkonfiguration

Für die Standard-Begriffssuche im Knowledge-Portal (ID "defaultSearch") ist im Knowledge-Portal eine Standard-Suchkonfiguration hinterlegt, die verwendet wird, wenn in der KP-Konfiguration im Wissensnetz keine angepasste Suchkonfiguration definiert ist.

In der Standardkonfiguration sind die folgenden drei Teilsuchen definiert:

- Semantische Suche
- "Meinten Sie" Suche (Interner Name "trigramSearch")
- Volltextsuche (Interner Name "fulltextSearch")

Für die semantische Suche ist der Parameter "searchName" in der Suchkonfiguration nicht angegeben, d.h. der Name der Benutzer wird aus der Benutzersuchanfrage entnommen. Der Benutzer kann den Namen der Suche über das Drop-Down-Feld unterhalb des Sucheingabefeldes in der Box "Suche" auswählen (siehe Bild). Im Drop-Down-Feld werden alle Suchen aus der K-Infinity-Suchkonfiguration angezeigt, deren interner Name mit dem Prefix "textSearch." beginnt und die als KEM-Suchen exportiert sind (siehe unten).

Knowledge-Portal-Suchfeld

Für jede der Teilsuchen muss im Wissensnetz eine Suchkonfiguration mit dem entsprechenden internen Namen definiert sein. Außerdem muss die betreffende Suchkonfiguration im Reiter KEM-Suchen in die Spalte "verwendete Konfigurationen" übernommen werden (Knowledge-Builder im Menü "Datei" -> Einstellungen -> Reiter "Suchen" bzw. Reiter "KEM-Suchen"). Im Basis-Wissensnetz für das Knowledge-Portal sind die entsprechenden Suchkonfigurationen bereits vordefiniert.

In der Standard-Suchkonfiguration ist außerdem festgelegt, dass die Treffermenge der semantischen Suche von der Treffermenge der "Meinten Sie"-Suche abgezogen wird. Dies ist sinnvoll, da die "Meinten Sie"-Suche ja nur solche Objekte zurückliefern soll, die etwa durch Schreibfehler nicht von der semantischen Suche gefunden wurden.

Die Volltextsuche kann nur dann ausgeführt werden, wenn zusammen mit dem Knowledge-Portal die NextBot-Suchmaschine (<http://brainbot.com/produkte/nextbot/>) der Firma BrainBot Technologies installiert und im Wissensnetz konfiguriert ist. Diese Suchmaschine ist nicht im Standardumfang des Knowledge-Portals enthalten.

Die Auswahlfelder für die Sortierreihenfolge sowie die Filter für Dateityp und Änderungsdatum werden standardmäßig nur bei Ausführung der Volltextsuche (NextBot-Suche) berücksichtigt, da diese Dokumente durchsucht. Die semantische Suche und die "Meinten Sie"-Suche liefert jedoch Wissensnetzobjekt, zu denen standardmäßig kein Änderungsdatum

Bei der Konfiguration der Suchfunktion wird zwischen der Konfiguration der Suche (Art und Konfiguration der auszuführenden Teilsuchen) und der Konfiguration der Suchergebnisseite (Reihenfolge, Layout und Sortieroptionen der anzuzeigenden Teilsuchergebnisse) unterschieden. Auf die Konfiguration der Suche wird im Kapitel 5.2 eingegangen, auf die Konfiguration der Suchergebnisseite im Kapitel 5.3.

5.1.6 Front-End bezogene Suchkonfiguration

5.1.6.1 Überblick

Ab Knowledge-Portal Version 3.5.0 ist es möglich, das Layout und die Darstellungsreihenfolge von den auf der Suchergebnisseite angezeigten Teilergebnissen zu konfigurieren. Neben dem Layout lassen sich außerdem die möglichen Optionen für die Sortierung und zusätzliche Aktionen für das Gesamtergebnis, das Teilergebnis und die einzelnen Treffer konfigurieren. Eine genauere Beschreibung folgt in den nächsten Kapiteln.

Die Konfiguration muss in einem XML-Dokument beschrieben werden, welches mit dem Wurzelement "Search" zu beginnen hat:

```
<Search xmlns="http://www.iviews.com/knowledgeportal/kpConfig">
  ...
</Search>
```

Das XML-Dokument muss in der Wissensnetz-Konfiguration unter dem Begriff "newSearch" im Attribut "xmlConfig" abgespeichert werden (siehe Bild).



Die möglichen Unter-elemente des Search-Elements werden genauer in der Referenz der Suchergebnisseitenkonfiguration (siehe Kapitel Referenz) beschrieben. Außerdem kann die XML-Struktur dem folgenden XML-Schema entnommen werden:

Knowledge-Portal-Version 3.5.0:



kpConfig.xsd

5.1.6.2 Zusätzliche Suchen

Wenn ein Benutzer im Suchformular eine Suchanfrage stellt, so führt das Knowledge-Portal immer eine Suche namens "defaultSearch" aus, für die im Knowledge-Portal auch eine Standard-Suchkonfiguration hinterlegt ist (siehe Kapitel zur XML-Suchkonfiguration).

Das Knowledge-Portal kann angewiesen werden, weitere Suchen auszuführen, wobei die Namen der dazugehörigen Suchkonfigurationen in einem Element namens "AdditionalDefaultSearches" angegeben werden müssen. Beispiel:

```
<Search ... >
  <AdditionalDefaultSearches>mySearch1</AdditionalDefaultSearches>
  <AdditionalDefaultSearches>mySearch2</AdditionalDefaultSearches>
  ...
</Search>
```

Damit das Knowledge-Portal diese Suchen ausführen kann, müssen Suchkonfigurationen mit den jeweiligen Namen definiert sein. Details hierzu sind im Kapitel "XML-Suchkonfiguration" zu finden.

Die Suchergebnisse der zusätzlichen Suchen werden auf der Suchergebnisseite in separaten Reitern dargestellt, wobei das Suchergebnis der "defaultSearch" immer als erstes angezeigt wird.

5.1.6.3 Layout-Konfiguration der Suchergebnisseite

Das Suchergebnis für eine Suchkonfiguration kann sich - abhängig von Anzahl der in der XML-Suchkonfiguration definierten Teilsuchen - aus einem oder mehreren Teilsuchergebnissen zusammensetzen. Die Anordnung der Teilsuchergebnisse auf der Suchergebnisseite muss für jede Suchkonfiguration mittels des Elements "ResultViewConfig" beschrieben werden.

Innerhalb eines ResultViewConfig-Elements muss als erstes ein sog. Container-Element definiert werden, welches das Layout der darin enthaltenen Elemente bestimmt. Derzeit ist als einziges Container-Element das Element "TwoColumnContainer" verfügbar, in neuen Versionen des Knowledge-Portals können aber durchaus noch weitere hinzukommen.

Innerhalb von Container-Elementen können dann andere Elemente eingefügt werden, welche die auf der Seite sichtbaren Komponenten repräsentieren. Elemente dieser Art sind z.B. das ResultHeader-Element, welches den Überschriftenbereich einer Suchergebnisseite darstellt und das SearchPartResult-Element, welches ein Teilsuchergebnis darstellt. In angepassten Knowledge-Portals oder in neuen Knowledge-Portal-Versionen können neben diesen Elementen durchaus noch weitere hinzukommen.

Das folgende Beispiel zeigt die Standardkonfiguration für die Suchergebnisseite der default-Search im Knowledge-Portal, bei der die Teilsuchergebnisse in ein TwoColumnContainer mit zwei Zeilen, einer ColumnRow und einer TwoColumnRow, angeordnet werden:



```
<Search xmlns="http://www.iviews.com/knowledgeportal/kpConfig">
  <ResultViewConfig id="defaultSearch">
    <TwoColumnContainer>
      <ColumnRow>
        <ResultHeader />
      </ColumnRow>
      <TwoColumnRow>
        <LeftColumn>
          <SearchPartResult id="semanticSearch" />
          <SearchPartResult id="trigramSearch" />
        </LeftColumn>
        <RightColumn>
          <SearchPartResult id="fulltextSearch" />
        </RightColumn>
      </TwoColumnRow>
    </TwoColumnContainer>
  </ResultViewConfig>
</Search>
```

Der ResultHeader kann die gesamte verfügbare Breite der Suchergebnisseite einnehmen, da er in einer ColumnRow enthalten ist. Die Teilsuchergebnisse (SearchPartResult) werden innerhalb der LeftColumn bzw. RightColumn einer TwoColumnRow angezeigt, und können deshalb jeweils nur halb so breit dargestellt werden, wie der Header. Will man ein Teilsuchergebnis über die gesamte verfügbare Breite darstellen, so kann man selbstverständlich auch einfach das SearchPartResult-Element in eine ColumnRow schachteln, anstatt in eine TwoColumnRow.

Das zuvor beschriebene Layout-Modell wird auf der Suchergebnisseite dann folgendermaßen dargestellt (ohne die roten Rahmen versteht sich):



Suchergebnisse
Suche nach „test“

Semantische Suche

- Basisnetz
- Test
- test
- test

Meinten Sie...?

- Test2
- Test 3
- Gast
- test.bt
- test.bt

Dokumentensuche

- test.bt
- bla test.bt
- ampersand-test.bt
- test.bt
- PLSQL Packages and Types Reference.pdf

ColumnRow

LeftColumn

RightColumn

TwoColumnRow

5.1.7 Treffererklärungen

Das KP kann die Gründe für den Treffer anzeigen. Ein Klick auf die Trefferqualität öffnet das Erklären-Popup:

Der Markt Santa Catarina (29%) ✕

Semantisch gefunden über: 💡

- Barcelona (28,875%)
- Spanien (28,875%)
- Essen + Trinken in Spanien (28,875%)
- Essen + Trinken in Spanien (28,875%)

Folgende Gründe können dabei angezeigt werden:

- Direkte Treffer: Der Treffer wurde direkt über ein Attribut gefunden
- Semantisch: Ein oder mehrere Topics haben zu dem Treffer geführt

Über das Lampenicon können semantischen Gründe auch im Net-Navigator geöffnet werden.



5.1.7.1 Konfiguration der Treffererklärungen

Bei Zusammenstellbaren Suchen kann über CustomCauses bestimmt werden was als Erklärung angezeigt werden soll. Es werden alle CustomCauses beachtet die eine Eigenschaft *kpExplain* mit dem Wert *true* haben. Folgende Eigenschaften eines CustomCause werden ausgewertet:

Name	Typ	Beschreibung
nnExplain	Boolean	Ist der Wert true werden die matchedTopics mit im Net-Navigator angezeigt
kpExplain	Boolean	Ist der Wert true wird der CustomCause mit im Erklärungsfenster angezeigt. Ansonsten wird der CustomCause ignoriert.
labelKey	Zeichenkette	Ein Message-Key der zur Gruppierung der CustomCauses genutzt wird. Dabei werden alle CustomCauses mit gleichem Label zusammen gruppiert.
quality	Flieskomma	Die Qualität für den CustomCause
matchedTerms	Menge von Zeichenketten	Eine Liste von Zeichenketten die zum Treffer geführt haben.
matchedTopics	Menge von Topics	Eine Liste von Topics die zum Treffer geführt haben

CustomCauses können z.B. über einen Skript-Block in der Zusammenstellbaren Suche gesetzt werden:

```
<Path path="var(myHits)">
  <Each>
    <CustomCause modifyExisting="false">
      <SetProperty propertyName="matchedTerms" value="var(myMatchedTerms)"/>
      <SetProperty propertyName="matchedTopics" value="var(myMatchedTopics)"/>
      <SetProperty propertyName="kpExplain" value="true"/>
      <SetProperty propertyName="nnExplain" value="true"/>
      <SetProperty propertyName="quality" value="0.9"/>
    </CustomCause>
  </Each>
</Path>
```



```
</CustomCause>  
</Each>  
</Path>
```

5.2 Suchkonfiguration in älteren Versionen

5.2.1 Einführung

Das Knowledge-Portal Version 3.4 stellt einen Zwischenschritt in der Entwicklung der Suchkonfiguration dar und wird daher in dieser Dokumentation nicht näher betrachtet. Bitte aktualisieren Sie das Knowledge-Portal auf eine neuere Version (3.5 oder höher), sofern Sie derzeit noch die Version 3.4 einsetzen.

In den Knowledge-Portal-Versionen bis Version 3.3 wird prinzipiell zwischen vier verschiedenen Suchen unterschieden:

1. Standard-Begriffssuche
2. Meinten-Sie-Suche
3. Dokumenten-Volltextsuche
4. Ähnliche-Dokumente-Suche

Wenn ein Benutzer einen Suchterm im Web-Frontend in das Eingabefeld im Fenster "Suche" eingibt und den Button "Starten" klickt, werden die Suchen 1 bis 3 ausgeführt und das Ergebnis zusammengefasst auf einer Ergebnisseite dargestellt. Die Ergebnisse von Suche 1 erscheinen dabei normalerweise unter der Überschrift "Semantische Treffer", die Ergebnisse von Suche 2 unter der Überschrift "Meinten Sie..." und die Ergebnisse von Suche 3 unter "Dokument-Treffer" (vgl. folgende Abbildung).

Für die Suche 1 kann eine Reihe von Suchkonfigurationen zur Verfügung gestellt werden, die der Benutzer dann in einem Dropdown-Feld unter dem Sucheingabefeld auswählen kann.

The screenshot shows the search interface with the following elements:

- Search Bar:** Contains the text "milch" and a "Starten" button. Below it is a dropdown menu set to "Suche nach Mitarbeiter".
- Sort Options:** "Dokumente sortieren nach:" with radio buttons for "Qualität" (selected) and "Datum".
- Filter Options:** Checkboxes for "Office Dokumente", "Adobe Acrobat PDF", and "TXT und HTML-Dateien".
- Results Section:**
 - Semantische Treffer:** Lists "Paula Projekt" and "Werner Wohlgeruch" with star ratings.
 - Meinten Sie...:** Lists "Milch" with a blue dot.
 - Dokument-Treffer:** Lists PDF files with their titles, descriptions, sizes, and dates.

Die Suchen 3 oder 4 werden automatisch ausgeführt, wenn eine Detailseite für einen Begriff geöffnet wird, der als Brainbot-Themen-Begriff ("Container") im Wissensnetz konfiguriert ist. Die Ergebnisse der Suchen werden dann unter der Überschrift "Dokumentsuche X" bzw. "Ähnliche Dokumente" angezeigt (Näheres dazu im Kapitel 7.4).

Die Verknüpfung der unterschiedlichen Suchen im Knowledge-Portal mit bestimmten Suchkon-

figurationen in K-Infinity wird in den folgenden Kapiteln erläutert.

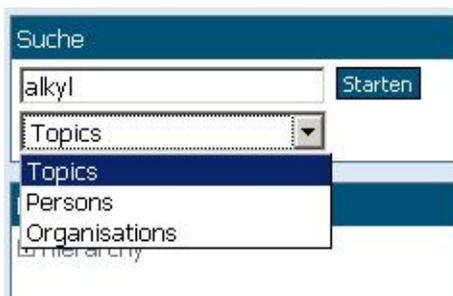
5.2.2 Konfiguration der Standard-Begriffssuche

Die Suchfunktion im Knowledge-Portal setzt voraus, dass im Wissensnetz mindestens eine Suchkonfiguration mit einer K-Infinity-Textsuche definiert und auch als KEM-Suche exportiert ist, die als Standard-Begriffssuche verwendet werden kann.

Damit die Suchkonfiguration vom Knowledge-Portal als Textsuche gefunden und verwendet werden kann, muss der Name der Suchkonfiguration mit dem Präfix *textSearch.* beginnen.

Dieses Präfix ist in der Konfigurationsdatei *names.xml* der Web-Anwendung unter der Eigenschaft *kpConfig.search.textSearchPrefix* festgelegt. Um dieses Präfix zu ändern, überschreiben Sie es in der Datei *customized.xml*.

Falls mehrere Suchkonfigurationen mit diesem Präfix im Wissensnetz gefunden werden, so werden sie in einem Dropdown-Feld unterhalb des Sucheingabefelds im Web-Frontend zur Auswahl angeboten. Wird nur eine Suchkonfiguration mit dem Präfix *textSearch.* gefunden, so wird das Dropdown-Feld nicht angezeigt und immer die gleiche Suchkonfiguration verwendet.



Dropdown-Feld zur Auswahl der Suchkonfiguration

Falls nichts anderes konfiguriert wird, ergibt sich das Label der Suchkonfiguration, das in der Dropdown-Liste angezeigt wird, aus dem Namen der Suchkonfiguration im Wissensnetz ab dem Ende des Präfixes.

Für eine Suchkonfiguration mit dem Namen *textSearch.Persons* wird in der Dropdown-Liste *Persons* angezeigt. Dieser Name wird sowohl im englisch- als auch im deutschsprachigen Modus angezeigt. Um völlig unabhängig vom Namen der Suchkonfiguration im Wissensnetz ein deutsch- und englischsprachiges Label für die Suchkonfiguration festzulegen, kann am Konfigurationsobjekt *kpConfig > messages* im Attribut *msg* bzw. *msg_de* ein Mapping zwischen dem Namen der Suchkonfiguration im Wissensnetz und dem gewünschten Label für die jeweilige Frontend-Sprache eingetragen werden. Um beispielsweise den deutschen Namen der Suche *textSearch.persons* zu definieren, muss in das Attribut *msg_de* das folgende Mapping eingetragen werden:

```
textSearch.Persons=Personen
```

... wobei natürlich *textSearch.Persons* durch den Namen der jeweiligen Suchkonfiguration im Wissensnetz und *Personen* durch den jeweiligen Namen für die Suche im Knowledge-Portal ersetzt werden muss. Analog dazu kann im Attribut *msg* das Label der Suche für den englischsprachigen Modus des Web-Frontends festgelegt werden.

Gruppieren von Suchergebnissen

Die Textsuche bietet die Möglichkeit, die Suchergebnisse nach Layoutkategorie zu grup-

pieren. Dies kann für jede Suche einzeln definiert werden, da eine Gruppierung nicht immer sinnvoll ist.

Hilfreich ist die Gruppierung bei Suchen, die als Ergebnisse Objekte verschiedener Art liefern. Bei der Suche nach Personen, d.h. wenn die Ergebnismenge ohnehin schon auf Personen beschränkt ist, ist die Gruppierfunktion hingegen kaum sinnvoll.

Um das Gruppieren für Suchen einzuschalten, muss das boolesche Attribut `"groupResults"` am Konfigurationsbegriff `"kpconfig > search > {Name der Suche}"` auf `"true"` gesetzt werden. Der Name der Suche ist dabei **ohne** Präfix (default: `"textSearch."`) anzugeben.

Der Eintrag für die Gruppierung der Suche `"textSearch.all"` würde also wie folgt aussehen:



Hinweis: Die Gruppierfunktion verwendet zur Visualisierung die Übersetzungen der Layout-Kategorien. Sollte die Gruppierfunktion verwendet werden, müssen diese ebenfalls konfiguriert werden, wie im Abschnitt `"Namen der Layout-Kategorien"` beschrieben.

5.2.3 Konfiguration der Meinten-Sie-Suche

Parallel zur Ausführung der Standard-Begriffssuche kann bei der Suche im Knowledge-Portal auch immer eine sog. `"Meinten-Sie-Suche"` mitausgeführt werden. Diese Suche wird normalerweise auf eine K-Infinity-Suchkonfiguration mit einer Trigrammsuche abgebildet und findet so Begriffe, die aufgrund von Schreibfehlern o.ä. nicht durch die Standard-Textsuche gefunden werden können.

Die Treffer der `"Meinten-Sie-Suche"` werden auf der Suchergebnisseite unter der Überschrift `"Meinten Sie..."` angezeigt; allerdings nur solche Begriffe, die nicht schon von der Standard-Textsuche (unter `"Semantische Treffer"`) gefunden wurden. Das K-Infinity Knowledge-Portal verwendet eine Suche mit dem Namen `trigramSearch`, die mit dem Knowledge-Builder als KEM-Suche verfügbar gemacht werden muss.

Soll der Name der Suche geändert werden, die bei der Ausführung der `"Meinten-Sie-Suche"` verwendet wird, kann dies in der Konfiguration am Begriff `"kpConfig > search"` unter dem Textattribut `"trigramSearch"` festgelegt werden.

5.2.4 Konfiguration der Dokumenten-Volltextsuche

In das Knowledge-Portal kann eine Dokumenten-Volltextsuche eingebunden werden. Diese durchsucht zusätzlich zu den Inhalten im Wissensnetz auch Dokumente nach den entsprechenden Suchbegriffen.

Für diese Funktionalität wird eine Installation der NextBot-Suchmaschine benötigt, welche separate Lizenzen erfordert.

Der Name der Suchkonfiguration, die für die Dokumenten-Volltextsuche im Knowledge-Portal verwendet wird, ist in der Konfiguration am Begriff `"kpConfig > search"` unter dem Textattribut `"fulltextSearch"` zu konfigurieren.

Um eine alternative Volltextsuche anzugeben, überschreiben Sie die Einstellung. Als Wert



muss hier der Name der alternativen Volltextsuche angegeben werden.

Hinweis: Erscheint im Knowledge-Portal die Fehlermeldung "*Konfiguration unvollständig*" ist die Bridge nicht in der Lage, mit der NextBot-Engine zu kommunizieren. Überprüfen sie dann die NextBot-Konfiguration im Admintool.

5.2.4.1 Dateiendungsfilter

Der Dateiendungsfilter ermöglicht die Einschränkung der Suchergebnisse auf bestimmte Dateitypen. **Dieser Filter funktioniert nur in Verbindung mit der BrainBot-Suchengine.**

Der Dateiendungsfilter ist in der Standardeinstellung vom Knowledge-Portal nicht aktiviert. Soll der Filter aktiviert werden, so müssen mit dem Knowledge-Builder die folgenden Einstellungen in der Knowledge-Portal-Konfiguration im Wissensnetz vorgenommen werden

- Legen Sie am Begriff "KP-Konfiguration" den Unterbegriff "applicationComponents" an
- Erzeugen Sie in der Schemadefinition für Unterbegriffe von applicationComponents das Attribut "class" (Typ: Zeichenkette) an
- Legen Sie am Begriff "applicationComponents" den Unterbegriff "search" an
- Legen Sie in der Schemadefinition für Unterbegriffe von search das Attribut "extensionFilters" an
- Legen Sie am Begriff "search" das Attribut "extensionFilters" an und Bearbeiten Sie das Attribut mit dem

Konfiguration

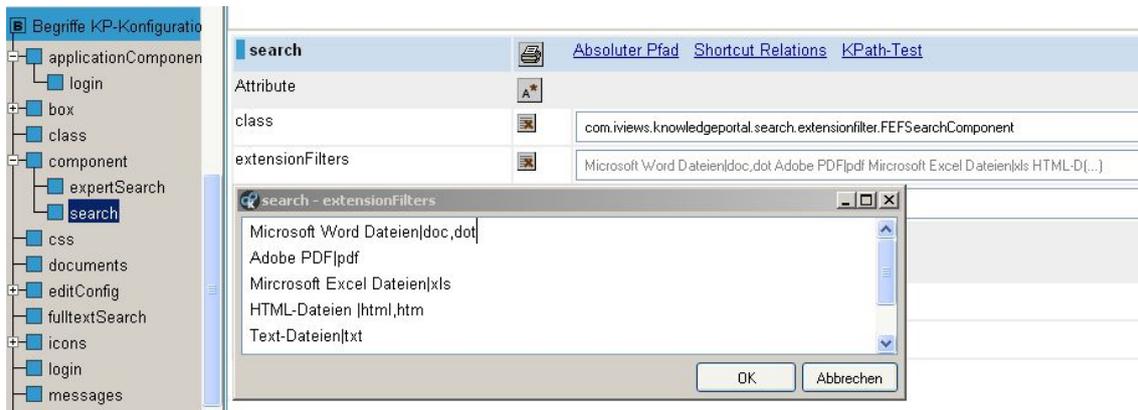
Der Dateiendungsfilter selbst benötigt eine zusätzliche Konfiguration. Die Parameter werden ebenfalls am Begriff "*kpConfig > applicationComponents > search*" definiert. Der Textattributname muss dabei "*extensionFilters*" lauten.

Als Wert der Eigenschaft kann eine Liste von Filtern definiert werden, die im Suchformular als Checkboxen angezeigt werden. Der erste Teil ist der angezeigte String im Auswahlformular. Durch "|" getrennt folgen dann alle Dateiendungen, die diesem Dateityp entsprechen. Die Dateiendungen selbst werden durch "," getrennt. Nach einem Zeilenumbruch folgt eine neue Dateiart.

Ein Konfiguration könnte beispielsweise wie folgt aussehen:

```
Microsoft Word Dokumente|doc, dot  
Text-Dateien|txt  
Source-Code|java, c, cc, h  
HTML-Dateien|html, htm  
Mails|mail
```

Hinweis: Beim zu setzenden Wert des Schlüssels handelt es sich um einen mehrzeiligen Text, weshalb empfohlen wird, die Konfiguration im Netz vorzunehmen (siehe dazu "Konfiguration im Netz"). In diesem Fall kann ein mehrzeiliges Textfeld verwendet werden. Sollte die Konfiguration über die Datei "*customized.xml*" erfolgen, müssen die Zeilenumbrüche durch "\n" angegeben werden!



Konfiguration für Dateieindungsfilter im Wissensnetz

5.2.4.2 Datumsfilter

Die Datumsfilterkomponente ermöglicht die Filterung der Volltextsuchergebnisse. Dabei kann ein Zeitraum angegeben werden, in dem das letzte Änderungsdatum der Datei bzw. Empfangsdatum der E-Mail liegen muss, um in die Ergebnismenge zu kommen.

Wird ein Feld leer gelassen, geht das Intervall endlos in die Vergangenheit bzw. bis zum aktuellen Zeitpunkt.

Aktivierung

Gegenwärtig wird der Datumsfilter mit dem Dateieindungsfilter aktiviert.

Konfiguration

Der Datumsfilter beschränkt sich auf die Felder "von" und "bis". Deshalb ist hier keine weitere Konfiguration nötig.

5.2.4.3 Download von Dokumenten

Beim Anklicken eines Dokument-Treffers auf der Suchergebnisseite wird das betreffende Dokument im Knowledge-Portal geöffnet. Hierfür stehen zwei unterschiedliche Modi zur Verfügung:

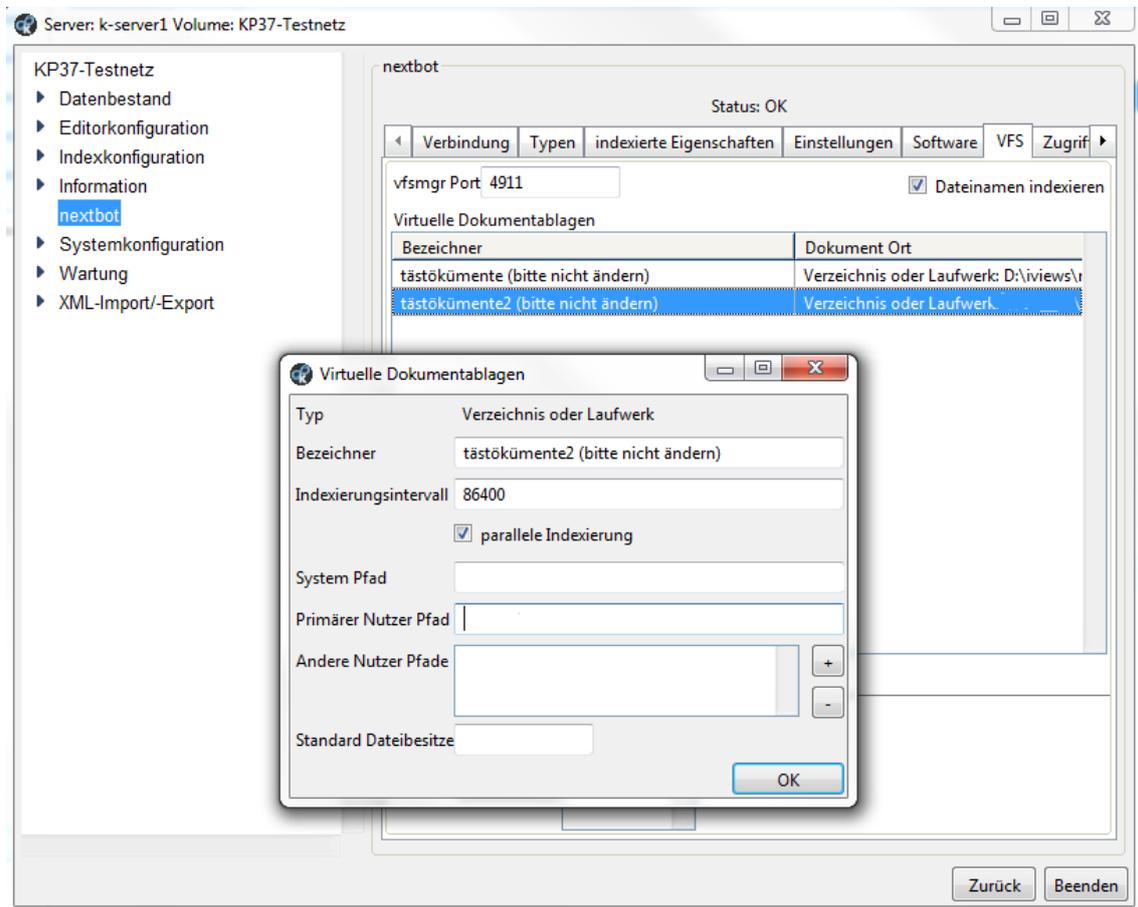
- Öffnen einer Kopie des Dokuments
- Öffnen des Dokuments an seinem Original-Speicherort im Netzwerk

Standardmäßig wird beim Anklicken des Dateinamens auf der Suchergebnisseite eine Kopie der Datei heruntergeladen und im Browser geöffnet. Dies hat den Vorteil, dass die Dokumente unabhängig vom tatsächlichen Speicherort immer im Knowledge-Portal geöffnet werden können. Da das Knowledge-Portal die Binärdaten des Dokuments aber beim NextBot-Dienst anfordern muss (der generell den Zugriff auf alle Dokumente hat), kann es abhängig von der Last auf dem NextBot-Dienst zu Verzögerungen beim Download kommen, bspw. wenn der NextBot-Dienst aufgrund eines Crawling-Vorgangs stark belastet ist.

Alternativ zum Download von Dokument-Kopien über NextBot können die Dokumente auch direkt über einen File-Link geöffnet werden. Hierfür ist es erforderlich, dass alle von Nextbot indexierten File-Shares über einen eindeutigen Pfad im Netzwerk (UNC-Pfad) angesprochen werden können. Die Zuordnung von File-Share zu dem UNC-Pfad erfolgt im K-Infinity Admin-Tool (Nextbot > Reiter "VFS" > Doppelklick auf Dokumentablage > Feld "Primärer Nutzer



Pfad" - siehe Bild unten). Um den Download über File-Links zu aktivieren, muss in der KP-Konfiguration im Wissensnetz am Unterbegriff "documents" ein boolesches Attribut mit dem Namen "useUncFileLink" angelegt und angehakt werden. Damit die neue Konfiguration aktiv wird, muss die KP-Konfiguration neu geladen oder ein Neustart des KP durchgeführt werden.



File-Links zu Network Ressourcen (UNC Links) können nicht von allen Browsern direkt geklickt werden. Hier ist eine Zusammenfassung von den Browsern und ihre Unterstützung.

HTML -> UNC Link: heißt, das der Link befindet sich auf einer HTML Seite.

New Tab/Window -> UNC Link: heißt, der Link wird in einem neuen Tab oder Window manuell eingegeben.

Z.B. UNC Link: file://myserver/somepath/filename.pdf

Browser	HTML -> UNC Link	New Tab/Window -> UNC Link
Internet Explorer	ja	ja
Firefox 3.5	nein	ja, muss aber mit "file:///" anfangen
Chrome 3.0	nein	ja
Safari 4.0	nein	nein



Für Firefox gibt es die folgende Möglichkeiten UNC Links zu unterstützen:

- Installiere die Extension LocalLink (<https://addons.mozilla.org/firefox/addon/281>). Dann Rechtsklick auf den Link und "Open Link in Local Context" wählen.
- Install die Extension NoScript (<https://addons.mozilla.org/firefox/addon/722>). Dann "allow scripts globally" und "Advanced -> Trusted -> "Allow local links". Schließlich die gewünschten Seiten als "Trusted" setzen.
- Mehr Info: http://kb.mozillazine.org/Links_to_local_pages_do_not_work

5.2.5 Gruppierende Suchen

Das Knowledge-Portal verwendet als Grundeinstellung bei der Sortierung von Suchergebnissen das globale Ranking des Treffers, unabhängig vom Typ des Wissensnetzknотens in der Trefferliste.

Für einzelne Suchen kann aber auch die Gruppierung der Suchergebnisse nach einer Kategorie konfiguriert werden. In diesem Fall werden die Ergebnisse, die zur gleichen Kategorie gehören, zusammen angezeigt.

Um dies für eine Suche zu konfigurieren, muss ein Unterbegriff von "search" erstellt werden, dessen Name dem Namen der Textsuche (z.B. "textSearch.Alles") entspricht, jedoch **ohne** das führende "textSearch." also "Alles." .Wird bei diesem Begriff das boolesche Attribut "groupResults" gesetzt, wird die Suchergebnisliste der konfigurierten Suche nach einem Reload der Konfiguration gruppiert angezeigt.



Unterbegriff Alles

Alles	
Attribute	
groupResults	<input checked="" type="checkbox"/>
Name (Deutsch)	German <input type="checkbox"/> univ. <input type="text" value="Alles"/>
Relationen	

Attributkonfiguration für den Begriff Alles

6 Layout-Konfiguration



6.1 Texte konfigurieren

Alle Texte die im Knowledge-Portal angezeigt werden können durch eigene Texte überschrieben werden. Die Konfiguration ist so aufgebaut, dass es zu jedem angezeigten Text im Portal einen Konfigurationsschlüssel gibt. Je nach Sprache wird zu diesem Konfigurationsschlüssel ein Text hinterlegt.

Ohne eine individuelle Anpassung werden die Defaultwerte verwendet, die in der Datei `msg.properties` abgelegt sind. Den Inhalt können Sie im Kapitel "msg.properties" einsehen. Sie haben die Möglichkeit die bestehenden Defaulteinstellungen zu überschreiben oder eigene Konfigurationsschlüssel zu definieren.

Eigene Konfigurationsschlüssel benötigen Sie zum Übersetzen von Layout-Kategorien oder anzuzeigenden Texten in View- und EditConfigs. Mehr dazu erfahren Sie in den entsprechenden Kapiteln.

Die Konfiguration der Texte kann in mehreren Sprachen erfolgen. Für jede Sprache gibt es eine eigene Übersetzungsdatei. Wird der Konfigurationsschlüssel in der aktiven Sprache nicht gefunden, wird auf die Fallsprache Englisch ausgewichen.

Ist der Konfigurationsschlüssel in Englisch ebenfalls nicht vorhanden, wird ein entsprechender Platzhalter angezeigt. Dieser entspricht dem Konfigurationsschlüssel mit vorangestelltem und folgendem "???".

Wäre z.B. der Konfigurationsschlüssel `viewConfig.tabname.tab1` weder in Deutsch noch in Englisch konfiguriert, würde der Text `???viewConfig.tabname.tab1???` angezeigt werden. Angezeigte Zeichenketten die von "???" umgeben sind lassen immer darauf schliessen, dass kein entsprechender Anzeigetext konfiguriert ist.

Die Einstellungen werden in der KP-Konfiguration vorgenommen. Wechseln Sie dazu zum Begriff: `kpConfig > messages`

An diesem Begriff kann für jede Sprache ein Attribut angelegt werden, das die Konfigurationsschlüssel und zugehörigen Texte enthält. Der Name des Attributes folgen dem Schema `msg_{languageCode}`. Eine Ausnahme bildet die Sprache Englisch. Diese hat keinen languageCode als Parameter.

Beispiele:

Sprache	Attributname
Englisch	msg
Deutsch	msg_de
Spanisch	msg_es
...	

In das entsprechende Attribut werden jetzt die jeweiligen Konfigurationsschlüssel und Text eingetragen. Schlüssel und Text sind jeweils durch ein "=" getrennt. Jeder Konfigurationsschlüssel beginnt in einer neuen Zeile.

Schema:



```
mein.configkey.for.messages = Mein Text für den angegebenen Schlüssel
```

Einfache Hochkommazeichen (') die man auf der deutschen Tastatur mittels SHIFT+# eingeben kann, werden in bestimmten Fällen aus den Texten herausgefiltert und sollten daher nicht verwendet werden. Statt des einfachen Hochkommazugs sollte man in den Texten die typographischen Anführungs- und Apostroph-Zeichen verwenden, die bspw. bei der Eingabe von Texten in Word automatisch erzeugt werden (z.B. "j'ai" anstatt "j'ai").

Beispiel zum Überschreiben von Texten:

Wenn man den Titel der Navigationsbox ändern möchte, muss zuerst der entsprechenden Schlüssel ermittelt werden. Dieser kann in der Datei `msg.properties` (Kapitel 10.3) eingesehen werden. Sort suchen Sie nach *Navigation* und finden den Schlüssel `box.hierarchies.title`

Um im Titel der Navigations-Box "Hauptmenü" anzuzeigen, tragen sie in das Attribut `msg_de` folgende Zeile ein:

```
box.hierarchies.title = Hauptmenü
```

Beispiel zum Neuanlegen von Texten:

Bei der Konfiguration des Knowledge-Portals werden Sie auch Texte verwenden, die nicht vorkonfiguriert sind. Z.B. beim anlegen neuer Layout-Kategorien oder in `ViewConfigs`. Diese zusätzlichen Übersetzungen werden auf die gleiche Art und Weise konfiguriert.

Wenn Sie in einer `ViewConfig` einen anderen Attributnamen anzeigen möchten würden Sie folgendes in die `ViewConfig` schreiben.

```
<Relation>  
  <LabelKey>viewConfig.person.worksAt</LabelKey>  
  <InternalName>worksAt</InternalName>  
</Relation>
```

Um auf der Detailseite diesen Text anzuzeigen:

```
Firma:          ACME Inc.
```

tragen Sie den in der `ViewConfig` angegebenen Konfigurationsschlüssel in die `msg_de` ein:

```
viewConfig.person.worksAt = Firma:
```

HINWEIS:

Die Änderungen in den Übersetzungen wirken sich erst nach einem Neustart des Tomcat-Server oder einem Reload der Webapplikation aus. Ein Reload der `VolumeConfiguration` ist **nicht** ausreichend.



6.2 Layout-Kategorien

Das Basisnetz definiert am Wurzelbegriff ein Attribut für seine Unterbegriffe mit dem Namen "*layoutCategory*".

Wird ein Begriff mit diesem Attribut ausgestattet, so kann die folgende Menge an Einstellungen konfiguriert werden, die die Anzeige dieses Begriffs, seiner Unterbegriffe und seiner Individuen steuert.

- Verwendung als Gruppierungselement bei einer Gruppierenden Suche
- Anzeige des Icons bei den Suchergebnissen
- Anzeige des Icons auf der Detailseite des Begriffs
- detaillierte Konfiguration, welche Attribute und Relationen in welcher Reihenfolge auf der Detailseite eines Knotens angezeigt werden

Die Namen von Layout-Kategorien dürfen keine Umlaute, Leerzeichen und Sonderzeichen (Interpunktionszeichen etc.) enthalten.

6.3 Anzeigenamen der Layoutkategorien

Der Name der "*layoutCategory*" wird bei der Darstellung einer gruppierten Suche verwendet. Daher reicht es nicht aus, im *kpConfig*-Netz nur den internen Namen zu verwenden, sobald eine gruppierte Suche definiert ist. Aktuell müssen die landesspezifischen Übersetzungen noch in einer *properties*-Datei konfiguriert werden.

Da Layoutkategorien kundenspezifisch sind, werden diese Namen im Netz konfiguriert (siehe Kapitel "*Texte konfigurieren*").

Der Eintrag folgt folgendem Schema:

```
layout.category.{Name der Layoutkategorie}.name
```

= Name in der entsprechenden Sprache

Der Eintrag für die Layoutkategorie "*person*" für die deutsche Sprache würde also folgendermaßen aussehen:

```
layout.category.person.name = Person
```

Sollte eine Layoutkategorie nicht konfiguriert sein, wird bei Bedarf ein Platzhalter in folgendem Format ausgegeben:

```
??? layout.category.{kategorienname}.name ???
```

Begriffe ohne Layoutkategorie

Existieren Begriffe, die keiner Layoutkategorie zugeordnet sind, wird bei Bedarf folgende Übersetzung ausgegeben:

```
layout.category.default.name
```

Es sollte also darauf geachtet werden, dass dieser Wert in jedem Fall gesetzt ist.



6.4 Darstellung von Boxen

Alle Boxen im Knowledge-Portal können vom Benutzer minimiert oder maximiert werden, um die Inhalte aus- bzw. einzublenden.

Die Anwendung speichert den Zustand der Fenster pro Benutzer und Sitzung und wird bei einer neuen Session wieder in der konfigurierbaren Grundeinstellung angezeigt.

Bei minimierten Boxen, die eine Liste von Knoten des Wissensnetzes zeigen, wird neben dem Namen der Box im Fenstertitel noch die Anzahl der Knoten angezeigt, damit der Benutzer einen Hinweis darauf erhält, ob es sich lohnt oder sinnvoll ist, die jeweilige Box aufzuklappen.

Voreinstellung ist, dass alle Boxen initial (d.h. beim Öffnen der Anwendung in einer neuen Benutzersitzung) maximiert dargestellt werden.

Die Einstellung für jede einzelne Box kann am entsprechenden Unterbegriff von "box" vorgenommen werden. Alle diese Begriffe erben das boolesche Attribut "defaultCollapse", mit dem gesteuert wird, ob die Box initial geschlossen ist.

box		
Attribute		
defaultCollapse		<input checked="" type="checkbox"/>
explainable		<input type="checkbox"/>
Name (Deutsch)		German <input type="checkbox"/> univ. <input type="text" value="box"/>
Relationen		
Eigenschaften		
Begriff kann Individuen haben		<input type="checkbox"/>
Begriff kann Individuen erweitern		<input type="checkbox"/>

Attributkonfiguration für box

Jeder einzelne Unterbegriff von "box" verfügt noch über eigene Attribute, über die ein an die Funktionalität angepasstes Verhalten konfiguriert werden kann.



Unterstruktur für box

Das Basisnetz enthält nur die Boxen, die modellunabhängig vorhanden sind. Sollen auch einzelne Kontextboxen vom Standardverhalten abweichen, so muss für jede Kontextbox ein eigener Unterbegriff des Begriffs "*context*" erstellt werden.

Der Name des Begriffs muss dabei identisch mit der "*externalld*" der Suche sein, die in der Box ausgeführt wird. Dies darf nicht mit der "*externalld*" des Ordners verwechselt werden, in dem die Suche enthalten ist. An diesem Unterbegriff kann dann das "*defaultCollapse*"-Attribut gesetzt werden, wenn die Box initial geschlossen sein soll.

6.4.1 Navigation

6.4.1.1 Hierarchie

Die Navigation im herkömmlichen Sinne wird im Knowledge-Portal über den Hierarchieordner realisiert. Beim Start der Applikation wird aus dem Netz der Ordner mit der externen ID "*hierarchies*" gelesen. Aus dessen Inhalt wird eine baumförmige Struktur erzeugt, die im Knowledge-Portal zur Navigation dient.

Zur Befüllung der Navigation bestehen verschiedene Möglichkeiten:

Ordner

Alle Unterordner des Ordners "*hierarchies*" werden in der Navigation als Knoten angezeigt. Sie selbst sind nicht klickbar, da sie nicht durch ein Topic im Netz repräsentiert werden. Diese Unterordner dienen zur Gruppierung von bestimmten Topics, die manuell ausgewählt werden. Dazu müssen im Knowledge-Builder alle Begriffe und Individuen, die Bestandteil der Navigation werden sollen, in den gewünschten Ordner gezogen werden. Im Knowledge-Portal werden alle diese Objekte als Unterpunkte des Ordners angezeigt. Die Ordner können selbst wieder Unterordner und Expertensuchen enthalten.

Hierarchieordner

Hierarchieordner sind Ordner, deren Inhalt automatisch generiert wird. Dazu wird ein Ausgangsbegriff gewählt. Von diesem ausgehend werden Relationen definiert, die auf die jeweils darunterliegende Hierarchieebene verweisen.

Ausgehend vom Begriff "*Organisation*" und der ersten Relation "*hat Individuum*" würden in der Hierarchie unter dem Knoten "*Organisation*" alle im Netz gespeicherten Firmen erscheinen. Definiert man die zweite Relation als "*beschäftigt Person*", würden als Unterobjekte der Firmen alle Mitarbeiter der jeweiligen Firma erscheinen.

Mehr zu Hierarchieordnern finden Sie in der Dokumentation des Knowledge-Builders.

Expertensuchen

Wie bei den Hierarchieordnern wird auch der Inhalt von Suchordnern dynamisch generiert. Die Suche wird im Hierarchiebaum als nicht klickbarer Knoten angezeigt. Klappt man den Knoten auf, werden alle Unterpunkte sichtbar. Diese ergeben sich aus der Ergebnismenge der definierten Expertensuche. Die Übergabe eines Parameters für diese Suchen ist dabei



nicht möglich.

Bitte beachten: Der Inhalt des Navigationsordners wird nicht laufend sondern nur beim Start einer Session aktualisiert.

6.4.1.2 KScript-Hierarchie

KScript Navigation

Alternativ zur 'normalen' Hierarchie-Komponente gibt es die Komponente KScriptXmlTreeComponent.

Der Vorteil gegenüber der alten Hierarchiekomponenten ist, dass das gerade im Knowledge-Portal angezeigte Topic im Baum als aktiv gekennzeichnet wird. Außerdem ist die Komponente wegen der erweiterbaren KPath-Ausdrücke flexibler. Wird ein im Baum angezeigtes Topic über eine Pflegeseite geändert, werden diese Änderungen auch im Baum übernommen.

Der Nachteil liegt in der erhöhten Komplexität der Konfiguration.

Die Komponente ist ab Version 3.6 Teil des Knowledge-Portals.

Attribute des Elements <KScriptXMLTree>

Attribut	Beschreibung
kpathRootNodes	<p>Ein KPath-Ausdruck, der die Topics der obersten Ebene liefert. Die Ergebnisse werden als Wurzelknoten in der Navigation angezeigt. Das Ergebnis ist nicht auf Begriffe festgelegt, es können also auch Individuen als Ausgangselemente verwendet werden. Die Topics müssen jedoch mit dem Ausdruck 'kpathChildren' kompatibel sein.</p> <p>In dem Ausdruck sollte kein Ausgangselement erwartet werden, d.h. der Ausdruck sollte explizit den Wurzelbegriff voranstellen.</p> <p>Beispiel: <code>//\$organisationUnit\$/instances()</code></p>
kpathChildren	<p>Dieser Ausdruck berechnet die Unterelemente eines Knotens. Als Ausgangspunkt dieses Ausdrucks wird das übergeordnete Element im Baum verwendet.</p> <p>Das Ergebnis dieses Ausdrucks sollten ein oder mehrere Topics sein. Falls diese weitere Unterelemente haben, wird der Ausdruck von "kpathChildren" wiederum auf den im ersten Durchgang gefundenen Elementen aufgerufen.</p> <p>Beispiel: <code>~\$hasSubUnit\$/target()</code></p>



kpathParents	<p>Der Ausdruck, der die übergeordneten Knoten im Baum für eine Topic berechnet. Dieser Ausdruck wird verwendet, um das gerade angezeigte Topic in die Baumstruktur einzuordnen.</p> <p>Der Ausgangspunkt des Ausdrucks ein beliebiges Topic. Das Ergebnis sollte ein einzelnes Element sein.</p> <p>Beispiel: ~\$hasSuperUnit\$/target()</p>
--------------	---

Die Berechnung der Wurzelknoten erfolgt am Anfang der Session, Änderungen werden somit erst nach dem Neuladen der Konfiguration und einem erneuten Login übernommen.

Unterelemente werden erst beim Aufklappen eines Knotens berechnet.

Beispielkonfiguration

Das hier gezeigte Beispiel ist ein Ausschnitt aus der XML-Konfiguration, die ab Version 3.6 verfügbar ist.

```
<UserComponents>
  <!-- ..... -->
  <KScriptXMLTree kpathChildren=~$hasSubUnit$/target() "
    kpathParents=~$hasSuperUnit$/target() "
    kpathRootNodes="//$organisationUnit$/instances()">
  </KScriptXMLTree>
</UserComponents>
```

6.4.2 Verlauf

Im Fenster "*Verlauf*" werden die letzten vom Benutzer in der Detailansicht geöffneten Begriffe und Individuen aufgelistet. Die maximale Anzahl von Begriffen und Individuen, die von der Anwendung in der Verlaufsliste gespeichert wird, ist standardmäßig auf 10 Einträge beschränkt.

Über das Attribut (Attributtyp Zeichenkette) "*limit*", kann die Box entweder ausgeblendet ("*limit=0*") oder für die maximale Anzahl der Einträge ein anderer Wert festgelegt werden (z.B. "*limit=30*"). Der Konfigurationsbegriff für den Verlauf ist "*kpConfig > box > history*".



Verlaufs-Box

6.4.3 Schnellanlage

Über die Schnellanlage können direkt Individuen erzeugt werden, ohne dass der Benutzer vorher auf den dazugehörigen Begriff navigieren muss.

Es können zwei Schnellanlage-Boxen konfiguriert werden, die dann in der linken Spalte des KnowledgePortals angezeigt werden.

Im Wissensnetz müssen Ordner mit der externen ID "quickCreate" bzw. "quickCreate2" erzeugt werden. In diese Ordner werden dann die Begriffe gelegt, von denen Individuen erzeugt werden sollen. Es kann sich bei den Ordnern auch um Expertensuchen handeln, die die gewünschten Begriffe als Suchergebnis zurück liefern. Zur Zeit können nur Individuen, keine Unterbegriffe erzeugt werden.

Konfiguration der Boxen im Knowledge-Portal 3.x

Limit und Standardaufklappverhalten der Boxen lassen sich über die Attribute "limit" bzw. "defaultCollapse" konfigurieren. Diese Attribute müssen am Konfigurationsbegriff "box.kpathContext.quickCreate" bzw. "box.kpathContext.quickCreate2" angebracht werden.

Die Überschrift der Boxen wird über folgende Messages-Parameter bestimmt:

- folder.quickCreate=Überschrift für erste Box
- folder.quickCreate2=Überschrift für zweite Box

Konfiguration der Boxen im Knowledge-Portal 2.x

Limit und Standardaufklappverhalten der Boxen lassen sich über die Attribute "limit" bzw. "defaultCollapse" konfigurieren. Diese Attribute müssen am Konfigurationsbegriff "box.folder.quickCreate" bzw. "box.folder.quickCreate2" angebracht werden.

Die Überschrift der Boxen wird über folgende Messages-Parameter bestimmt:

- folder.quickCreate=Überschrift für erste Box
- folder.quickCreate2=Überschrift für zweite Box



The screenshot shows the KnowledgePortal interface. At the top, the title 'KnowledgePortal' is displayed. Below it is a login section with a dropdown menu labeled 'Benutzer'. Underneath are two input fields: 'Benutzerkennung' and 'Passwort', followed by an 'Anmelden' button. Below the login section is a navigation menu with several items: 'Suche', 'Schnellsuche', 'Navigation', 'Verlauf', 'Neu anlegen... (3)', and 'Neu anlegen 2... (3)'. The 'Neu anlegen 2... (3)' item is highlighted with a red border and contains a list of options: 'Anwendung', 'Gemüse', 'Person', 'Foo', 'Kunde', and '[Alles zeigen...]'. The 'Neu anlegen... (3)' item contains 'Anwendung', 'Gemüse', and 'Person'.

6.4.4 Schnellsuche

Über die Schnellsuch-Komponente kann eine Expertensuche ausgeführt werden, die als Eingabe mehrere Parameter erwartet. Im folgenden Beispiel werden Städte gesucht, die in bestimmten Bundesländern und Staaten liegen.



Stadt	Land	Staat	Länderkürzel
Wiesbaden	Hessen	Deutschland	de
Frankfurt am Main	Hessen	Deutschland	de
Darmstadt	Hessen	Deutschland	de
Kassel	Hessen	Deutschland	de

Schnellsuche: Eingabemaske und Suchergebnisse

Die Felder der Eingabemaske werden von den Parametern der Expertensuche abgeleitet. Die Darstellung der Suchergebnisse kann über einen Tabellenexport-Ordner definiert werden.

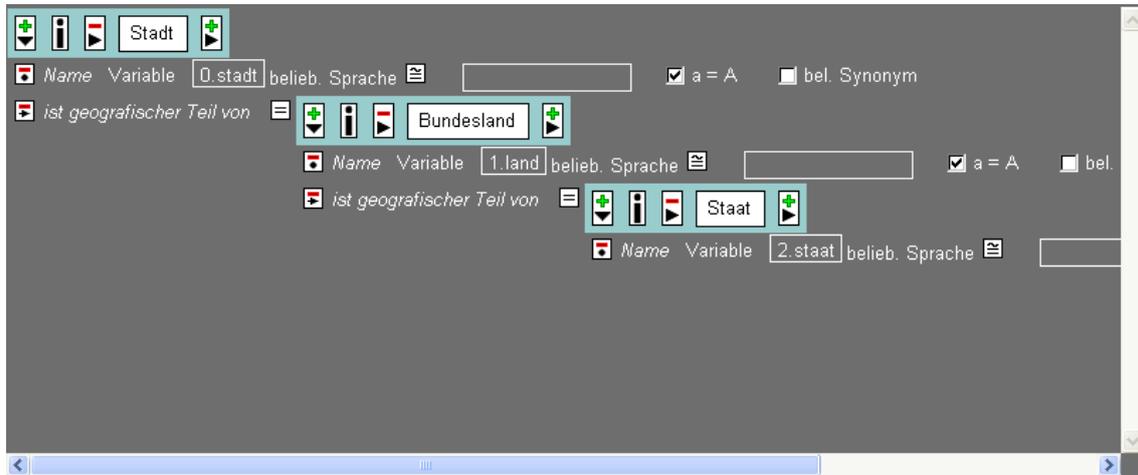
Die sich anschließende Abfolge von Schritten definiert und konfiguriert die Schnellsuch-Komponente:

1. Erstellen Sie den Begriff "*kpConfig > component > quickSearch*"
2. Erstellen Sie an diesem Begriff folgende Textattribute:
 - class
 - queryFolderEID
 - exportFolderEID
 - exportFolderMappingName
3. Setzen Sie die folgenden Attributwerte:
 - class = *com.iviews.knowledgeportal.quicksearch.QuickSearchComponent*
 - queryFolderEID = *quickSearch*
 - exportFolderEID = *quickSearchExport*
 - exportFolderMappingName = *quickSearch*

Das Attribut "*class*" definiert die Komponente. Wenn diese Zeile auskommentiert ist, wird keine Schnellsuch-Maske angezeigt.

Der Suchordner, der die auszuführende Expertensuche enthält, muss die externe ID haben, die im Attribut "*queryFolderEID*" festgelegt ist. Im Attribut "*exportFolderEID*" ist die externe ID des "*Ordnern für Tabellen-Exporte*" angegeben, der das Mapping ("*exportFolderMappingName*") enthält, das zur Formatierung der Suchergebnisse verwendet wird.

Das folgende Bild stellt die Expertensuche dar, von der das Suchformular im obigen Beispiel abgeleitet ist.



Expertensuche

Für jeden der drei Suchparameter wird in der Suchmaske ein Feld erzeugt. Die Reihenfolge ergibt sich aus der alphabetischen Sortierung der Parameternamen. In den Textattributen für Übersetzungen "*quickSearch_de*" und "*quickSearch*" am Konfigurationsbegriff "*kpConfig > messages*" können Übersetzungen und Aussehen des Formulars definiert werden:

Schlüssel	Beschreibung	Beispiel
<i>parameter.label</i>	Überschrift des Felds von <i>parameter</i>	1.land.label = Land
<i>parameter.title</i>	Tooltipp der Überschrift des Felds von <i>parameter</i>	1.land.title = Bundesland
<i>parameter.size</i>	Breite des Felds von <i>parameter</i>	1.land.size = 4
title	Überschrift der Ergebnistabelle	title = Schnellsuche

Da die Ergebnistabellen der Schnellsuche durchaus sehr lang werden können, gibt es eine Schnellfilter-Funktion, mit der die Zeilen der Tabelle durchsucht werden können. Dazu wird oberhalb der Ergebnistabelle ein Eingabefeld angezeigt. Wird eine Eingabe in diesem Feld vorgenommen, so werden nur noch solche Zeilen aus der Ergebnistabelle angezeigt, in denen der eingegebene Wert vorkommt. Der Titel des Eingabefelds in der Ergebnistabelle kann über die Eigenschaft "*filterLabel*" in den Textattributen für Übersetzungen "*quickSearch*" bzw. "*quickSearch_de*" (oder für andere Übersetzungen) eingetragen werden. Der Standardwert ist "*Filter results*" bzw. "*Ergebnisse filtern*".



6.5 Style-Anpassungen

Das Aussehen des Knowledge-Portal kann über Konfigurations-Optionen angepasst werden.

1. Die Farben und Schriftarten die in den Style-Sheets der Web-Applikation festgelegt sind, können verändert werden.
2. Das Header-Bild kann ausgetauscht werden (siehe Kapitel 7.6 "Konfiguration des Seitenkopfes")
3. Die Icons, die Begriffen zugeordnet werden und in Suchergebnissen oder auf der Detailseite angezeigt werden, können konfiguriert werden.

6.5.1 CSS-Anpassungen

In der Web-Applikation existieren drei CSS-Dateien im css-Verzeichnis. Die Style-Sheet-Definitionen in jeder Datei können über entsprechende Attribute in der Netz-Konfiguration überschrieben werden. Dazu muss am Konfigurationsbegriff ein Unterbegriff mit dem Namen **css**, sowie drei Stringattribute mit den Namen **all**, **ie**, **other** angelegt werden.

1. Die Datei **styles_all.cssj** enthält generelle Style-Sheet-Definitionen, die für alle Browser verwendet werden. Definitionen in dieser Datei werden von denen im Attribut **all** überschrieben.
2. Die Datei **styles_ie.cssj** enthält Style-Sheet-Definitionen, die nur für den Internet-Explorer verwendet werden. Definitionen in dieser Datei werden von denen im Attribut **ie** überschrieben
3. Die Datei **styles.cssj** enthält Style-Sheet-Definition, die für alle Browser außer dem Internet-Explorer verwendet werden. Definitionen in dieser Datei werden von denen im Attribut **other** überschrieben.

Beispielsweise existiert in der Datei **styles_all.cssj** die Definition:

```
body {  
  font-family: tahoma, verdana, arial, helvetica, sans-serif;  
  font-size: 10pt;  
  color: #101010;  
  background-color: #dbe9f3;  
  float: left;  
  height: 100%;  
  width: 100%;  
}
```

Um die Hintergrundfarbe des Knowledge-Portals anzupassen (z.B. auf die Farbe Weiß), kann im Attribut **all** die Definition:

```
body { background-color: #ffffff; }
```

hinzugefügt werden.

6.5.2 Eigene Dateien in das Webapps-Verzeichnis laden

Für manche Anpassungen sind Dateien (z.B. Bilder) nötig, die nicht im Knowledge-Portal enthalten sind. Beispiele dafür sind neue Begriff-Icons, angepasste Net-Navigator-Konfigurationen

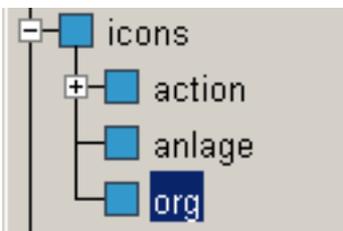


und neue Header-Bilder.

Solche Dateien können in einem ZIP-Archiv als Blob-Attribut in das Wissensnetz geladen werden. Dazu muss der Konfigurations-Begriff ein Blob-Attribut mit dem Namen *volumeResources* erhalten, in dem das Archiv gespeichert wird. Dieses wird beim Start der Applikation in das Unterverzeichnis *volumeResources/* des Applikationsverzeichnisses entpackt.

6.5.3 Konfiguration von Icons

Um das Icon festzulegen, das für die zu einer Layout-Kategorie gehörenden Knoten des Wissensnetzes auf der Suchergebnisseite angezeigt wird, muss der Begriff *icons* im **kpConfig**-Netz über einen Unterbegriff mit dem gleichen Namen wie der Wert des "*layoutCategory*"-Attributs des Begriffs verfügen.



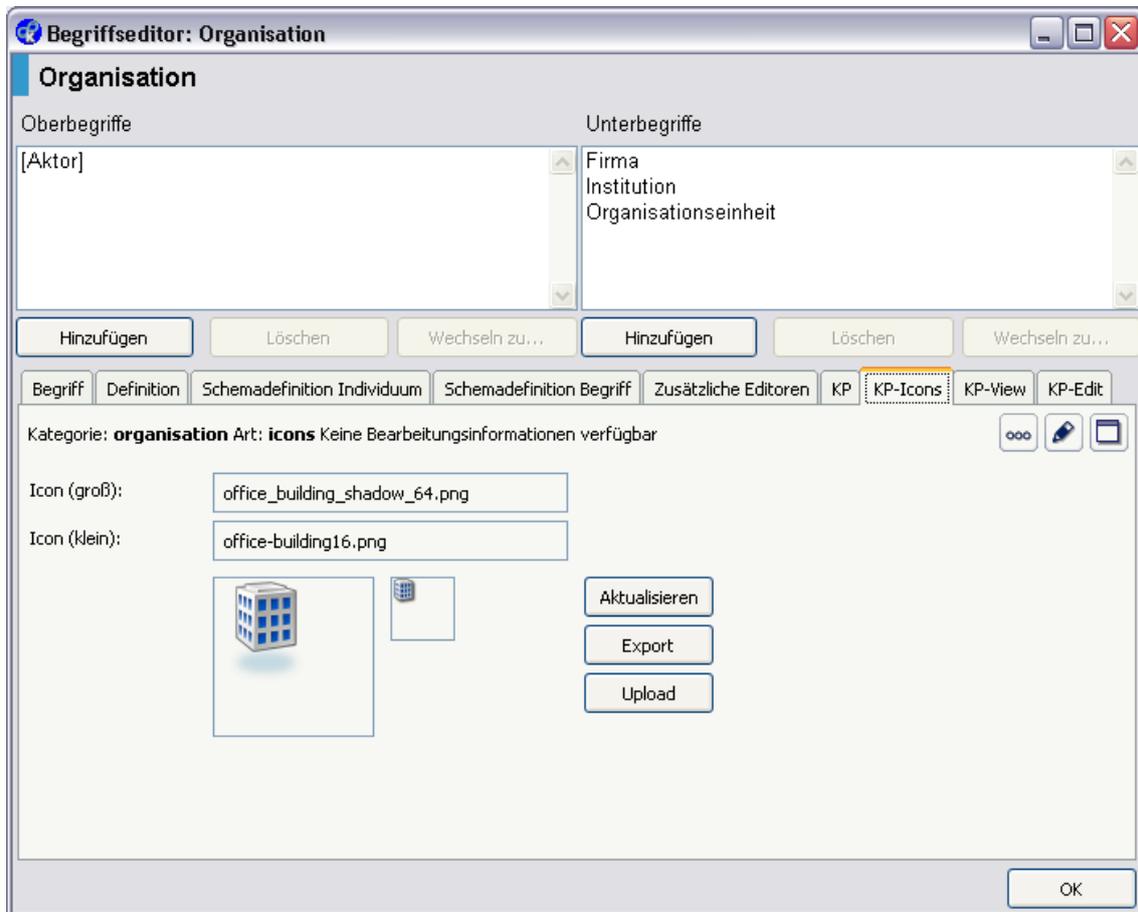
Teilstruktur des kpConfig-Netzes

An diesem können dann die beiden bei "*icons*" definierten Attribute (Attributtyp Zeichenkette) "*small*" und "*large*" gesetzt werden. Der Pfad muss dabei auf eine existierende Grafik-Datei verweisen, die im "*gfx/topicIcons*"-Verzeichnis der Web-Applikation liegt.

organisation	
Attribute	
Geschätzte Anzahl Individuen	0
large	volumeResources/gfx/topicIcons/office_building_shadow_64.png
Name	organisation
small	volumeResources/gfx/topicIcons/office-building16.png

Das Icon, das im Wert für "*small*" referenziert wird, wird bei der Anzeige im Suchergebnis verwendet. Der Wert im "*large*"-Attribut verweist auf das Icon, das bei der Detailansicht eines Knotens dieser Kategorie angezeigt wird.

Eine Ansicht der ausgewählten Icons im Knowledge-Builder erhält man auf dem Reiter *KP-Icons* im Begriffseditor des Begriffs, für den die Konfiguration gilt:



Neue Icons können mit Hilfe der volumeResources in das Applikations-Verzeichnis geladen werden (siehe Kapitel xxx)

6.5.4 Themes

Umfangreichere CSS-Anpassungen des Knowledge-Portal-Frontends können jetzt gebündelt in den Volume-Ressourcen abgelegt und über den Konfigurationsschlüssel "themes.current" gezielt aktiviert bzw. deaktiviert werden. Als Wert für den Konfigurationsschlüssel muss ein Theme-Name angegeben werden. Wenn unter dem angegebenen Namen kein Theme gefunden werden kann oder wenn der Konfigurationsschlüssel nicht definiert bzw. leer ist, wird das im Knowledge-Portal hinterlegte Default-Theme verwendet.

Wird ein Theme-Name angegeben, so geht das Knowledge-Portal davon aus, dass die zugehörigen Theme-Dateien in einem gleichnamigen Verzeichnis unterhalb des Verzeichnisses "themes" innerhalb der Volume-Ressourcen zu finden sind. Innerhalb des Theme-Verzeichnisses muss dann ein Verzeichnis namens "css" existieren, in dem die CSS-Definitionen abgelegt sind.

Das CSS-Verzeichnis MUSS die folgenden drei Dateien enthalten:

Dateiname	Beschreibung
-----------	--------------



styles_all.cssj	allgemeingültige CSS-Definitionen für alle Browser
styles_ie.cssj	spezielle CSS-Definitionen für den Microsoft Internet Explorer
styles.cssj	spezielle CSS-Definitionen für alle anderen Browser

Diese Dateien sind keine normalen CSS-Dateien, sondern JSP-XML-Dateien mit der Dateierweiterung .cssj. Am Anfang der Dateien muss deshalb immer der folgende Text stehen:

```
<?xml version="1.0" encoding="ISO-8859-1"?>  
<jsp:root version="1.2" xmlns:jsp="http://java.sun.com/JSP/Page">  
<jsp:directive.page contentType="text/css"/>
```

Danach können die CSS-Definitionen des jeweiligen Themes eingefügt werden. Innerhalb der CSS-Definitionen müssen alle "&" Zeichen durch "&" und alle "<"-Zeichen durch "<" ersetzt werden. Am Ende der Datei muss dann immer der folgende Text stehen:

```
</jsp:root>
```

Wenn zu einem Theme auch Bilddateien gehören, auf die in den CSS-Definitionen verwiesen wird, so legt man innerhalb des Theme-Verzeichnisses in den Volume-Ressourcen ein Unterverzeichnis namens "gfx" an und kopiert die Bilddateien in dieses Verzeichnis. In der CSS-Definition ersetzt man dann den Pfad zur Bilddatei durch die Zeichenkette "\${themeBasePath}/gfx/". Hier ein Beispiel:

```
background-image: url(${themeBasePath}/gfx/xy.gif);
```

Hier noch ein Beispiel für den Verzeichnisbaum eines Themes namens "roundCorner" in den Volume-Ressourcen:

```
themes  
+- roundCorner  
  +- css  
    +- styles.cssj  
    +- styles_all.cssj  
    +- styles_ie.cssj  
  +- gfx  
    +- ... in diesem Verzeichnis liegen die Bilddateien des Themes ...
```

Im Anhang zu diesem Abschnitt finden Sie eine ZIP-Datei, die das Default-Theme mit Vorlagen für alle benötigten Dateien und Verzeichnisse enthält.



defaultTheme.zip



6.6 Angepasste Net-Navigator-Konfiguration

6.7 Konfiguration des Seitenkopfes

Das Bild und der HTML-Code, der im Seitenkopf des Knowledge-Portals angezeigt wird, kann über die KP-Konfiguration angepasst werden. Dadurch ist es möglich, ein kundenspezifisches Bild im Seitenkopf anzuzeigen oder zusätzliche kundenspezifische Links in den Seitenkopf mit aufzunehmen.

Die folgenden Konfigurationsschlüssel stehen zur Anpassung des Seitenkopfes in der KP-Konfiguration zur Verfügung (im Namensraum `kpConfig.header`):

Konfigurationsschlüssel	Beschreibung
action	Dieser Konfigurationsschlüssel gibt den Namen der Struts-Action (genauer gesagt den Namen des Struts-Action-Mappings) an, welches beim Anklicken des Hintergrundbildes (definiert über den Konfigurationsschlüssel "backgroundImage"). Als Standardwert für diesen Schlüssel ist "exit" definiert, d.h. es wird die Exit-Action ausgeführt, welche die Benutzersitzung beendet und angemeldete Benutzer somit auch abmeldet. Als Alternative kann hier noch "home" angegeben werden, was dazu führt dass die Anwender einfach auf die Willkommenseite weitergeleitet werden (ohne Abmeldung).
additionalHtml	Unter diesem Konfigurationsschlüssel kann zusätzlicher HTML-Code angegeben werden, der unmittelbar hinter dem DIV in dem das Hintergrundbild angezeigt wird eingefügt werden soll. Damit lassen sich z.B. zusätzliche Links im Kopfbereich einfügen oder es lässt sich im Zusammenspiel mit angepassten CSS-Definitionen auch ein völlig anderer Inhalt im Seitenkopf platzieren (DIV mit dem Hintergrundbild über die CSS-Definition ausblenden, eigenen Inhalt über diesen Konfigurationsschlüssel einfügen).
asLink	Mit diesem Konfigurationsschlüssel lässt sich steuern, ob das über den Parameter "backgroundImage" angegebene Hintergrundbild mit der im Konfigurationsschlüssel "action" angegebenen Action verlinkt sein soll. Wird der Konfigurationsschlüssel auf den Wert "true" gesetzt, so wird das Bild mit einem Link hinterlegt. Die Auswirkungen des Klicks können über den Parameter "action" festgelegt werden. Ist der Wert "false" so wird das Bild nicht mit einer Aktion hinterlegt.



<p>backgroundImage (<i>deprecated</i>)</p>	<p>Hiermit kann die absolute oder relative URL zu der Bilddatei angegeben werden, die im Seitenkopf des Knowledge-Portals angezeigt werden soll. Als Standardwert wird die relative URL "gfx/header.jpg" verwendet, die auf das im Knowledge-Portal standardmäßig enthaltene Hintergrundbild verweist. Alternativ kann hier z.B. auf ein Hintergrundbild verwiesen werden, welches in den Volume-Ressourcen im Wissensnetz abgelegt ist (z.B. durch Angabe des Pfades "volumeResources/gfx/myHeader.jpg"). Näheres hierzu erfahren Sie unter dem Stichwort "Volume-Ressourcen".</p>
--	---

6.8 Cross-Site-Scripting Filterung (ab KP 3.8.8)

Der Cross-Site-Scripting-Filter steht ab der KP-Version 3.8.8 zur Verfügung und soll verhindern, dass durch Kommentare oder andere Texteingaben absichtlich aktive Inhalte (z.B. JavaScript oder Flash) in das System eingeschleust werden, die das normale Verhalten des Knowledge-Portals ändern um andere Benutzer auszuspionieren oder in die Irre führen zu können (z.B. durch Weiterleitung auf andere Webseiten).

Mit dem Konfigurationsschlüssel "system.filterXss" kann ab KP-Version 3.8.8 in der KP-Konfiguration festgelegt werden, dass vor dem Speichern von Textinhalten generell ein Cross-Site-Scripting-Filter auf die eingegebenen Inhalte angewendet werden soll. Wird der angegebene Konfigurationsschlüssel auf den Wert "true" gesetzt, so erfolgt eine Filterung vor dem Speichern von Kommentaren (ViewConfig / Element "Comments"), HTML-Inhalten im CK-Editor sowie Attributwerten auf der Pflegeseite.

Die Aktivierung des Filters auf globaler Ebene bewirkt nicht automatisch die Filterung von Inhalten bei der Ausgabe, da eine Filterung aller Textinhalte bei der Ausgabe die Seitenladezeit (abhängig von der ausgegebenen Textmenge) dramatisch erhöhen kann.

Wenn eine Filterung von Textinhalten vor der Ausgabe gewünscht ist, z.B. weil die Inhalte nicht über das KP eingegeben werden, so kann dies gezielt über den Render-Hint "filterXss" in der View-Config aktiviert werden (funktioniert auch in Column-Elementen von Objektlisten). Bei "htmltext" Attributen die nicht auf der Pflegeseite angezeigt werden, bewirkt dieser Render-Hint gleichzeitig auch eine Filterung beim Speichern.

Um die Filterung vor dem Speichern von Kommentaren zu aktivieren (nur notwendig, wenn "filterXss" global auf "false" gesetzt ist), muss in der ViewConfig am Element "Comments" das Attribut "filterXss" auf "true" gesetzt werden.

7 Detailseite für Begriffe und Individuen



7.1 Detailseiten konfigurieren

Mithilfe der Konfiguration von Detailseiten kann festgelegt werden, wie bestimmte Begriffe oder Individuen dargestellt werden. Die Konfiguration bezieht sich dabei immer auf einen bestimmten Begriff und seine Unterbegriffe oder auf die Individuen eines Begriffs.

Um eine Detailseite zu konfigurieren, muss eine View-Konfiguration erstellt werden. Diese wird in XML definiert und bietet folgende Möglichkeiten:

- Bestimmung welche Eigenschaften des Begriffs angezeigt werden
- Sortierung der Eigenschaften
- logische Gruppierung von Eigenschaften
- Modifikation der Darstellungsart (Renderer)
- Modifikation der Darstellungsparameter (Trennzeichen, Limits...)

Die einzelnen Möglichkeiten werden in den folgenden Kapiteln anhand von Beispielen erläutert.

Die erzeugten Konfigurationen werden den Begriffen/Individuen über die "*layoutCategory*" zugeordnet und gelten daher für alle Unterbegriffe. Für Begriffe und Individuen werden generell unterschiedliche Konfigurationen verwendet.

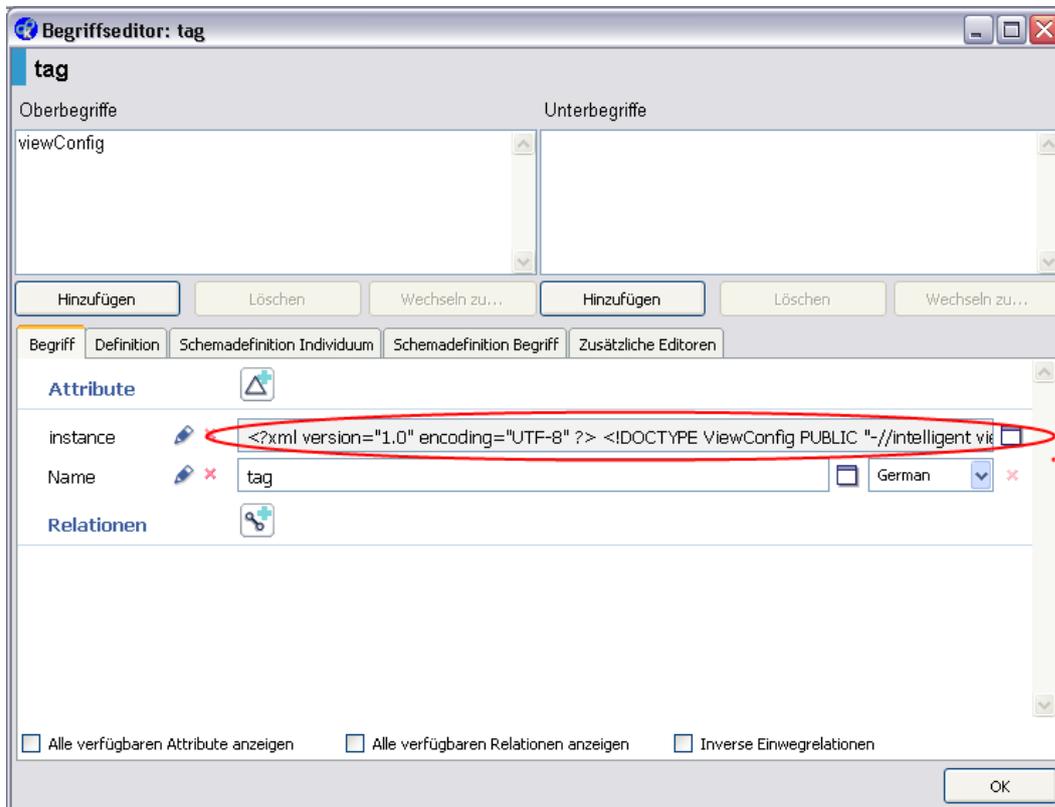
7.1.1 Erstellen einer View-Konfiguration

Die View-Konfigurationen werden im Wissensnetz konfiguriert. Dazu muss der Begriff "*viewConfig*" als Unterbegriff des Konfigurations-Begriffs "*kpConfig*" erstellt werden. Dieser dient als Oberbegriff für alle erstellten View-Konfigurationen.

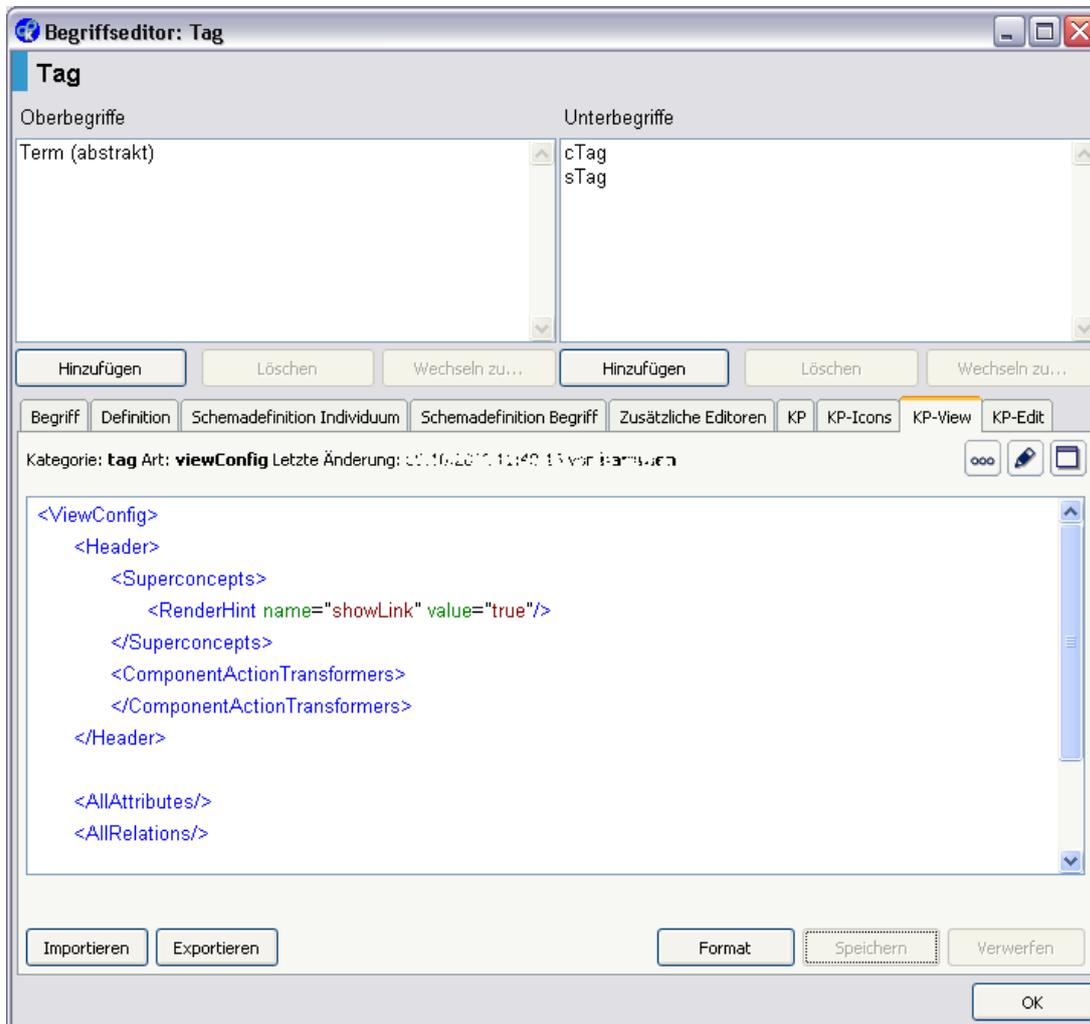
Um eine Konfiguration für eine bestimmte Layoutkategorie zu erstellen, muss ein Unterbegriff mit dem Namen der Layout-Kategorie erstellt werden, für die die Konfiguration angewendet werden soll. Für die Konfiguration des Personenbegriffs mit der Layout-Kategorie "*person*" würde also der Begriff "*kpConfig > viewConfig > person*" erstellt werden.

Die Konfigurationsdatei selbst wird in einem Textattribut an diesem Begriff gespeichert. Dabei sind die Attributnamen "*instance*" für Individuen dieses Begriffs oder "*concept*" für diesen Begriff und seine Unterbegriffe wählbar.

Die Konfigurationsdatei kann im Knowledge-Builder einerseits direkt im Attributeditor am Unterbegriff von *viewConfig* bearbeitet werden (siehe Abbildung).



Es besteht aber auch die Möglichkeit die Konfiguration für Individuen direkt im Begriffeditor des Begriffs auf dem Reiter *KP-View* zu bearbeiten (siehe Abbildung unten). Auf dem Reiter wird der Wert des Attributs *instance* bearbeitet.



Die Konfigurationsdatei selbst wird in XML definiert und muss nach der DTD "*viewConfig.dtd*" valide sein. Die DTD selbst finden Sie im Anhang dieses Dokuments.

Aufbau der Konfigurationsdatei

Um die Konfigurationsdatei richtig einlesen zu können, muss sie nach der "*viewConfig.dtd*" validiert werden. Daher ist die Angabe der DTD im XML-Header zwingend notwendig. Der Wurzelbegriff des XML-Dokuments muss das Element "<ViewConfig>" sein.

Das Gerüst der Konfiguration sieht also wie folgt aus:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE ViewConfig PUBLIC
  "-//intelligent views GmbH//DTD knowledgeportal viewConfig"
  "viewConfig.dtd">

<ViewConfig>
<!-- Definition der View-Konfiguration -->
</ViewConfig>
```

Default-Konfiguration



Die Standardkonfiguration beinhaltet alle Attribute, Relationen und Erweiterungen, die zu einem Objekt gehören und listet sie auch in dieser Reihenfolge auf. Die entsprechende Konfiguration sieht folglich so aus:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE ViewConfig PUBLIC
  "-//intelligent views GmbH//DTD knowledgeportal viewConfig"
  "viewConfig.dtd">

<ViewConfig>
  <AllAttributes />
  <AllRelations />
  </AllExtensions />
</ViewConfig>
```

Für das Anzeigen aller existierenden Eigenschaften werden hier die XML-Elemente "*AllAttributes*", "*AllRelations*" und "*AllExtensions*" verwendet. Diese Optionen suchen nach allen existierenden Elementen und listen diese nach Typ gruppiert auf.

Hinweis: Die Defaultkonfiguration können Sie ändern, indem Sie eine Konfiguration direkt am Begriff "*kpConfig > viewConfig*" anlegen. Beachten Sie aber bitte, dass diese auf alle Begriffe/Individuen angewendet wird, die keiner Layout-Kategorie zugeordnet sind.

7.1.2 Verfügbare Konfigurationselemente

Um die Detailseite nach eigenen Wünschen zu gestalten stehen verschiedene XML-Tags zur Verfügung. Diese erlauben, gezielt Attribute, Relationen und Erweiterungen zu selektieren und so deren Sichtbarkeit und Reihenfolge zu definieren. Zusätzlich existieren Tags, die eine Gruppierung und Möglichkeiten zur Interaktion bereitstellen.

Jedem dieser Tags können "RenderHints" übergeben werden. Dies sind Parameter, die die Darstellung und das Editierverhalten der einzelnen Attribut-Gruppen, Attribut-Werte, Label etc. beeinflussen. Eine Liste der möglichen RenderHints finden Sie jeweils bei der Beschreibung der Tags.

7.1.2.1 Hauptelemente

7.1.2.1.1 Attribute

Der Tag `<Attribute>` wird verwendet um Attribute zu konfigurieren. Diese werden über den internen Namen oder über einen KPath (auch von anderen Topics) gelesen. Dieser Tag bildet eine sogenannte Attributgruppe, da ein Attribut unter Umständen mehrfach vorkommen kann. Die Attributgruppe beinhaltet alle Instanzen - bzw. deren Werte - des angegebenen Attributes. Eine Gruppe besteht immer aus einem Label und einer Liste aller Instanzen.

Attribute für den Tag `<Attribute>`

Die Attribute des Tags `Attribute` ermöglichen die Übergabe von Parametern an die Attributgruppe. Parameter die für die einzelnen Elemente bestimmt sind, werden über `RenderHints` definiert.

Beispiel zum Definieren eines XML-Attributes:

```
<Attribute styleClass="myStyleClass" />
```



Name	Possible	Default	Beschreibung
styleClass	(String)		weist der Attributgruppe ein zusätzliches StyleSheet zu. Dieses kann selbst definiert werden.

Unterelemente <Label> oder <LabelKey>

Als Label wird ohne explizite Angabe der Name des Attributes benutzt. Optional kann hier durch das Unterelement <Label> ein fester String gesetzt werden, oder durch die Angabe von <LabelKey> ein Schlüssel für eine Übersetzung angegeben werden. (Diese muss in der msg.properties definiert werden!)

Unterelement <InternalName> oder <KPath>

Welches Attribute angesprochen werden soll wird über eines dieser Unterelemente definiert. Daher ist die Angabe des Unterelementents <InternalName> oder <KPath> zwingend erforderlich.

Unterelement <RenderHint>

Über dieses Konfigurationselement können dem Renderer, der HTML-Quelltext generiert, verschiedenen Parameter übergeben werden. Diese gelten immer für die Items, die in der Attributgruppe enthalten sind.

Name	Possible	Default	Beschreibung
showImage	true/false	true	[Nur Blob Attribute] false -> Es werden nur Links anstatt Bildern dargestellt.
showWhen-Empty	true/false	true	false -> Blendet die Gruppe aus wenn kein Attribut existiert.
hideLabel	true/false	false	ermöglicht das Ausblenden der Gruppenbeschreibung
itemClass	(String)		Zusätzliches Stylesheet das für die Items verwendet werden soll
itemSeparator	(String)	,	Zeichen durch die die Elemente der Gruppe getrennt werden
listMode	(String)		Dieser Render-Hint bewirkt, dass zur Auflistung der Attributwerte auf der HTML-Seite eine HTML-Aufzählung anstelle von DIV-Elementen erzeugt wird. Als Wert für diesen Render-Hint kann "ul" (Liste mit Aufzählungszeichen) oder "ol" (Liste mit Nummerierung) angegeben werden.
limit	(Integer)	5	Anzahl der Attribute die direkt nach dem Laden der Seite angezeigt werden sollen. Existieren mehr Attribute, so werden diese zunächst nicht angezeigt.



isLimited	true/false	false	Limitiert die Anzeige auf die Anzahl von "limit". Weitere Attribute werden ignoriert
format	true/false	true	Formatiert einen Ganzzahlwert und fügt Tausender-Trennzeichen ein.
minDigits	(Integer)	0	Minimale Anzahl der Nachkommastellen bei Fließkommawerten
maxDigits	(Integer)	12	Maximale Anzahl der Nachkommastellen bei Fließkommawerten

HINWEIS: Die Reihenfolge der Unterelemente muss in der XML-Definition mit der hier vorgegebene übereinstimmen.

Beispiele:

Alle hier gezeigten Beispiele gehen inhaltlich davon aus für Individuen von Personen definiert zu sein.

Konfiguration zum Ausgeben des Attributes "Vorname" (Zeichenkette)

```
<Attribute>
  <InternalName>firstname</InternalName>
</Attribute>
```

Konfiguration eines Attributes das an einem anderen Topic definiert ist. (Name der Firma)
Die Anzeige wird auf eine Firma limitiert. Arbeitet die Person für mehrere Firmen, können diese per Klick eingbledet werden. Die Firmennamen werden durch einen Zeilenumbruch voneinander getrennt.

```
<Attribute> <KPath>~$worksAt$/target()/name()</KPath>
  <RenderHint name="limit" value="1" />
  <RenderHint name="itemSeparator" value="&lt;br/&gt;" />
</Attribute>
```

Angabe eines alternativen Labels

```
<Attribute> <LabelKey>myConfig.person.fullname</LabelKey>
  <KPath>concat(" ",@$vorname$/value(),@$nachname$/value())</KPath>
</Attribute>
```

Angabe von RenderHints. Das Bild wird mit dem Stylesheet "pictureCenterd" gerendert. Dazu wird das Label ausgebledet, das Bild also auf voller Breite dargestellt. Darunter wird nochmals der Link zum Bild dargestellt.

```
<Attribute styleClass="pictureCentered">
  <InternalName>picture.big</InternalName>
  <RenderHint name="showLabel" value="false" />
</Attribute>
<Attribute>
  <Label>Downbload</Label>
```



```
<InternalName>picture.big<InternalName>  
<RenderHint name="showImage" value="false" />  
</Attribute>
```

7.1.2.1.2 Relation

Der Tag `<Relation>` wird verwendet um Relationen zu konfigurieren. Der aufbau erfolgt dabei gleich wie beim Konfigurationselement `<Attribute>`.

Attribute für den Tag `<Relation>`

Name	Possible	Default	Beschreibung
styleClass	(String)		weist der Relationsgruppe ein zusätzliches StyleSheet zu. Dieses kann selbst definiert werden.
metasMode	<i>off</i> <i>lazy</i> <i>direct</i>	off	Gibt an ob und wie Meta-Eigenschaft angezeigt werden sollen. <i>off</i> deaktiviert die Anzeige der Meta-Eigenschaften. Bei <i>direct</i> werden sie direkt eingeblendet, bei <i>lazy</i> erst auf Anfrage nachgeladen.

Unterelemente `<Label>` oder `<LabelKey>`

Als Label wird ohne explizite Angabe der Name des Attributes benutzt. Optional kann hier durch das Unterelement `<Label>` ein fester String gesetzt werden, oder durch die Angabe von `<LabelKey>` ein Schlüssel für eine Übersetzung angegeben werden. (Diese muss in der `msg.properties` definiert werden!)

Unterelement `<InternalName>` oder `<KPath>`

Welches Attribute angesprochen werden soll wird über eines dieser Unterelemente definiert. Daher ist die Angabe des Unterelements `<InternalName>` oder `<KPath>` zwingend erforderlich.

Unterelement `<RenderHint>`

Über dieses Konfigurationselement können dem Renderer, der HTML-Quelltext generiert, verschiedenen Parameter übergeben werden.

Name	Possible	Default	Beschreibung
showWhen-Empty	true/false	true	false -> Blendet die Gruppe aus wenn kein Attribut existiert.
hideLabel	true/false	false	ermöglicht das Ausblenden der Gruppenbeschreibung
itemClass	(String)		Zusätzliches Stylesheet das für die Items verwendet werden soll
itemSeparator	(String)	,	Trennzeichen, das die Liste von Relationzeilen voneinander trennt



limit	(Integer)	5	Anzahl der Relationen, die direkt nach dem Laden der Seite angezeigt werden sollen. Existieren mehr Relationen, so werden diese zunächst nicht angezeigt.
listMode	(String)		Dieser Render-Hint bewirkt, dass zur Auflistung der Attributwerte auf der HTML-Seite eine HTML-Aufzählung anstelle von DIV-Elementen erzeugt wird. Als Wert für diesen Render-Hint kann "ul" (Liste mit Aufzählungszeichen) oder "ol" (Liste mit Nummerierung) angegeben werden.
isLimited	true/false	false	Schaltet den Link zum Einblenden der weiteren Relationsziele aus

HINWEIS: Die Reihenfolge der Unterelemente muss in der XML-Definition mit der hier vorgegebenen übereinstimmen.

Unterelement `<ItemActionTransformers>`

Definiert eine Liste von Action-Transformern, die zur Erzeugung von Actions für alle angezeigten Relationen ausgeführt werden sollen. Details zu den verfügbaren Action-Transformern sind im Kapitel "Referenz" zu finden.

Beispiele:

Alle hier gezeigten Beispiele gehen inhaltlich davon aus für Individuen von Personen definiert zu sein.

Konfiguration zum Ausgeben der Relation "arbeitet bei"

```
<Relation>
  <InternalName>worksFor</InternalName>
</Relation>
```

Konfiguration einer Relation die an anderen Topic definiert ist. (Länder in denen die Person gearbeitet hat)

```
<Relation>
  <Label>Arbeitet in Ländern</Label>
  <KPath>~$worksInProject$/target()/~$hasLocation$/target()/~$isPartOfCountry$
</KPath>
</Relation>
```

7.1.2.1.3 Extension

Der `<Extension>`-Tag wird zum einblenden von Erweiterungen verwendet. Diese werden über den internen Namen selektiert.

Alternativ können die Erweiterungen auch über den über diesen Tag auch andere Begriffe/Individuen spezifiziert werden. Der Pfad zu diesen Objekten wird mit dem Unterele-



ment `<KPath>` spezifiziert.

Attribute für den Tag `<Extension>`

Name	Possible	Default	Beschreibung
styleClass	(String)		weist der Relationsgruppe ein zusätzliches StyleSheet zu. Dieses kann selbst definiert werden.
layoutCategory	(String)		Definiert die View-Konfiguration, mit der die Erweiterung dargestellt werden soll. Hier wird immer die Konfiguration für "extension" verwendet. Ist dieser Parameter leer, werden alle Attribute und alle Relationen ausgegeben
itemLabelKPath	(KPath)	conceptOr-Self()/name()	Hier kann ein KPath angegeben werden durch den die Beschreibung der einzelnen Erweiterungen generiert wird. Dieser KPath ist immer ausgehend vom Erweiterungsobjekt!

Unterelemente `<Label>` oder `<LabelKey>`

Als Label wird ohne explizite Angabe der Name der Erweiterung benutzt. Optional kann hier durch das Unterelement `<Label>` ein fester String gesetzt werden, oder durch die Angabe von `<LabelKey>` ein Schlüssel für eine Übersetzung angegeben werden. (Diese muss in der `msg.properties` definiert werden!)

Unterelement `<InternalName>` oder `<KPath>`

Über das Unterelement `<InternalName>` gibt alle Erweiterungen aus, die vom Typ des spezifizierten Begriffes oder dessen Unterbegriffe sind.

Unterelement `<KPath>` ermöglicht den Verweis auf andere Objekte im Wissensnetz. Hier muss darauf geachtet werden, dass durch das Attribut "layoutCategory" eine entsprechende View-Konfiguration gesetzt wird, da sonst alle Attribute und Relationen an diesem Objekt ausgegeben werden.

Unterelement `<RenderHint>`

Über dieses Konfigurationselement können dem Renderer, der HTML-Quelltext generiert, verschiedenen Parameter übergeben werden.

Name	Possible	Default	Beschreibung
showWhen-Empty	true/false	false	false -> Blendet die Gruppe aus wenn kein Attribut existiert.
alwaysShow-Header	true/false	false	Zeigt den Header auch an wenn er leer ist
hideLabel	true/false	false	Blendet den Header in jedem Fall aus



isCollapsable	true/false	true	ermöglicht die Erweiterung ein- und auszuklappen
isCollapsed	true/false	false	legt fest ob die Erweiterung initial eingeklappt ist oder nicht
showNumberOfExtensions	true/false	true	Zeigt die Anzahl der in der Gruppe enthaltenen Objekte an
itemClass	(String)		Zusätzliches Stylesheet das für die Items verwendet werden soll
limit	(Integer)	5	Anzahl der Relationsziele die per Default eingeblendet werden sollen. Der Rest kann per klick eingeblendet werden.
isLimited	true/false	false	Schaltet den Link zum Einblenden der weiteren Relationsziele aus

HINWEIS: Die Reihenfolge der Unterelemente muss in der XML-Definition mit der hier vorgegebene übereinstimmen.

Beispiele:

Alle hier gezeigten Beispiele gehen inhaltlich davon aus für Individuen von Personen definiert zu sein.

Zeigt die UnternehmensRolle der Person an. Hat die Person mehrere Rollen inne, können diese ausgeklappt werden. Jede Rolle wird nach der View-Konfiguration von "personRole" dargestellt. Als Name wird das Start-Datum und der Name der Erweiterung angegeben.

```
<Extension layoutCategory="personRole" itemLabelKPath="concat(":",@$$since$/value(),conceptOrSelf()/name())">
  <InternalName>role</InternalName>
  <RenderHint name="limit" value="1" />
</Extension>
```

Kurzer Überblick über alle Projekte an denen die Person beteiligt ist.

```
<Extension layoutCategory="projektStatus">
  <LabelKey>viewConfig.person.projektStatus</LabelKey>
  <KPath>~$worksInTask$/target()/~$isPartOfProject$/target()</KPath>
</Relation>
```

ViewConfig für projektStatus > instance:

```
<ViewConfig>
  <Attribute><KPath>name()</KPath></Attribute>
  <Attribute><InternalName>project.date.end</InternalName></Attribute>
  <Attribute><InternalName>project.status</InternalName></Attribute>
```



</ViewConfig>

7.1.2.1.4 CustomGroup

Die *<CustomGroup>* ermöglicht die Bildung einer Gruppe aus beliebigen Attributen und/oder Relationen. Sie stellt einen Container für die angegebenen Unter-elemente dar, der über die gleichen Interaktionsmöglichkeiten verfügt wie das Element *<Extension>*.

CustomGroup kann z.B. dazu genutzt werden, eine Adresse als eine Einheit darzustellen.

Über die Angabe des Tag-Attributes *kpath* kann auch auf andere Objekte verwiesen werden. Das Ziel muss dabei immer ein Begriff oder eine Instanz sein. Werden über diesen KPath mehrere Zielobjekte referenziert, werden pro Zielobjekt eine Gruppe dargestellt.

Alle Elemente die innerhalb der CustomGroup definiert sind, beziehen sich auf das referenzierte Objekt.

Attribute für den Tag *<CustomGroup>*

Name	Possible	Default	Beschreibung
styleClass	(String)		weist der Relationsgruppe ein zusätzliches StyleSheet zu. Dieses kann selbst definiert werden.
kpath	(KPath)		Dieser KPath führt zum Objekt, auf dem die Eigenschaften ausgelesen werden, die innerhalb der CustomGroup spezifiziert sind. Wird kein KPath angegeben, beziehen sich diese auf das aktuelle Objekt.

Unterelemente *<Label>* oder *<LabelKey>*

Als Label wird ohne explizite Angabe der Name der Erweiterung benutzt. Optional kann hier durch das Unterelement *<Label>* ein fester String gesetzt werden, oder durch die Angabe von *<LabelKey>* ein Schlüssel für eine Übersetzung angegeben werden. (Diese muss in der *msg.properties* definiert werden!)

Unterelement zur Auswahl der anzuzeigenden Eigenschaften

Als Unterelemente können hier alle Hauptelemente benutzt werden. Die CustomGroup bildet nur einen Container um diese Objekte und ermöglicht den Zugriff auf andere Begriffe.

Im Gegensatz zum *<Extension>*-Tag muss hier keine *LayoutCategory* angegeben werden. Die CustomGroup rendert immer mit der View-Konfiguration im Body. Es ist also eine Art View-Konfiguration in der View-Konfiguration.

Unterelement *<RenderHint>*

Über dieses Konfigurationselement können dem Renderer, der HTML-Quelltext generiert, verschiedenen Parameter übergeben werden.

Name	Possible	Default	Beschreibung
------	----------	---------	--------------



alwaysShow-Header	true/false	false	True -> Gibt an, dass die Überschrift eines Gruppenelementes angezeigt werden soll, wenn dieses weniger als zwei Unterelemente enthält.
showContainerTitle	true/false	false	Gibt an, ob der Überschriftenbereich der CustomGroupContainerViewComponent angezeigt werden soll (true) oder nicht (false). Da die einzelnen CustomGroups jeweils über einen eigenen Überschriftenbereich verfügen, ist der Überschriftenbereich des Containers normalerweise ausgeblendet. Der RenderHint muss auf "true" gesetzt werden, damit Actions, welche mittels "ComponentActionTransformers" konfiguriert sind, angezeigt werden.
showWhenEmpty	true/false	true	false -> Blendet die CustomGroup aus, wenn diese keine Elemente enthält.
showContainerEmptyMessage	true/false	true	true -> zeigt eine Meldung 'keine Einträge' an, wenn die CustomGroup keine Elemente enthält
hideLabel	true/false	false	Gibt an, ob Überschriften von Gruppenelementen grundsätzlich angezeigt werden sollen (true) oder nicht (false).
isCollapsible	true/false	true	ermöglicht die Erweiterung ein- und auszuklappen
isCollapsed	true/false	false	legt fest ob die Erweiterung initial eingeklappt ist oder nicht
itemClass	(String)		Zusätzliches Stylesheet das für die Items verwendet werden soll
limit	(Integer)	5	Anzahl der Relationsziele die per Default eingeblendet werden sollen. Der Rest kann per klick eingeblendet werden.
isLimited	true/false	false	Schaltet den Link zum Einblenden der weiteren Relationsziele aus
listMode	(String)		Render-Mode für die Liste der CustomGroupViewComponents.

HINWEIS: Die Reihenfolge der Unterelemente muss in der XML-Definition mit der hier vorgegebene übereinstimmen.



7.1.2.1.5 TabGroup

Das TabGroup-Element erlaubt die Gruppierung anderer ViewConfig-XML-Elemente in Reitern (auch Registerkarten oder Tabs genannt).

Attribute	Relationen	Weitere Instanzen
hat Teil (*)	keepactivetab1	
ist Individuum von (*)	Basisnetz TabGroup-Tests Testbegriffe für ViewConfigs Wurzelbegriff Wurzelbegriff für normale Begriffe	
relation with metas (global)	Baden-Wuerttemberg	

Beispiel:

```
<TabGroup>
  <Tab>
    <Label>Relationen</Label>
    <AllRelations/>
  </Tab>
  <Tab>
    <Label>Attribute</Label>
    <AllAttributes/>
  </Tab>
</TabGroup>
```

Zu jedem Tab muss eine Überschrift konfiguriert werden, welche im Titel des Reiters angezeigt wird. Hierfür können die folgenden Unterelemente verwendet werden:

- "Label" - Einfache Zeichenkette
- "LabelKey" - Zeichenkette aus Message-Bundle
- "LabelKPath" - Rückgabewert von KPath-Ausdruck, auch in Kombination mit "LabelKey" verwendbar

Nach dem Überschrift-Element kann dann ein beliebiges ViewConfig-XML eingefügt werden, welches den im Tab anzuzeigenden Inhalt definiert. Auch das Element "TabGroup" kann hier angegeben werden, d.h. es sind auch mehrfach ineinander geschachtelte Tab-Gruppen möglich.

Beim erstmaligen Laden einer Seite wird normalerweise das erste innerhalb der TabGroup definierte Tab aktiviert bzw. sichtbar. Wenn der Benutzer ein anderes Tab auswählt, so merkt sich das System diese Auswahl und aktiviert nach einem Seiten-Reload oder beim Aufruf einer anderen Detailseite mit derselben ViewConfig das zuletzt aktivierte Tab automatisch. Erst wenn die Benutzersitzung abläuft oder vom Benutzer aktiv beendet wird (Logout), wird



die Tab-Auswahl wieder auf das erste Tab innerhalb der TabGroup zurückgesetzt.

Dieses Standardverhalten kann über die folgenden Attribute am TabGroup-Element verändert werden:

Attributname	Beschreibung
keepActiveTab	Gibt an, ob das zuletzt vom Benutzer aktivierte Tab pro ViewConfig (Wert "true") oder pro Topic bzw. pro Detailseite (Wert "false") gespeichert werden soll. Der Standardwert ist "true".
resetActiveTabOnReload	Gibt an, ob die Tab-Auswahl des Benutzers für diese Tab-Group bei jedem Aufruf einer Detailseite zurückgesetzt werden soll ("true") oder nicht ("false"). Der Standardwert ist "false". Wird der Wert auf "true" gesetzt, so wird das Attribut "keepActiveTab" nicht mehr ausgewertet.
activeTabIDKPath	KPath-Ausdruck, der die ID des initial anzuzeigenden Tabs zurückliefert. Voraussetzung hierfür ist, dass die Tab-Unterelemente über das Attribut "id" mit einer ID versehen wurden. Wird dieses Attribut nicht angegeben, so wird automatisch das erste Tab innerhalb einer TabGroup aktiviert (Standardverhalten). Ein Beispiel für diese Funktionalität ist unten zu finden.
enabledKPath	KPath-Ausdruck, der einen booleschen Wert liefert. Wird der Wert 'false' geliefert, so wird der Tab, an dem das Attribut angebracht ist, nicht angezeigt. Wenn 'true' geliefert wird oder wenn das Attribut nicht definiert wurde, wird das Tab angezeigt. Falls das Tab bei letzten Seitenaufruf aktiv war und nun ausgeblendet wird, wird das erste sichtbare Tab aktiviert.

Das folgende Beispiel definiert eine TabGroup, in bei der das zweite Tab initial angezeigt werden soll:

```
<TabGroup activeTabIDKPath="attributes">  
  <Tab>  
    <Label>Relationen</Label>  
    <AllRelations/>  
  </Tab>  
  <Tab id="attributes" enabledKPath="conceptOrSelf() = .">  
    <Label>Attribute</Label>  
    <AllAttributes/>  
  </Tab>  
</TabGroup>
```

Es können natürlich auch komplexere KPath-Ausdrücke oder KScript-Funktionen aufgerufen werden, die abhängig vom Benutzer oder abhängig vom Wissensnetzmodell unterschiedliche Tab-IDs zurückliefern.



7.1.2.1.6 TableMapping

Das Element `<TableMapping>` ermöglicht die tabellarische Ansicht von Objekte. Eine Menge von Objekten, die über einen KPath spezifiziert werden kann, wird in ein oder mehreren Spalten in Tabellenform dargestellt.

WICHTIG: Von der Verwendung des TableMapping-Elements mit große Mengen von Objekten wird abgeraten, da es hierdurch zu Performance-Problemen (insbesondere in Produkktivsystemen) kommen kann. Ab Knowledge-Portal Version 3.7 steht als Ersatz für TableMapping das Element ObjectList zur Verfügung, mit dem auch für große Mengen von Objekten performant Listen erzeugt werden können.

Konfiguration des Komponente

```
<TableMapping
  folderName="exportFolderExternalId"
  mappingName="Name der Abbildung"
  kpath=".">

</TableMapping>
```

Attribute des Elements `<TableMapping>`

Name	Standardwert	Beschreibung
folderName	(String)	Die externe ID des Ordners, der das Mapping enthält. Der Ordner muss vom Typ "Ordner für die Abbildung externer Daten" sein.
mappingName	(String)	Der Name des Mappings, das im konfigurierten Ordner liegt.
kpath	.	Der KPath, der die anzuzeigenden Topics liefert. Wird dieses Attribut nicht angegeben, dann wird das gerade angezeigte Topic verwendet, d.h. der KPath "." .

Das Element `<TableMapping>` kann als Element der obersten Ebene und innerhalb der Elemente `<CustomGroup>` und `<TabGroup>` eingefügt werden.

Unterelement `<RenderHint>`

Die Komponente TableMapping unterstützt die folgenden RenderHints, die als Unterelemente angegeben werden.

RenderHint	Mögliche Werte	Standardwert	Beschreibung
showWhen-Empty	true / false	false	Wenn der KPath eine Leere Menge liefert, dann wird die Tabelle trotzdem angezeigt, wenn der Wert dieses Renderhints auf 'true' gesetzt ist.



searchMode	filter / index		<p>Möchte man eine Sprungleiste mit einem Index A-Z über der Tabelle anzeigen, dann muss der Wert auf 'index' gesetzt werden.</p> <p>Mit dem Wert 'filter' erhält man über jeder Spalte eine Eingabemaske, mit der man die Anzeige der Tabelle auf die Zellen einschränken kann, die den eingegebenen Text enthalten.</p>
------------	----------------	--	---

Definition des Export-Mappings

Die Definition der Tabelle geschieht über eine Abbildung externer Daten im Knowledge-Builder (d.h. über eine Export-Definition). Alle Elemente, die über den konfigurierten KPath gefunden werden, werden in der Tabelle angezeigt. Somit müssen die Tabellendefinition und der KPath aufeinander abgestimmt werden.

Es macht beispielsweise keinen Sinn, in der Export-Definition Begriffe abzubilden und mit dem KPath Individuen zurückzugeben.

Die Spalten der angezeigten Tabelle richten sich nach den konfigurierten Spalten der Export-Definition.

Möchte man eine neue Spalte für die Anzeige definieren, legt man also manuell eine neue Spalte vom Typ "Standard" in der Export-Definition an. Der hier eingegebene Name der Spalte wird als Spaltenüberschrift in der angezeigten Tabelle verwendet.

Um die Zellen der Tabelle mit Werten zu füllen legt man eine neue Attributabbildung in der Export-Definition an. Relationsabbildungen werden nicht unterstützt.

Verweis auf das angezeigte Individuum

Jeder Zeile der Tabelle bildet die Daten eines einzelnen Individuums oder eines einzelnen Begriffs ab. Möchte man nun einen Link auf dieses Individuum konfigurieren, muss man eine neue DMID-Abbildung auf eine gesonderte Spalte anlegen. Der Name der Spalte muss mit dem Prefix "dmid." beginnen. Das Suffix ist optionale, gültige Namen sind beispielsweise "dmid.", "dmid.DMID" oder "dmid.Projekt". Diese besondere Spaltenart wird im Knowledge-Portal nicht angezeigt, sondern nur für die Referenzierung des aktuellen Topics verwendet. Der Link auf das Topic wird immer in der ersten Spalte der Tabelle eingefügt.

Zelleninhalte

Als Inhalte der Tabellenzellen werden nur Attribute unterstützt. Die Auswertung von Relationsabbildungen wird nicht unterstützt.

7.1.2.1.7 Report

Element <Report>

Dieses Element wird dazu verwendet, um die Ausgabe eines KScript-Reports auf einer Detailseite auszugeben. Der Inhalt kann optional als HTML interpretiert werden.

Attribute von <Report>



Attribut	Mögliche Werte	Standardwert	Beschreibung
folderName	(String)		Die externe ID des ReportFolders, d.h. die ID, die den Ordner im Knowledge-Builder bezeichnet, der der Report mit dem Namen 'reportName' enthält.
reportName	(String)		Der Name des Reports, der ausgeführt werden soll. Änderungen an diesem Report werden sofort im Knowledge-Portal angewandt. Es ist kein Neuladen der Konfiguration erforderlich.
escapeHtml	true / false	false	<ul style="list-style-type: none">• <i>true</i> : Die Ausgabe des Reports wird als reiner Text interpretiert.• <i>false</i> : Die Ausgabe wird als HTML interpretiert, d.h. der Report kann dazu verwendet werden, HTML in die Seite einzufügen.
initialTopic- skpath	(String)	/	Ein KPath, der die Liste der initialen Topics ausrechnet, auf die der Report angewandt wird. Wird der KPath nicht angegeben, wird der Wurzelbegriff als Eingabeobjekt verwendet.
renderMode	(String)		Falls ein besonderer RenderMode für das Rendern der Komponente verwendet werden soll, kann dieser hier angegeben werden. Vordefinierte RenderModes: <ul style="list-style-type: none">• <i>property</i> : In diesem Mode wird die Ausgabe wie ein Attribut dargestellt. Das Label erscheint auf der linken Seite, die Ausgabe erscheint wie ein Attributwert auf der rechten Seite.
showWhen- Empty	true / false	false	<ul style="list-style-type: none">• <i>true</i> : Die Komponente wird angezeigt, auch wenn die Ausgabe des Reports leer ist.• <i>false</i> : Die Komponente wird nur angezeigt, wenn auch eine Ausgabe vorhanden ist.



styleClass	(String)		Eine zusätzliche CSS Klasse für die Komponente. In jedem Fall werden die CSS Klassen <i>customgroup</i> und <i>reportViewComponent</i> verwendet.
id	(String)		Die CSS ID, die für die Ausgabe der Komponente verwendet wird.

Unterelemente <Label> / <LabelKey> / <LabelKPath>

Diese Elemente werden verwendet, um das Label der Komponente anzugeben. Es darf jeweils nur eines der Elemente gleichzeitig in einer Komponente verwendet werden.

Wird keines der Elemente angegeben, dann wird keine Bezeichnung über der Komponente eingefügt. In diesem Falle muss, damit etwas erscheint, der RenderHint isCollpased den Wert 'false' haben, da sonst die Ausgabe nicht aufgeklappt werden kann und somit

Unterelement <RenderHint>

Render-Hint	Mögliche Werte	Standardwert	Beschreibung
isCollapsible	true / false	true	
isCollapsed	true / false	false	
hideLabel	true / false	false	<ul style="list-style-type: none">• <i>true</i> : Das Label der Komponente wird versteckt• <i>false</i> : Das Label wird angezeigt.

Unterelement <ReportParameters>

Über das Element ReportParameters können zusätzliche Werte an den Report übergeben werden. Diese werden immer als String übergeben.

```
...  
<ReportParameters>  
  <Parameter name="param_name" value="param_value" /> </ReportParameters>  
..
```

Die Unterelemente müssen in der Reihenfolge verwendet werden, wie sie hier beschrieben sind.

Beispielkonfiguration

In diesem Beispiel werden alle Individuen eines Begriffs in einer Aufzählungsliste dargestellt.

Komponentenkonfiguration

```
<Report folderName="reportFolderExternalId"
```



```
reportName="projectsReport"  
initialTopicsKpath="//$projects$/instances()/sort(name(), true)"  
escapeHtml="false">  
<Label>Projekte</Label>  
  
<RenderHint name="isCollapsed" value="true"/>  
</Report>
```

Report

Das Element `<Path path=".">` bezieht sich auf die Menge der Eingabeobjekte.

```
<Script output="xml">  
  <ul>  
    <Path path=".">  
      <Do>  
        <li><path path="name()"/></li>  
      </Do>  
    </Path>  
  </ul>  
</Script>
```

7.1.2.1.8 AllAttributes

`<AllAttributes />` listet alle verfügbaren Attribute am Objekt auf. Diese sind nach Attributgruppen gruppiert und nach dem Namen des Attribut alphabetisch sortiert. Dieser Tag wird auch in der Default-Konfiguration verwendet.

Als Attribut kann ein zusätzliches Stylesheet übergeben werden.

Attribute für den Tag `<AllAttributes />`

Name	Possible	Default	Beschreibung
styleClass	(String)		weist der Relationsgruppe ein zusätzliches StyleSheet zu. Dieses kann selbst definiert werden.
hidden	(String)		In diesem Attribut kann analog zu den global einstellbaren hiddenAttributes eine Liste der internen Namen von Attributen angegeben werden, die ausgeblendet werden sollen. Die internen Namen müssen durch ein Leerzeichen voneinander getrennt sein. Außerdem kann das Stern-Zeichen "*" als Wildcard verwendet werden (z.B. "hidden*", um alle Attribute mit dem Prefix "hidden" zu ignorieren).



Unterelement

Name	Possible	Default	Beschreibung
InternalName	(String)		Interner Name eines Attributbegriffs, der zur Berechnung der Attribute verwendet werden soll (ab KP 4.4.5)

7.1.2.1.9 AllRelations

<AllRelations /> listet alle verfügbaren Relationen am Objekt auf. Diese sind nach Relationart gruppiert und nach dem Namen der Relation alphabetisch sortiert. Dieser Tag wird auch in der Default-Konfiguration verwendet.

Als Tag-Attribut kann ein zusätzliches Stylesheet übergeben werden.

Attribute für den Tag <AllRelations />

Name	Possible	Default	Beschreibung
styleClass	(String)		weist der Relationsgruppe ein zusätzliches StyleSheet zu. Dieses kann selbst definiert werden.
hidden	(String)		In diesem Attribut kann analog zu den global einstellbaren hiddenRelations eine Liste der internen Namen von Relationen angegeben werden, die ausgeblendet werden sollen. Die internen Namen müssen durch ein Leerzeichen voneinander getrennt sein. Außerdem kann das Stern-Zeichen "*" als Wildcard verwendet werden (z.B. "hidden*", um alle Relationen mit dem Prefix "hidden" zu ignorieren).
showOneway-InverseRelations	true / false	true	Ab KP 3.9.26. Hiermit lässt sich steuern, ob die Ziele von inversen Einwegrelationen gezeigt werden sollen oder nicht. Die Berechnung dieser Ziele kann je nach Anwendungsfall sehr aufwändig sein. Es wird dringend angeraten, inverse Einwegrelationen auszublenden. Ab Version 3.10.0 ist der Standardwert "false", d.h. ausgeblendet. In vorigen Versionen werden inverse Einwegrelationen ohne manuelle Anpassung angezeigt.



Unterelemente

Name	Possible	Default	Beschreibung
InternalName	(String)		Interner Name eines Attributbegriffs, der zur Berechnung der Attribute verwendet werden soll (ab KP 4.4.5)

7.1.2.1.10 AllExtensions

`<AllExtensions />` ermittelt alle existierenden Erweiterungen für das Individuum. Die gefundenen Erweiterungen werden nach gemeinsamen Konzepten Gruppieren. Zum Darstellen der Erweiterung wird das jeweilige Konzept nach einer `layoutCategory` gefragt und entsprechend die `viewConfig` selektiert. Ist keine `layoutCategory` konfiguriert, wird die Erweiterung mit "`<AllAttributes /><AllRelations />`" dargestellt.

Als Tag-Attribut kann ein zusätzliches Stylesheet übergeben werden.

Attribute für den Tag `<AllExtensions />`

Name	Possible	Default	Beschreibung
styleClass	(String)		weist der Relationsgruppe ein zusätzliches StyleSheet zu. Dieses kann selbst definiert werden.

Unterelemente sind nicht möglich!

7.1.2.1.11 ObjectList (ab Version 3.7)

Dieses Element ist ab Knowledge-Portal 3.7 im ViewConfig-XML verwendbar. Es kann verwendet werden, um basierend auf einer Menge von Topics (z.B. einer Menge von Individuen) eine Tabelle mit Attributwerten und Relationszielen zu erzeugen.

Die Objektlisten sind als Kernfunktion von K-Infinity ab Version 3.2 sowohl im Knowledge-Builder als auch im Knowledge-Portal verfügbar. Die Konfiguration der Objektlisten erfolgt mittels des Admin-Tools, wobei für ein bestimmtes Konzept (z.B. Person) eine Menge von Objektlistenkonfigurationen (Spaltenkonfigurationen) definiert werden kann. Zur Auswahl einer Objektlistenkonfiguration für die Anzeige muss also ein Konzept und eine Unterkonfiguration spezifiziert werden.

Wenn die anzuzeigende Topic-Menge mehr als zehn Topics enthält, so werden die Tabellenzeilen auf mehrere Seiten von standardmäßig 10 Zeilen pro Seite aufgeteilt. Der Benutzer kann die Tabelle dann seitenweise durchblättern. Die Objektliste ist daher zur Anzeige von Wertetabelle für größere Topic-Mengen wesentlich besser geeignet als die mit dem Element "TableExport" erzeugten Wertetabellen. Bestehende Wertetabellen in der ViewConfig sollten daher auf die Objektlisten umgestellt werden. Das Element "TableExport" wird mittelfristig nicht mehr in der ViewConfig verfügbar sein.

Attribute von `<ObjectList>`



Attribut	Mögliche Werte	Standardwert	Beschreibung
itemsKPath	String		KPath zur Bestimmung der in der Objektliste angezeigten Topic-Menge. Wird dieses Attribut nicht angegeben, so zeigt die Objektliste entweder alle Individuen oder alle Unterbegriffe (abhängig vom Attribut "forInstances") des errechneten oder mittels "conceptInternalName" manuell angegebenen Konzepts an. Dieses Feld ist ein Pflichtfeld in KP 3.7. Im KP 3.8 ist das Feld optional
conceptInternalName	String		Der interne Name eines Konzepts zur Auswahl der Objektlistenkonfiguration. Wird das Attribut nicht angegeben, so wird das aktuell angezeigte Konzept bzw. das Konzept des aktuell angezeigten Individuums verwendet. Ist das Attribut "itemsKPath" angegeben, so wird der erste Eintrag der von KPath zurückgelieferten Topic-Menge zur Berechnung des Konzepts herangezogen. Dieses Feld ist ein Pflichtfeld in KP 3.7. Im KP 3.8 ist das Feld optional.
tableName	String		Der Name einer Unterkonfiguration zur Auswahl der Objektlistenkonfiguration. Dieses Feld ist ein Pflichtfeld in KP 3.7. Im KP 3.8 ist das Feld optional
forInstances	Boolesch	true	Gibt an, ob eine Objektlistenkonfiguration zur Anzeige von Individuen (true) oder Unterbegriffen (false) verwendet werden soll.
sortKPath	String		Ein KPath, der für jedes Topic der Objektliste evaluiert werden soll, um das Sortierkriterium zu bestimmen. Wird dieses Attribut nicht angegeben, so werden die Objekte der Objektliste nach der ersten Spalte sortiert



sortAscending	String	true	Gibt an, ob beim initialen Sortieren der Einträge der Objektliste aufsteigend (true) oder absteigend (false) sortiert werden soll. Diese Einstellung kann durch das Attribut "sortAscending" am Columns-Unterelement überschrieben werden.
sortColumn	Ganzzahl	0	<i>Ab Version 3.8.</i> Index der Objektlistenspalte (0..n) nach der initial sortiert werden soll. Das Attribut wird nur ausgewertet, wenn das Attribut "sortKPath" nicht angegeben ist.
sortColumns	Ganzzahl, komma-separiert		<i>Ab Version 3.8.7.</i> Kommaseparierte Aufzählung von Spalten-Indices (0..n), nach denen die Liste initial sortiert werden soll. Wird nur ausgewertet, wenn das Attribut "sortKPath" nicht angegeben ist. Die Sortierreihenfolge kann mittels des Attributes "sortAscending" am ObjectList-Element oder am entsprechenden Column-Unterelement definiert werden (überschreibt die Einstellung von "sortColumn" an ObjectList).
initialSortLimit	Ganzzahl	deaktiviert	<i>Ab Version 3.8.9.</i> Anahl der Einträge in der Objektliste bis zu der eine automatisch Sortierung durchgeführt wird
resetSortOnReload	Boolesch	true	<i>Ab Version 3.8.9.</i> Gibt an, ob die vom Benutzer eingestellte Sortierreihenfolge in der Objektliste beim Reload der Seite verworfen (true) oder beibehalten werden soll (false).
resetFiltersOnReload	Boolesch	true	<i>Ab Version 3.9.31.</i> Gibt an, ob die vom Benutzer festgelegte Filterung in der Objektliste beim Reload der Seite verworfen (true) oder beibehalten werden soll (false).



pageSize	Ganzzahl		Gibt an, wieviele Tabellenzeilen auf einmal angezeigt werden sollen. Wird dieser Wert nicht angegeben, so werden standardmäßig 10 Zeilen angezeigt, der Benutzer kann aber zwischen 10, 20 und 30 Einträgen umschalten. Wird ein Wert für "pageSize" angegeben, so kann die Anzahl der angezeigten Einträge nicht mehr vom Benutzer verändert werden.
styleClass	String		Eine zusätzliche CSS Klasse für die Komponente. In jedem Fall wird die Klasse "objectListViewComponent" verwendet.
id	String		Die CSS ID, die für die Ausgabe der Komponente verwendet wird.
renderMode	String		Falls ein besonderer RenderMode für das Rendern der Komponente verwendet werden soll, kann dieser hier angegeben werden. Vordefinierte RenderModes: <ul style="list-style-type: none">• <i>simple</i> (ab 3.8): Nicht angezeigt werden das Menü zur Seitengrößenänderung und der Pager am Ende.• <i>small</i> (ab 3.8): Die Objektliste wird dann mit dem Label links und mit der Tabelle rechts dargestellt, d.h. ähnlich zu Attributen und Relationen.
clickableRow	Boolesch	true	Gibt an, ob beim Klick auf eine Zeile in der Objektliste die Detailseite des dazugehörigen Wissensnetz-Objektes geöffnet werden soll.
searchable	Boolesch	true	Gibt an, ob oberhalb der Spalten ein Eingabefeld zur Suche (Filterung) innerhalb der Einträge der Spalte angezeigt werden soll. Diese Einstellung kann durch das Attribut "searchable" an einem Columns-Unterelement überschrieben werden.



autoSearchMinLength	Ganzzahl	3	Die minimale Anzahl von Zeichen, bevor eine automatische Filterung der Tabelle durchgeführt wird. Der Wert "0" schaltet die automatische Suche ab.
autoSearchOnTab	Boolesch	true	<i>Ab Version 3.8.</i> Gibt an, ob die Filterung der Tabelle bei Verlassen einer Filtereingabe angestoßen werden soll.
initialLoad	Boolesch	true	<i>Ab Version 3.8.</i> Gibt an, ob die Einträge der ersten Seite initial geladen werden sollen oder nicht. In der Voreinstellung werden die Einträge geladen. Diese Einstellung macht dann Sinn, wenn es sich um sehr große Datenmengen handelt, die ohne Filtereinstellungen nur langsam laden.
inheritanceDisabled	Boolesch	true	<i>Ab Version 3.9:</i> Gibt an, ob die Objektliste nur die direkten Individuen bzw. Unterbegriffe des aktuellen Konzeptbegriffs anzeigen soll (true) oder ob alle Individuen bzw. Unterbegriffe angezeigt werden sollen (false).
optionsColumn	Ganzzahl		Ab Version 3.9.25 Definiert eine Spalte, in der Optionen zum Rendern der Spalte enthalten sind. Die Spalte muss einen String enthalten, der als JSON-Objekt interpretiert wird. Mögliche Optionen sind: <pre>{ "styleClass": "cssKlasseDerZeile" }</pre>

Unterelemente <Label> / <LabelKey> / <LabelKPath>

Diese Elemente können verwendet werden, um die Überschrift der Objektliste festzulegen. Eine feste Überschrift kann direkt mit dem Label-Element angegeben werden. Um eine übersetzte Überschrift anzugeben, kann ein Message-Key mit LabelKey angegeben werden. Um eine dynamische Überschrift anzugeben, kann mit LabelKPath ein KPath zur Bestimmung des Labels angegeben werden. LabelKey und LabelKPath können auch in Kombination verwendet werden, wobei das Ergebnis des KPath als erster Platzhalter bei der Ausgabe der mit LabelKey spezifizierten Message eingefügt wird.

Unterelement <RenderHint>

Über dieses Konfigurationselement können dem Renderer, der HTML-Quelltext generiert,



verschiedenen Parameter übergeben werden. Diese gelten immer für die Items, die in der Attributgruppe enthalten sind.

Name	Possible	Default	Beschreibung
itemSeparator	(String)	", "	Trennzeichen für mehrfach vorkommende Werte in einer Zelle

HINWEIS: Die Reihenfolge der Unterelemente muss in der XML-Definition mit der hier vorgegebene übereinstimmen.

Unterelement `<Column>`

Dieses Element dient dazu, eine einzelne Spalte zu konfigurieren. Für eine Spalte können neben einem speziellen RenderMode eigene RenderHints konfiguriert werden.

Attribute von `<Column>`:

Attribut	Mögliche Werte	Standardwert	Beschreibung
index	Ganzzahl		Der Index gibt an, welche Spalte mit den Unterelementen von <code><Column></code> konfiguriert werden sollen. Die Nummerierung beginnt mit 0.
styleClass	String		Zusätzliche Inhalte des Attributs "class", die bei allen HTML-TD-Elementen innerhalb dieser Spalte eingetragen werden sollen.
renderMode	String		Render-Mode, der beim Rendern der Zellen-Werte dieser Spalte gesetzt werden soll.
searchable	Boolesch		Gibt an, ob über dieser Spalte ein Eingabefeld zur Eingabe von Filterkriterien eingeblendet werden soll (true) oder nicht (false).
sortable	Boolesch		Gibt an, ob diese Spalte als Sortierkriterium für die Objektliste verwendet werden kann (true) oder nicht (false).
visible	Boolesch		Gibt an, ob diese Spalte versteckt werden soll (true) oder nicht (false).
sortAscending	Boolesch		Gibt an, ob aufsteigend (true) oder absteigend sortiert werden soll, wenn diese Spalte als Sortierkriterium ausgewählt ist (siehe auch Attribut "sortColumn" und "sortColumns" am Element "ObjectList").



wildcardsEnabled	Boolesch		Gibt an, ob bei Eingabe eines Filterkriteriums für diese Spalte (siehe Attribut "searchable") automatisch ein Wildcard-Zeichen (*) am Ende der eingegebenen Zeichenkette angefügt werden soll (true) oder nicht (false).
rowTopicLink (ab 3.8.20)	Boolesch	false	Gibt an, ob der Spalteninhalt als Link auf das Zeilenobjekt ausgegeben werden soll. Diese Einstellung hat nur Auswirkungen auf Zeichenkettenwerte. HTML-Sonderzeichen werden als sichtbare Zeichen ausgegeben.

Unterelement von `<Column>`: `<RenderHint>`

Mittels des `RenderHint`-Elements können zusätzliche Parameter für das Rendering der jeweiligen Spalte angegeben werden. Dies ist jedoch nur für angepasste Renderer interessant, da der Standard-Renderer keine speziellen Render-Hints unterstützt.

Mögliche Renderhints:

Name	Beschreibung	Beispiel
escapeHtml	Für Spalten, für die in der Konfiguration der Objektliste ein KScript konfiguriert wurde, kann hiermit festgelegt werden, dass die Skriptausgabe direkt als HTML eingefügt wird.	<pre><Column index="0"> <RenderHint name="escapeHtml" value="false"/> </Column></pre> <p>Hiermit wird der Inhalt der Spalte als HTML interpretiert. Damit sind eigene HTML-Links, Bilder und ähnliches möglich.</p>
showLink	Standardwert: <i>true</i> Spalten, die Relationen enthalten und für diese keinen Link anzeigen sollen, können mit diesem <code>RenderHint</code> versehen werden.	<pre><Column index="0"> <RenderHint name="showLink" value="false"/> </Column></pre>
minDigits	Standardwert: 3 Minimale Anzahl der ausgegebenen Nachkommastellen bei der Ausgabe von Fließkommazahlen.	<pre><Column index="0"> <RenderHint name="minDigits" value="2"/> </Column></pre>



maxDigits	Standardwert: - MaAnzahl der ausgegebenen Nachkommastellen bei der Ausgabe von Fließkommazahlen.	<pre><Column index="0"> <RenderHint name="maxDigits" value="2"/> </Column></pre>
maxLength (ab KP 3.8)	Standardwert: - Zeichenketten werden ggf. auf die angegebene Länge abgekürzt und mit "..." am Ende ausgegeben.	<pre><Column index="0"> <RenderHint name="maxLength" value="10"/> </Column></pre>
filterXss	Standardwert: false Gibt an, ob Zeichenketten vor der Ausgabe um evtl. enthaltene Cross-Site-Scripting-Bestandteile bereinigt werden sollen (z.B. Javascript). Wird dieser Render-Hint aktiviert, so kann dies die Darstellungsgeschwindigkeit der Objektliste stark verlangsamen.	<pre><Column index="0"> <RenderHint name="filterXss" value="true"/> </Column></pre>

Unterelement <ComponentActiontransformers>

Dieses Unterelement kann verwendet werden, um Actions im Header-Bereich der Objektliste anzubringen, z.B. um eine Objektliste als Excel-Datei zu exportieren.

Unterelement <ItemActiontransformers>

Dieses Unterelement kann verwendet werden, um Actions an jeder Zeile der Liste anzubringen. ActionTransformer, die hier konfiguriert werden, können seit Version 3.9.32 folgende Objekte in KPath-Ausdrücken verwenden.

Ausdruck	Bezeichnung
.	(Eingangsobjekt:) Objekt, das in der Zeile dargestellt wird.
var(kpContextTMD)	ID des Objektes, in dessen Kontext die Objektliste dargestellt wird. (z.B. Eingangsobjekt des "itemsKPath")



```
var(kpFocusTopic) DWD des Objektes, dessen Detailseite gerade angezeigt wird.
```

Beispielkonfiguration

Das folgende Beispiel-XML erzeugt an einem Konzept eine Objektliste mit den direkten Individuen:

```
<ObjectList itemsKPath="instances()" conceptInternalName="Person"
  tableName="Standard">
  <Label>Personen</Label>
</ObjectList>
```

Personen		
10 Einträge anzeigen		
Name	Wohnort	Sonstiges
Christof Luick		Person
Eva Hermann (nicht löschen/ ändern)	Berlin, Düsseldorf	Person
Fred Feuerstein		Person
Günther Netzer (bitte nicht ändern/löschen)		Person
Max Mustermann	Münster, Belfast, Afrika, Flensburg	Person
Test-Person 2		Person
Testperson		Person
Testperson 2		Person
Testperson 3		Person
Testperson für Dokumente (nicht löschen/ ändern)		Person

1 bis 10 von 11 Einträgen

Erster Zurück 1 2 Nächster Letzter

Objektliste aus der Beispielkonfiguration (siehe unten)

Konfiguration der Objektliste im Knowledge-Builder

Übersetzte Spaltentitel

Um dies zu erreichen, muss die Bezeichnung der Spalte das Prefix "msg:" enthalten, also z.B. "msg:objectlist.person.nameColumn". Ist das Prefix "msg:" gesetzt, dann wird der nachfolgende Teil als Message-Key interpretiert. In diesem Beispiel müsste der folgende Eintrag in den Message-Übersetzungen gemacht werden:

objectlist.person.nameColumn = Name der Person



Breite einer Spalte

Die Konfiguration der Spaltenbreite erfolgt mittels des Knowledge-Builders. Hier muss unter "Breite der Spalte festlegen" die Breite in % eingetragen werden.

Bilder in einer Spalte

Bilder werden von Haus aus nicht unterstützt. Eine Spalte kann das Ergebnis eines registrierten Skripts enthalten. Dieses kann einen HTML-Schipsel zurückliefern, also auch ein Tag.

7.1.2.1.12 Comments (ab Version 3.8)

Mittels des Elements "Comments" kann auf einer Detailseite ein Bereich zum Anzeigen, Anlegen und Löschen von Kommentaren eingeblendet werden. Das Element löst die in früheren KP-Versionen fest eingebaute Kommentarfunktion ab. Details hierzu werden im Kapitel "Konfiguration der Kommentarfunktion" (siehe Verweis in rechter Spalte) beschrieben.

Über die folgenden Attribute am Comments-Element kann die Erzeugung der Kommentar-Komponente konfiguriert werden:

Attribut	Möglich Werte	Standardwert	Beschreibung
styleClass	(String)		Zusätzliche CSS-Klassen, die diesem Comments-Element im HTML zugewiesen werden sollen. Standardmäßig werden die folgenden CSS-Klassen zugewiesen: - commentsViewComponent - objectListViewComponent - header - comments (Optional)
renderMode	(String)		Render-Mode, der bei der Auswahl des Renderers für die CommentsViewComponent berücksichtigt werden soll (dient zur Konfiguration spezieller Renderer in KP-Anpassungsprojekten) (Optional)
authorRelInternalName	(String)	commentedBy	Interner Name der Relation, die beim Kommentieren vom kommentierten Objekt aus zum Autor gezogen werden soll. (Optional)
newAutorRelTargetKPath	KPath-Ausdruck (String)	userInstance()	KPath-Ausdruck, der beim Anlegen eines Kommentars zur Bestimmung des Kommentar-Autors ausgewertet werden soll. (Optional)



tableName	(String)	kpComments	Name der Objektlistenkonfiguration, die zur Anzeige der Kommentare verwendet werden soll. (Optional)
commentsExpertQuery-Name	(String)		Name einer Expertensuche zur Berechnung der angezeigten Kommentar-Relationen, die anstelle einer Standardsuche verwendet werden soll. Die Standardsuche liefert alle Relationen mit dem im Attribut "authorRelInternalName" konfigurierten internen Namen zurück. Sofern die Expertensuche einen Frame-ID-Parameter mit dem Namen "topic" enthält, so wird dieser bei der Ausführung der Suche auf die Frame-ID des Kontext-Objektes (angezeigtes Topic) gesetzt. (Optional)
collapsed	(Boolean)	true	Gibt an, ob die Objektliste zur Anzeige der Kommentare initial zu- (true) oder aufgeklappt (false) angezeigt werden soll. (Optional)
collapsible	(Boolean)	true	Gibt an, ob bei der Objektliste zur Anzeige der Kommentare ein Link zum Zuklappen angezeigt werden soll (true) oder nicht (false). (Optional)

7.1.2.1.13 Darstellung von Metadaten

7.1.2.1.14 BoxArea (ab KP 3.9)

Mittels des Elements "BoxArea" kann auf einer Detailseite ein neuer Anzeigebereich für Boxen definiert werden. Die ID des Anzeigebereichs muss in dem Attribut "displayAreald" am BoxArea-Element konfiguriert werden. Die folgenden ID's sind vom Knowledge-Portal bereits für andere Box-Anzeigebereiche vergeben und dürfen daher nicht im displayAreald-Attribut angegeben werden (es sei denn, man möchte die Boxen aus diesen Anzeigebereichen explizit innerhalb einer ViewConfig einblenden):

- functionColumn
- contextBoxes
- aboveMainColumn
- mainColumn



- belowMainColumn

Damit Boxen in dem neu definierten Anzeigebereich eingeblendet werden, muss die ID des Anzeigebereichs in der Box-Konfiguration eingetragen werden. Für Boxen, die über Individuen in der KP-Konfiguration konfiguriert werden (MicroEdit, TopicDetail), muss hierzu das Auswahlattribut "Anzeigebereich" um die ID des neuen Anzeigebereichs ergänzt und das Attribut an der entsprechenden Box dann mit dem neuen Auswahlwert angelegt werden. Für Boxen mit begriffsbasierter Konfiguration (Context, KPath, etc.) muss am Box-Konfigurationsbegriff das Attribut "defaultDisplayArea" mit der ID des neuen Anzeigebereichs als Wert angelegt werden.

Hier nochmal ein Überblick über die möglichen Attribute am BoxArea-Element:

Attribut	Möglich Werte	Standardwert	Beschreibung
displayAreald	(String)		ID des Anzeigebereichs für Boxen (Pflichtfeld)
styleClass	(String)		Zusätzliche CSS-Klassen, die diesem Comments-Element im HTML zugewiesen werden sollen. Standardmäßig werden die folgenden CSS-Klassen zugewiesen: - commentsViewComponent - objectListViewComponent - header - comments (Optional)
renderMode	(String)		Render-Mode, der bei der Auswahl des Renderers für die CommentsViewComponent berücksichtigt werden soll (dient zur Konfiguration spezieller Renderer in KP-Anpassungsprojekten) (Optional)

7.1.2.1.15 SubConfig

Ab Version 3.9 kann über den SubConfig-Tag eine ViewConfig eingebunden werden:

```
<SubConfig [ resource="path" | configKey="key" ]
```

Das Attribut resource kann verwendet werden, um die Config aus einer Datei zu laden, die im Projekt hinterlegt ist. Diese Variante wird nur verwendet, wenn Entwicklern festverdrahtete ViewConfigs in Anpassungsprojekten hinterlegt haben.

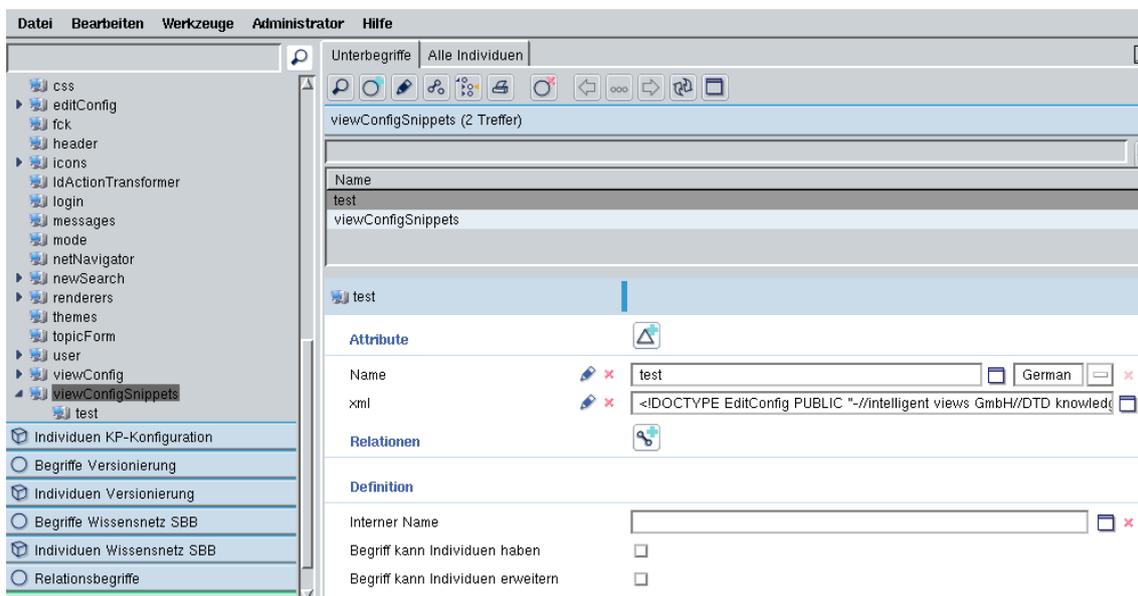


Das Attribut `configKey` kann verwendet werden, um die Config aus einem Konfigurations-Schlüssel zu laden. Dieser kann auch im Wissensnetz definiert sein. Für ViewConfigs wird dem Schlüssel noch das Prefix "viewConfigSnippets.", für EditConfigs "editConfigSnippets." vorangestellt, damit die Konfigurationen alle im gleichen Bereich der Config zu finden sind.

Beispiel:

```
<SubConfig configKey="text.xml" />
```

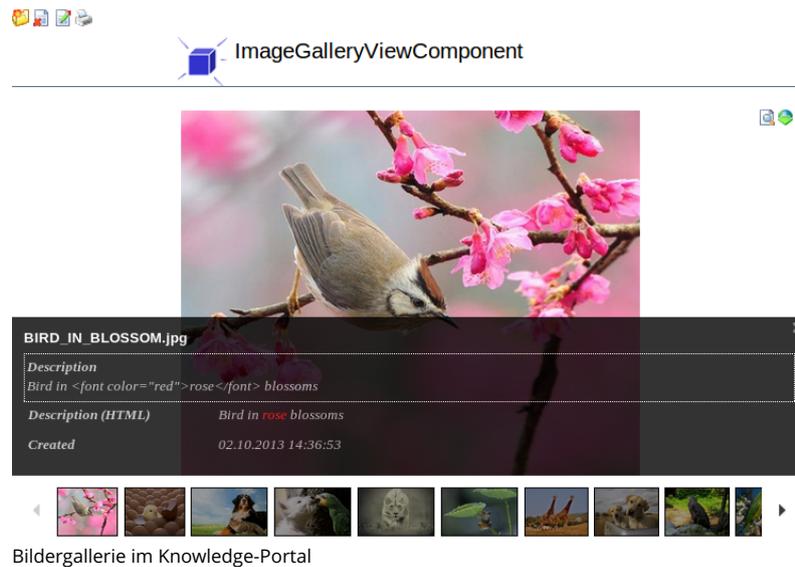
Wird dieser Ausdruck in einer ViewConfig verwendet, so wird die einzubindende Konfiguration unter "viewConfigSnippets.test.xml" gesucht, was in der Volume-Konfiguration dem Attribut "xml" am Begriff "viewConfigSnippets -> test" entspricht:



7.1.2.1.16 ImageGallery (ab KP3.9.21)

Einbinden von Bilder-Galerien

Über den `<ImageGallery>`-Tag können Bilder-Galerien mit Vorschaubildern und Überblendungseffekten in das Knowledge-Portal eingebunden werden. Als Grundlage dient die JavaScript-Bibliothek <http://galleria.io>. Die Daten aus dem Wissensnetz werden von einer Objektliste geliefert, bei der Bilder in verschiedenen Größen in speziell benannten Spalten konfiguriert sind.



Konfiguration der Objektliste

Die Objektliste muss so konfiguriert werden, dass pro Bild eine Zeile erzeugt wird. Dies kann z.B. über Bild-Individuen geschehen, die mehrere Bild-Datei-Attribute für die verschiedene Bild-Größen haben. Die folgenden Spalten sind vorgegeben:

Spalte	Attribut-Typ	Beschreibung
small	Datei	Vorschaubild in der Thumbnail-Ansicht (keine Darstellung unter dem großen Bild) ACHTUNG: Diese Spalte fällt im KP 3.8.28 weg und wird statt dessen über das Attribut "thumbnailSizeSmall" im ViewConfig-XML konfiguriert.
large	Date	Großes Vorschaubild für den Haupt-Bildbereich ACHTUNG: Diese Spalte fällt im KP 3.8.28 weg und wird statt dessen über das Attribut "thumbnailSizeLarge" im ViewConfig-XML konfiguriert.
original	Date	Originalbild. Dieses Bild wird im Vollbildmodus verwendet und kann über den Download-Knopf heruntergeladen werden
name	Zeichn-kette	Name des Bildes

Alle weiteren Spalten werden als Beschreibung im Info-Bereich der Galerie ausgegeben.

View-Config

Der <ImageGallery>-Tag hat eine ähnliche Syntax wie der <ObjectList>-Tag mit folgenden Einschränkungen:

- Das Attribut "clickableRow" hat keine Auswirkung
- RenderHints für die oben genannten Spezial-Spalten haben keine Wirkung.



- ComponentActionTransformer sind noch nicht implementiert.
- Sortierung und Filter sind nicht implementiert.
- **Das Attribut "pageSize" hat keine Auswirkung. Die Komponente ist momentan nicht in der Lage, Bilder dynamisch nachzuladen. Es wird sofort die gesamte Objektliste ausgelesen.**

Zusätzlich zu den Attributen des ObjectList-Elements können am XML-Element "ImageGallery" die folgenden Attribute angegeben werden:

Attribut	Attribut-typ	Beschreibung
dummyImagePath (ab KP-Version 3.8.28)	Zeichenkette	Relativer Pfad zu einer Bilddatei, die angezeigt werden soll, wenn das eigentlich anzuzeigende Thumbnail nicht geladen werden kann (nicht vorhanden oder Zugriff verweigert). Standardwert: gfx/imageGallery/notFound.png
height (ab KP-Version 3.8.28)	Ganzzahl o. Fließkommazahl	Gibt die Höhe des Gallery-Widget an. Ein Wert ≥ 2 wird als feste Höhe in Pixel interpretiert. Wird ein Wert kleiner als 2 angegeben, so wird der Wert als Seitenverhältnis interpretiert. Die Höhe wird in diesem Fall ermittelt, indem der Wert mit der aktuellen Breite des Gallery-Widgets multipliziert wird. Da das Gallery-Widget so konfiguriert ist, dass es immer die maximal verfügbare Breite im Browserfenster einnimmt, behält das Widget dadurch beim Verkleinern des Browserfensters immer dasselbe Seitenverhältnis. Standardwert: 0.5625 (9/16)
loadThumbnailsLazy (ab KP-Version 3.8.28)	Boolesch (true/false)	Gibt an, ob die Vorschaubilder der kleinsten Größe erst beim Blättern in der Vorschauliste "on-demand" geladen werden sollen (true) oder direkt nach der Initialisierung der Gallery-Komponente (false). Standardwert:false



loadThumbnailChunkSize (ab KP-Version 3.8.28)	Ganzzahl	Gibt an, wie viele Vorschaubilder der kleinsten Größe geladen werden dürfen, bevor dem Server eine kurze Pause gegönnt wird, um die Verarbeitung anderer Requests zu ermöglichen. Diese Einstellung wirkt sich nur aus, wenn "loadThumbnailsLazy" auf "true" gesetzt ist. Standardwert: 10
theme (ab KP-Version 3.8.28)	Zeichenkette	Gibt an, welches Galleria-Theme vom Gallery-Widget verwendet werden soll. Standardwert: js/galleria-iviews-theme/galleria.iviews.js
thumbnailSizeLarge (ab KP-Version 3.8.28)	Zeichenkette	Gibt an, welche Thumbnail-Größe zur Anzeige der vergrößerten Ansicht beim Anklicken eines Bildes in der Liste verwendet werden soll. Der K-Infinity-Image-Processor muss so konfiguriert sein, dass er die Thumbnails in der passenden Größe erzeugt (siehe Link in der rechten Spalte). Standardwert: 640x480
thumbnailSizeOriginal (ab KP-Version 3.8.28)	Zeichenkette	Gibt an, welche Thumbnail-Größe zur Anzeige der Originalversion eines Bildes (Anklicken der vergrößerten Ansicht) verwendet werden soll. Der K-Infinity-Image-Processor muss so konfiguriert sein, dass er die Thumbnails in der passenden Größe erzeugt (siehe Link in der rechten Spalte). Standardwert: Kein Wert konfiguriert - es wird das Originalbild zurückgeliefert.
thumbnailSizeSmall (ab KP-Version 3.8.28)	Zeichenkette	Gibt an, welche Thumbnail-Größe zur Anzeige der kleinsten Vorschaubilder in der Liste verwendet werden soll. Der K-Infinity-Image-Processor muss so konfiguriert sein, dass er die Thumbnails in der passenden Größe erzeugt (siehe Link in der rechten Spalte). Standardwert: 120x80

Beispiel

Im oben genannten Beispiel wurde folgende Objektliste mit dem Namen "ImageGallery" für Bild-Individuen konfiguriert:



Spalte	Beschreibung
small	Datei-Attribut, das Bilder in maximal 100 Pixeln Höhe enthält. ACHTUNG: Diese Spalte fällt im KP 3.8.28 weg (siehe oben)
large	Datei-Attribut, das Bilder in maximal 640x480 Pixeln Größe enthält ACHTUNG: Diese Spalte fällt ab KP 3.8.28 weg (siehe oben)
original	Datei-Attribut, Originalbild (z.B. 1600x1200 Pixel)
name	Namens-Attribut
Description	Zeichenketten Attribut "description" (enthält HTML)
Description (HTML)	Zeichenketten Attribut "description" (enthält HTML, wie in voriger Spalte).
Created	Zeitstempel-Attribute

Das folgende `<ImageGallery>`-Element erzeugt die Galerie:

```
<ImageGallery forInstances="true" inheritanceDisabled="false" pageSize="10" >
  <Column index="4" styleClass="textBelowLabel" />
  <Column index="5">
    <RenderHint name="escapeHtml" value="false" />
  </Column>
  <ItemActionTransformers>
    <ActionTransformer
      class="com.iviews.knowledgeportal.actions.transformer.IdActionTransformer">
      <Property name="actionId" value="galleria.imageDetails" />
      <Property name="actionClass"
        value="com.iviews.knowledgeportal.actions.action.ShowDmidAction"/>
      <Property name="actionForward" value="viewdmid" />
    </ActionTransformer>
  </ItemActionTransformers>
</ImageGallery>
```

Spalte 4 ist so konfiguriert, dass die CSS-Klasse "textBelowLabel" gesetzt wird. Diese ist im Standard so eingestellt, dass der Text der Spalte die gesamte Breite einnimmt und unter dem Label dargestellt wird.

Spalte 5 ist mit dem RenderHint "escapeHtml=false" konfiguriert, so dass der HTML-Code der Spalte direkt dargestellt wird.

Der ItemActionTransformer zeigt beispielhaft, dass auch Aktionen für Bilder konfiguriert werden können. Diese Konfiguration öffnet die Detailseite des Bild-Individuums

7.1.2.1.17 Breadcrumb ViewComponent (ab KP 3.9)

Einbinden von Breadcrumb

Über den `<Breadcrumb>`-Tag kann eine Leiste eingeblendet werden, mit der die aktuelle



Position in einer Hierarchie oder einer Reihe von Schritten dargestellt werden kann.

View-Config

Die folgenden Attribute werden unterstützt:

Attribut	Attribut-typ	Beschreibung
itemsKPath	String	KPath, der eine Menge von Topics berechnet, die die Schritte definieren. Die Menge gewöhnlich eine Sortierung haben, da sonst die Darstellung nicht gleichbleibend ist. Oft ist es nützlich, hier eine Abkürzungsrelation zu verwenden.
itemTooltipKPath	String	Optional. Ein KPath, der den Tooltip für jeweils einen einzelnen Schritt berechnet. Wenn mehr als nur das Label des Schrittes dargestellt werden soll, kann dieser KPath konfiguriert werden. Standardwert: label()
itemDetailsKPath	String	<i>Optional.</i> Ein KPath, der eine Detailangabe für jeweils einen Eintrag bestimmt. Ist dieser KPath konfiguriert, so wird die Breadcrumb-Leiste zweizeilig dargestellt.
appendCurrentTopic	true / false	Gibt an, ob das aktuelle Kontextobjekt als letztes Element des Pfads angehängt werden soll oder nicht. Bei "true" wird es angehängt und ohne Link dargestellt. Standardwert: true

Beispiel

```
<Breadcrumb  
  itemsKPath=~$previousSteps$/target()/sort(@$stepOrder$/rawValue(), true)"  
  itemTooltipKPath="label() + ' (' + concept() + ')'"  
  itemDetailsKPath="concept()/@$explanation$/value()/singleElement()"  
  appendCurrentTopic="true"  
>
```

7.1.2.1.18 TemplateReport (ab Version 3.9.35)

Element <TemplateReport>



Dieses Element funktioniert wie das Report Element, mit dem Unterschied, daß die Reportausgabe als Eingabe (Context) für ein Template genommen wird. Die Ausgabe des Templates wird dann auf der Detailseite angezeigt.

Als Template kann Mustache oder Handlebars genommen werden. Daher muss die Ausgabe des Reports im JSON-Format sein.

Die Konfiguration ist ähnlich zu der des Report Elements. Im Unterschied dazu sind *reportFolderName* und *reportName* optional. Zusätzlich muss noch ein Handlebars Template konfiguriert werden:

Zusätzliche Attribute zu <Report>

Attribut	Mögliche Werte	Standardwert	Beschreibung
templateKPath	(String)		Ein KPath, der einen String als Template zurück liefert. Entweder muss dieses Attribut oder das Unterelement <Template> konfiguriert werden.

Unterelement <Template>

Alternativ zum *templateKPath* Attribut kann das Template auch über einen Unterelement spezifiziert werden. Um XML Konflikte zu vermeiden, empfiehlt es sich den Inhalt in einem CDATA Block einzuschließen.

```
<TemplateReport>
  ...
  <Template>
    <![CDATA[
      <b>{{name}}</b>
    ]]>
  </Template>
  ...
```

In den Templates können folgende Handlebar-Helper verwendet werden:

Helper	Beschreibung	Parameter
<code>{{action dmid-Property 'transformerKey'}}</code>	Fügt eine konfigurierte Action in der Ausgabe ein. S. auch Beschreibung ReportActionTransformers	<i>dmidProperty</i> : DMID eines Topics für das die Action erstellt werden soll <i>transformerKey</i> : Key über den ein konfigurierter ActionTransformer identifiziert werden kann



<code>{{msg 'messageKey' param1 paramN}}</code>	<p>Fügt eine über messages konfigurierte Übersetzung in der Ausgabe ein.</p> <p>S. auch Kapitel "Texte konfigurieren"</p>	<p><i>messageKey</i>: Key über den die Message identifiziert wird</p> <p><i>paramN</i>: Beliebige weitere Parameter die in der Message verwendet werden können</p>
---	---	--

Parameter können entweder eine Eigenschaft des aktuellen Kontext-Objektes sein oder statisch angegeben werden (Wert in Hochkommas).

Existiert eine Eigenschaft die den gleichen Namen wie ein Helper hat muss auf die Eigenschaft über *this.eigenschaft* zugegriffen werden:

```
{{this.action}}: {{action id 'showTopic'}}
```

Unterelement <ReportActionTransformers>

Über das Element ReportActionTransformers können ActionTransformer konfiguriert werden, die dann im Template verwendet werden können. Die Konfiguration ist analog zu anderen ActionTransformer. Zusätzlich muss noch ein Attribute *key* am ActionTransformer angegeben werden über den der Transformer identifiziert wird:

```
<TemplateReport>  
...  
  <ReportActionTransformers>  
    <ActionTransformer key="showTopic" class="com.iviews.knowledgeportal.actions.transformer...>  
      <Property name="actionId" value="viewdmid" />  
      <Property name="actionClass" value="com.iviews.knowledgeportal.actions.action.Download...>  
    </ActionTransformer>  
  </ReportActionTransformers>  
...</pre>
```

Im Template kann dann über einen Handlebars Helper die Action eingebettet werden:

```
<TemplateReport>  
...  
  <Template>  
    <![CDATA[  
      {{action id 'showTopic'}}  
    ]]>  
  </Template>  
...</pre>
```

Beispielkonfiguration

In diesem Beispiel werden alle Individuen eines Begriffs in einer Aufzählungsliste dargestellt. Für jedes Individuum wird eine Action erzeugt. Zusätzlich wird eine übersetzte Überschrift aus der Message-Konfiguration erzeugt.

Komponentenkonfiguration

```
<TemplateReport folderName="reportFolderExternalId"  
  reportName="projectsReport"  
  initialTopicsKpath="//$projects$/instances()/sort(name(), true)">
```



```
        escapeHtml="false">
<Label>Projekte</Label>

<ReportActionTransformers>
  <ActionTransformer key="showTopic" class="com.iviews.knowledgeportal.actions.transform
    <Property name="actionId" value="viewdmid" />
    <Property name="actionClass" value="com.iviews.knowledgeportal.actions.action.Down
  </ActionTransformer>
</ReportActionTransformers>

<Template>
  <![CDATA[
    <h1>{{msg 'exampletemplate.title'}}</h1>
    <ul>
      {{#projects}}
        <li>{{name}} {{action id 'showTopic'}}</li>
      {{/projects}}
    </ul>
  ]]>
</Template>

</TemplateReport>
```

Report

Das Element `<Path path=".">` bezieht sich auf die Menge der Eingabeobjekte.

```
<Script output="json">
  <JSONObject>
    <JSONProperty name="projects">
      <JSONArray>
        <Path path=".">
          <Each>
            <JSONObject>
              <JSONProperty name="name" value="name()" />
              <JSONProperty name="id" value="id()" />
            </JSONObject>
          </Each>
        </Path>
      </JSONArray>
    </JSONProperty>
  </JSONObject>
</Script>
```

Erzeugt wird diese Ausgabe:

Alle Projekte

- Projekt 1 
- Projekt 2 
- Projekt 3 



Rekursion

Auch Rekursion lässt sich mit Templates bewerkstelligen. Dafür können die `#block` und `#partial` Helper verwendet werden:

```
{
  "name": "1",
  "children": [
    {
      "name": "1.1",
      "children": [{ "name": "1.1.1",
    }]
    }
  ]
}
```

```
{{#partial "node"}}
  {{this.name}}

  {{#each this.children}}
    {{#block "node"}}{{/block}}
  {{/each}}
{{/partial}}

{{#each children}}
  {{#block "node"}}{{/block}}
{{/each}}
```

Dabei ist darauf zu achten das über den Qualifier *this* auf die entsprechenden Eigenschaften zugegriffen wird da sonst, falls die Eigenschaft nicht vorhanden ist, im darüber liegenden Kontext gesucht wird und dies zu Endlosschleifen führen kann.

7.1.2.1.19 Condition (ab Version 3.9.34)

Das Condition Element überprüft einen KPath ausdruck und führt die enthaltenen Unterelemente nur aus, falls dieser Ausdruck wahr ist.

Das Element selbst erzeugt keine Ausgabe. Nur die Unterelemente werden ausgegeben.

Attribute für den Tag `<Condition>`

Name	Typ	Benötigt	Beschreibung
enabledKPath	KPath	ja	Ein KPath der einen Boolean zurück gibt. Es muss darauf geachtet werden das die Bedingung unter allen Umständen einen booleschen Wert zurück liefert da es sonst zu einem Fehler kommt.



Beispielkonfiguration

In diesem Beispiel werden die Attribute nur angezeigt wenn das Topic einen Attribut mit internen Namen "showAttributes" hat und dieses den Wert *Wahr* hat.

Komponentenkonfiguration

```
...
<Condition enabledKPath="@$showAttributes$/rawValue()/singleElement() = true">
  <AllAttributes />
</Condition>
...
```

7.1.2.2 Unterelemente

7.1.2.2.1 Label

Dieser Tag dient zur Benennung einer Gruppe. Wird der Tag `<Label>` verwendet, wird dessen Inhalt als unübersetzter String ausgegeben.

Um Formatierungsfehler zu vermeiden, wird ein leerer `<Label>`-Tag durch ` ` ersetzt. Wenn Sie das Label ausblenden möchten, verwenden Sie bitte den Renderhint "hideLabel"

```
<RenderHint name="hideLabel" value="true" />
```

7.1.2.2.2 LabelKey

Genau wie `<Label>` wird `<LabelKey>` zur Benennung von Gruppen verwendet. Als Text wird hier jedoch der Schlüssel für die Übersetzung in der `msg.properties` eingetragen. Sie müssen also den hier eingetragenen Schlüssel in folgenden Attributen übersetzen:

```
kpConfig > messages > msg*
```

7.1.2.2.3 InternalName

Dieser Tag spezifiziert, dass nach dem internen Namen gesucht werden soll, der im Body dieses Tags steht.

7.1.2.2.4 KPath

Dieser Tag beinhaltet in seinem Body eine KPath, die das gewünschte Objekt referenziert. Bitte beachten Sie, dass die unterschiedlichen Tags unterschiedliche KPath Ergebnisse erwarten.

CustomGroup, Extension:

Erwarten ein Topic als Ergebnis. Wird eine Relation abgefragt, muss am Ende immer das "target()" stehen, das auf das Ziel-Topic verweist.

Attribute:

Erwartet ein Attributobjekt. Der KPath muss also immer mit `@$attName$` enden.

Relation:

Erwartet eine Relation. Der KPath endet also mit `~$relationName$`



7.1.2.2.5 RenderHint

Wird zum Übergeben von Parametern an die Unterelemente benutzt. Die Liste der verfügbaren Parameter finden Sie bei den jeweiligen Tags.

Der Renderhint ist immer als *letztes Element* anzugeben und hat folgendens Format:

```
<Renderhint name="paramName" value="paramWert" />
```

7.2 Ausblenden bestimmter Attribute und Relationen

7.2.1 Konfiguration von versteckten Attributen und Relationen (bis KP 3.7)

7.2.1.1 Attribute ausblenden

Attribute in der Detailansicht ausblenden

Um bestimmte Attribute aus der Detailansicht für Begriffe auszublenden, kann das Attribut vom Typ Zeichenkette "*hiddenAttributes*" verwendet werden, das beim Konfigurationsbegriff "*kpConfig*" definiert ist und auch bei diesem Begriff bearbeitet wird. Als Wert für dieses Attribut können mehrere interne Namen angegeben werden, die durch ein Leerzeichen voneinander getrennt werden müssen, also z.B.:

```
hiddenAttributes=RDF-about phoneNumber
```

In den internen Namen kann auch das Zeichen "*" als Platzhalter angegeben werden, um festzulegen, dass alle internen Namen mit einem bestimmten Präfix bzw. Suffix ausgeblendet werden sollen. Beispiel:

```
hiddenAttributes=RDF* phoneNumber *hidden
```

KP-Konfiguration	
Attribute	
hiddenAttributes	<input type="text" value="modified*"/>
hiddenRelations	<input type="text" value="comments commentedBy hasInternalDocument* employs has_"/>
Name (Deutsch)	German <input type="checkbox"/> univ. <input type="text" value="KP-Konfiguration"/>

Attributkonfiguration für KP-Konfiguration

Attribute im Pflegefenster ausblenden

Um Attribute aus den Pflegefenstern anwendungsweit auszublenden, muss das Attribut "*hiddenAttributes*" beim Konfigurationsbegriff "*kpConfig > edit*" verwendet werden. Es gelten die gleichen Regeln, wie beim Attribut für die Detailseite.



edit	
Attribute	A*
hiddenAttributes	modified*
hiddenRelations	internal*
Name (Deutsch)	German <input type="checkbox"/> univ. <input type="checkbox"/> edit

Attributkonfiguration für edit

7.2.1.2 Relationen ausblenden

Um bestimmte Relationen aus der Detailansicht von Begriffen oder Individuen auszublenden, kann das Attribut (Typ Zeichenkette) "*hiddenRelations*" verwendet werden, das bei "*kpConfig*" definiert ist und auch im Schema dieses Begriffs bearbeitet wird.

KP-Konfiguration	
Attribute	A*
hiddenAttributes	modified*
hiddenRelations	comments commentedBy hasInternalDocument* employs has_
Name (Deutsch)	German <input type="checkbox"/> univ. <input type="checkbox"/> KP-Konfiguration

Attributkonfiguration für KP-Konfiguration

Die Eigenschaft kann genauso konfiguriert werden wie die Eigenschaft "*hiddenAttributes*" zum Ausblenden von Attributen. Es müssen dabei nur die internen Namen der Relationen angegeben werden, die ausgeblendet werden sollen.

Relationen, die in der Pflege ausgeblendet werden sollen, werden analog beim "*kpConfig* > *edit*"-Begriff angegeben.



edit	
Attribute	A*
hiddenAttributes	modified*
hiddenRelations	internal*
Name (Deutsch)	German <input type="checkbox"/> univ. <input type="checkbox"/> edit

Attributkonfiguration für edit

7.2.2 Konfiguration von versteckten Attributen und Relationen (ab KP 3.8)

Die Konfiguration von Attributen und Relationen, die auf Detailseiten und Pflegeseiten generell nicht angezeigt werden sollen, findet ab der Knowledge-Portal-Version 3.8 über die Rechtssystem-Konfiguration statt.

Über den Rechtssystem-Operator "KP: Auf Detailseite anzeigen" (int. Name "visibleForView") kann gesteuert werden, ob eine Eigenschaft auf der Detailseite angezeigt werden soll, über den Operator "KP: Auf Pflegeseite anzeigen" (visibleForEdit) wird die Anzeige von Eigenschaften auf der Pflegeseite gesteuert. Für die Entscheidung zur Anzeige einer Eigenschaft wird eine Rechteprüfung von Eigenschaftsbegriffen durchgeführt, d.h. in der Rechtssystem-Konfiguration dürfen nur solche Filter verwendet werden, die unabhängig von konkreten Eigenschaften bzw. Topics der Eigenschaften arbeiten (also reine Eigenschaftsfilter und Expertensuchen-Filter, die sich ausschließlich auf den Eigenschaftsbegriff und/oder auf das Benutzer-Objekt beziehen).

Die Rechtekonfiguration für die Anzeige von Eigenschaften muss im Rechtssystem vor allen anderen Operationen konfiguriert werden, es sei denn es soll eine unterschiedliche Konfiguration für verschiedene Benutzergruppen konfiguriert werden.

Beim Einrichten bzw. Upgrade der KP-Wissensnetzkomponente für das KP 3.8 im Admin-Tool werden automatisch Rechtssystem-Filter für die beiden oben beschriebenen Rechtssystem-Operationen direkt unterhalb des Rechtssystem-Wurzelordners eingefügt. Bestehende Einstellungen zu ausgeblendeten Eigenschaften aus älteren KP-Konfigurationen (Attribute "hiddenAttributes" und "hiddenRelations") werden beim Upgrade automatisch in die Rechtssystem-Konfiguration übernommen (siehe unten).

Der Rechtssystem-Filter für die anzuzeigenden Eigenschaften auf der Detailseite ist standardmäßig als Negativliste angelegt, d.h. Eigenschaften, die nicht angezeigt werden sollen, müssen zu einem Eigenschaftsfilter hinzugefügt werden. Die ausgeblendeten Eigenschaften aus der KP 3.7-Konfiguration (interne Namen aus den Attributen "hiddenAttributes", "hiddenRelations", "user.hiddenAttributes" und "user.hiddenRelations") werden beim Upgrade automatisch zu dem Eigenschaftsfilter hinzugefügt.

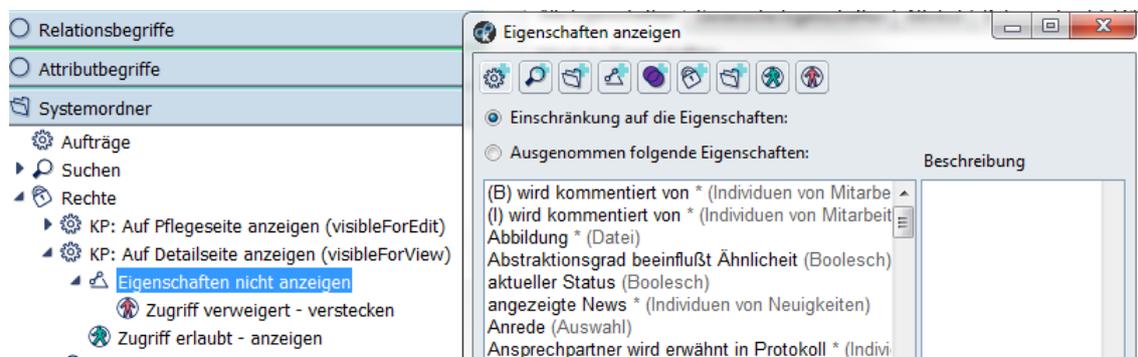
Der Rechtssystem-Filter für die anzuzeigenden Eigenschaften auf der Pflegeseite ist standardmäßig als Positivliste angelegt. Es müssen also alle Eigenschaften, die auf der Pflegeseite angezeigt werden sollen, zu einem Eigenschaftsfilter hinzugefügt werden. Dies ergibt sich daraus, dass in früheren KP-Versionen generell nur Eigenschaften mit internen Namen auf der Pflegeseite angezeigt wurden. Beim Anlegen des Eigenschaftsfilters werden daher automatisch alle Eigenschaften mit internen Namen abzüglich der versteckten Eigenschaften



aus der KP-3.7-Konfiguration (Attribute "edit.hiddenAttributes", "edit.hiddenRelations", "user.edit.hiddenAttributes" und "user.edit.hiddenRelations") zu dem Eigenschaftsfilter hinzugefügt.

Wenn also eine Eigenschaft in der EditConfig nicht direkt über ein Attribute- oder Relation-Element sondern über ein AllAttributes- bzw. AllRelations-Element in die Pflegeseite eingebunden werden soll, so muss man darauf achten, dass die betreffende Eigenschaft in der Positivliste im Rechtesystem eingetragen ist.

TIPP: Beim Entfernen einer Eigenschaft aus einem der Eigenschaftsfilter kann es vorkommen, dass die betreffenden Eigenschaften dennoch angezeigt (Pflgeseite) bzw. auf nicht angezeigt (Detailseite) wird. Dies hängt meist damit zusammen, dass noch eine übergeordnete Eigenschaft (Oberbegriff der Eigenschaft) bzw. eine Haupt-Eigenschaftsbegriff (Benutzerrelation, Abkürzungsrelation, Wurzelbegriff für Attribute/Relationen) in der entsprechenden Liste eingetragen ist und der Eigenschaftsfilter alle untergeordneten Eigenschaften mit einbezieht.



7.3 Trainingsdokumente und Ähnlichkeitssuche (Brainbot)

7.3.1 Beschreibung

Wenn im Knowledge-Portal die Detailansicht eines Begriffs geöffnet wird, der im Admin-Tool als Brainbot-Themen-Begriff ("Container") konfiguriert ist, so werden in der Detailansicht immer die sog. Primärdokumente ("Trainingsdokumente") zum jeweiligen Thema angezeigt.

Außerdem wird abhängig davon, wie viele Trainingsdokumente mit einem Thema verknüpft sind, entweder eine Brainbot-Volltextsuche (siehe Abb. 1) oder eine Brainbot-Ähnlichkeitssuche (siehe Abb. 2) ausgeführt, um Dokumente anzuzeigen, die mit dem jeweiligen Thema zu tun haben.



The screenshot shows the Alkyl interface with a search for 'Alkyl'. The header includes 'poly-topic' and the Alkyl logo. Below the header, there are two sections: 'Zentrale Dokumente' and 'Dokumentsuche "Alkyl"'. The 'Zentrale Dokumente' section lists two items: 'Alkyl unloading Pernis' and 'Alkyl unloading practices', both with downward arrows. The 'Dokumentsuche "Alkyl"' section lists seven items, each with an upward arrow: '050413 Spheripol Summary SRichard.ppt', 'Kopie von 050413 Spheripol Summary SRichard.ppt', 'Alkyl unloading Pernis', 'Answer from Terni', 'Alkyl unloading practices', 'DF301 incident.pdf', and 'Alkyl unloading Ferrara'.

Abb. 1: Primärdokumente und Ergebnisse der Brainbot-Volltextsuche

The screenshot shows the Alkyl interface with a similarity search for 'Alkyl'. The header includes 'poly-topic' and the Alkyl logo. Below the header, there are two sections: 'Zentrale Dokumente' and 'Ähnliche Dokumente'. The 'Zentrale Dokumente' section lists four items, each with a downward arrow: '050413 Spheripol Summary SRichard.ppt', 'Alkyl unloading Ferrara', 'Alkyl unloading Pernis', and 'Alkyl unloading practices'. The 'Ähnliche Dokumente' section lists six items, each with an upward arrow: 'Alkyl unloading Pernis', '050413 Spheripol Summary SRichard.ppt', 'Kopie von 050413 Spheripol Summary SRichard.ppt', 'PSNF122 Petroken DEAC Cylinder Piping Rupture', 'Varenne's practices', and 'Answer from Terni'.

Abb. 2: Primärdokumente und Ergebnisse von Brainbot-Ähnlichkeitssuche

7.3.2 Schwellwert für Anzahl der Trainingsdokumente

In der Datei "*settings.properties*" kann der Schwellwert für die Anzahl der Trainingsdokumente festgelegt werden, ab dem die Ähnlichkeitssuche ausgeführt wird.



Dazu muss das Attribut (Attributtyp Ganzzahl) "*wellTrainedThreshold*" am Konfigurationsbegriff "*kpConfig > documents*" auf den entsprechenden Wert (normalerweise 4) gesetzt werden. Es kann sinnvoll sein, den Wert zu erhöhen, wenn man feststellt, dass die Ähnlichkeitssuche bei zu wenigen Trainingsdokumenten keine guten Treffer liefert.

7.3.3 Konfiguration der Ähnlichkeitssuche

Der Name der Ähnlichkeitssuche, die ausgeführt wird, wenn ausreichend viele Trainingsdokumente mit einem Thema verknüpft sind, kann am Konfigurationsbegriff "*kpConfig > search*" unter dem Textattribut "*similarDocuments*" hinterlegt werden.

7.3.4 Konfiguration der Volltextsuche

Die Suche, die ausgeführt wird, wenn zu wenige Trainingsdokumente mit einem Themenbegriff verknüpft sind, ist die gleiche Suche, die auch bei der Verwendung des normalen Suchfelds im Knowledge-Portal als Volltextsuche ausgeführt wird. Der Name der Suche kann daher, wie im Kapitel "Konfiguration der Suchen" beschrieben, über die entsprechende Eigenschaft konfiguriert werden. Die Ergebnisse der Suche werden in einem Fenster mit dem Titel "Dokumentensuche" angezeigt. Bei der Ausführung der Volltextsuche wird standardmäßig der Name des gerade dargestellten Begriffs als Suchbegriff verwendet. Mittels eines Eingabefelds in der Titelzeile des Fensters kann der Benutzer den Suchbegriff aber auch abändern und die Suche erneut ausführen.

Mittels des Attributs (Attributtyp Ganzzahl) "*limit*" am Konfigurationsbegriff "*kpConfig > box > relatedFulltextHits*", lässt sich die Anzahl der angezeigten Dokument-Treffer für diese Suche limitieren. Standardmäßig werden unbegrenzt viele Treffer angezeigt. Wird das Limit auf 0 gesetzt, so wird das Fenster nicht mehr angezeigt. Das Limit wird nur berücksichtigt, wenn die Volltextsuche beim Öffnen der Detailseite automatisch ausgeführt wird. Bei manueller Ausführung der Suche wird das Limit nicht berücksichtigt.

TODO: Tabellenexport für Default-Suchbegriffe beschreiben

7.4 Kontextsuchen / Kontextboxen

7.4.1 Expertensuchergebnisse in Kontextboxen anzeigen

7.4.1.1 Beschreibung

Das Knowledge-Portal kann so konfiguriert werden, dass beim Aufruf der Detailansicht von Begriffen eine Reihe von Expertensuchen ausgeführt wird und die Ergebnisse dieser Suchen rechts von der Attribut- und Relationsübersicht in kleinen Fenstern (den sog. Kontextboxen) angezeigt werden. Dies erlaubt, abhängig vom jeweiligen Anwendungskontext und vom gerade betrachteten Begriff, bestimmte verwandte oder relevante Begriffe im Wissensnetz anzuzeigen.



The screenshot shows the KnowledgePortal interface. At the top, there is a search bar with a 'Starten' button and a 'Topics' dropdown. The main content area displays search results for 'TEAL', categorized as 'Co-catalyst, substance'. The results are organized into a table with columns for 'abbreviation', 'standard', 'synonym', 'is involved in', 'related to', and 'to be considered'. The 'standard' column lists 'TEAL ratios', 'TEAL cylinder', 'TEAL flow', 'teal pump', 'TEAL refrigerator', 'TEAL scrubber', 'TEAL seal pot', and 'TEAL supply'. The 'related to' column lists 'TEAL ratios', 'TEAL cylinder', 'TEAL flow', 'teal pump', 'TEAL refrigerator', 'TEAL scrubber', 'TEAL seal pot', and 'TEAL supply'. The 'to be considered' column lists 'decomposition' and 'TEAL Fire protection'. On the right side, there are sections for 'Autoren (10)', 'Organisationen (8)', and 'Ähnliche Themen'. The 'Ähnliche Themen' section lists '20 S/L - Teal, Donor, Catalyst (incl. activity / yield)', 'None', 'cylinder', 'fire protection', 'hoses', 'HSE issues', 'Mary Pergande', 'TEALHOSE.jpg', and 'TEAL unloading'.

Kontextboxen

7.4.1.2 Definition im Wissensnetz

Die Verknüpfung von Begriffen mit Expertensuchen erfolgt im Wissensnetz über ein Attribut mit dem internen Namen "*contextSearchFolderName*" (Attributtyp Zeichenkette), das an einem Konzeptbegriff angebracht werden muss. Als Wert dieses Attributs muss die externe ID eines Ordners gesetzt werden, der als Unterordner die Expertensuchen enthält, die zu dem jeweiligen Begriff ausgeführt werden sollen.

Für Kontextboxen, die auf den Pflegeseiten erscheinen sollen, funktioniert das Mapping analog über das Attribut "*contextSearchFolderNameEdit*" am Konzeptbegriff.

Soll der im Knowledge-Portal aktuell geöffnete Begriff (d.h. das Objekt im Fokus) als Bedingung in einer Expertensuche verwendet werden, so muss eine variable Bedingung vom Typ "*hat FrameID*" mit dem Variablennamen "*contextId*" in der Suche verwendet werden. Vor der Ausführung der Expertensuche wird diese Variable dann vom Knowledge-Portal immer auf die Frame-ID des gerade betrachteten Begriffs gesetzt.

Neben dem Parameter mit dem Namen "*contextId*" befüllt die Applikation auch noch einige andere Parameter der Expertensuche mit Werten, sofern die Parameter entsprechend benannt sind:

Parameter-Name	Bedeutung
contextLabel	Ist dieser Parameter in der Expertensuche definiert, so wird er mit dem Namen des Fokus-Objekts gefüllt.
date	Dieser Parameter wird mit dem aktuellen Datum und der aktuellen Uhrzeit gefüllt.
userDmid	Dieser Parameter wird mit der Frame-ID des aktuell angemeldeten Benutzers gefüllt.



kpath:[KPath-Ausdruck]	Ist ein Parameter mit dem Präfix " <i>kpath</i> :" im Namen definiert, so wird der nach dem Doppelpunkt stehende KPath-Ausdruck auf das Fokus-Objekt angewendet und das Ergebnis als Wert des betreffenden Parameters in der Expertensuche gesetzt. Der Platzhalter " <i>[KPath-Ausdruck]</i> " ist durch den auszuwertenden KPath-Ausdruck zu ersetzen.
------------------------	--

7.4.1.3 Titel der Kontext-Boxen festlegen

Um den Titel einer Kontextbox sowohl in deutscher als auch in englischer Sprache hinterlegen zu können, muss zunächst der jeweiligen Expertensuche eine externe ID zugeordnet werden. Danach kann der Fenstertitel für den deutschen Modus im Übersetzungsattribut "*msg_de*" und die englische Übersetzung davon im Attribut "*msg*" am Konfigurationsbegriff "*kpConfig > messages*" eingetragen werden, also z.B.:

```
context.externalId=Translation
```

wobei "*externalId*" durch die externe ID der Expertensuche und "Translation" durch den Fenstertitel in seiner jeweiligen Übersetzung ersetzt werden muss.

7.4.1.4 Erklären von Treffern

Eine Kontextbox kann so konfiguriert werden, dass hinter jedem Treffer ein Link erscheint, um den jeweiligen Treffer zu erklären. Klickt der Benutzer auf den Link, öffnet sich der Net-Navigator und es wird die Relations- und Begriffskette angezeigt, die bei Ausführung der Expertensuche zum entsprechenden Treffer führte.

Um die Erklärung von Treffern für eine Kontextbox zu aktivieren, muss beim zugehörigen Begriff das boolesche Attribut "*explainable*" gesetzt sein.

context	
Attribute	
explainable	<input checked="" type="checkbox"/>
Name (Deutsch)	German <input type="checkbox"/> univ. <input type="text" value="context"/>

Attribut explainable für den Begriff context

7.4.2 Ergebnisse von KPath-Ausdrücken in Kontextboxen anzeigen



7.4.2.1 Beschreibung

Neben der Anzeige von Expertensuchen in Kontextfenstern können auch die Ergebnisse von KPath-Ausdrücken (z.B. Listen von Attributwerten oder Konzepten/Instanzen) in Kontextfenstern ausgegeben werden.

Um die KPath-Ausdrücke zu definieren, die bei der Anzeige eines Konzept- oder Instanzbegriffs ausgewertet werden sollen, muss am Konzeptbegriff selbst bzw. am Konzeptbegriff der Instanz oder einem übergeordneten Konzeptbegriff ein Attribut mit dem internen Namen "*kpathContextBoxDefinitions*" angelegt werden. In dieses Attribut kann eine kommaseparierte Liste von KPath-Definitionen in der Form "*Name:Rückgabety:p:Kpath*" eingetragen werden. Der Platzhalter "*name*" sollte einen eindeutigen Namen für die KPath-Kontextbox enthalten, der keine Leerzeichen, Sonderzeichen und Umlaute enthalten darf. Der Platzhalter "*Rückgabety*" muss abhängig vom verwendeten KPath-Ausdruck durch den Buchstaben "v" (für "*values*") oder "t" (für "*topics*") ersetzt werden. Der Typ "v" muss für KPath-Ausdrücke angegeben werden, die eine Liste von Werten zurückliefern (normalerweise enden solche KPath-Funktion mit "*label()*", "*value()*" oder "*size()*"), der Typ "t" für KPath-Ausdrücke, die eine Liste von Topics zurückliefern. Der Platzhalter "*Kpath*" ist schließlich durch den zu verwendenden KPath-Ausdruck selbst zu ersetzen. Jede KPath-Kontextbox-Definition muss in einer neuen Zeile beginnen (d.h. eine Zeile pro Definition).

Hier ein Beispiel für den Inhalt des Attributs "*kpathContextBoxDefinitions*":

```
employeesSortedByIncome:t:~$hasEmployee$/target()/sort(@$income$,false)
notes:v:@$note$/value()
```

Im Beispiel werden zwei KPath-Kontextfenster mit den Namen "*employeesSortedByIncome*" und "*notes*" definiert. Die erste Box enthält eine Liste von Instanzen (die über die Relation "*hasEmployee*" zugeordneten Mitarbeiter, sortiert nach Einkommen) und die zweite Box eine Liste mit Zeichenketten (die Werte des Attributs "*note*").

7.4.2.2 Titel der Kontext-Boxen festlegen

Um den Fenstertitel für eine KPath-Kontextbox in der deutschen oder englischen Übersetzung festzulegen, muss die folgende Zeile in das Textattribut "*msg_de*" (bzw. "*msg*" für die englische Übersetzung) am Konfigurationsbegriff "*kpConfig > messages*" eingetragen werden:

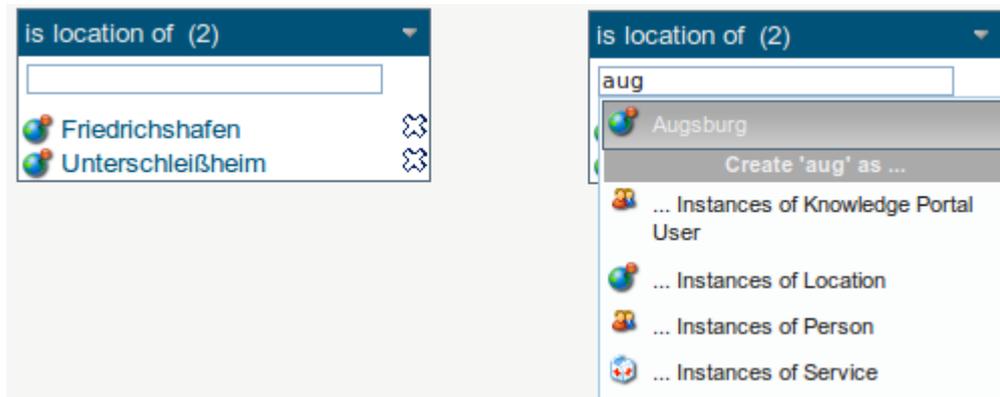
```
kpathContext.Name=Translation
```

wobei "*Name*" durch den im Definitionsattribut angegebenen Namen der KPath-Kontextbox und "*Translation*" durch den jeweiligen Text für den Fenstertitel ersetzt werden muss.

7.4.3 Micro-Edit-Boxen (ab Version 3.7.0)

7.4.3.1 Übersicht

Die Micro-Edit-Box ist ein neuer Box-Typ, der in Version 3.7 in das Knowledge-Portal eingefügt wurde. Die primäre Funktion dieser Box ist die Pflege von Relationen an einem angezeigten Individuum oder Begriff, ohne dass die Pflge-seite geöffnet werden muss.



Die Box wird für eine bestimmte Relation konfiguriert und zeigt als Inhalt die Relationsziele an. Über das Icon rechts der Elemente lassen sich Relationen löschen. Zusätzlich wird ein Textfeld eingblendet, über das man neue Relationen und Relationsziele anlegen kann.

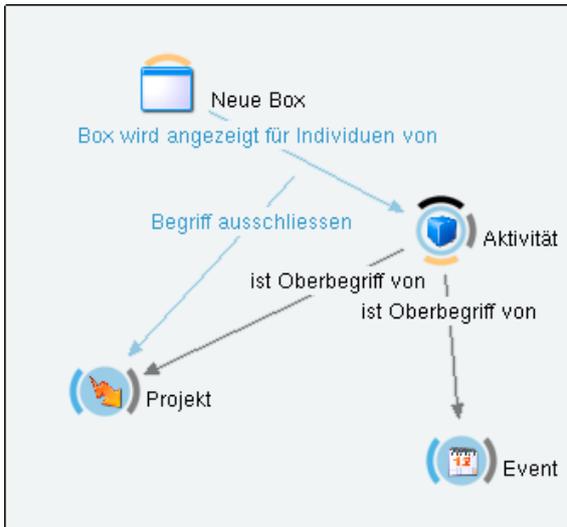
Während der Eingabe eines Namens wird automatisch nach möglichen Relationszielen gesucht. Wird das gewünschte Ziel nicht gefunden kann über weitere Einträge eine Domäne ausgewählt werden in der ein neues Relationsziel mit dem gewünschten Namen angelegt werden soll.

7.4.3.2 Zuordnung zu Begriffen

Die folgenden Relationen werden verwendet um zu definieren, bei welchen Begriffen und Individuen im Wissensnetz eine Micro-Edit-Box angezeigt werden soll:

Relation	Beschreibung
Box wird angezeigt für Individuen von	Wenn diese Relation zu einem Begriff des Wissensnetzes gezogen wird, dann wird diese Box auf der Detailseite von Individuen diese Begriffen und von Individuen von Unterbegriffen angezeigt.
Box wird angezeigt für Unterbegriffe von	Wenn diese Relation zu einem Begriff des Wissensnetzes gezogen wird, dann wird diese Box auf der Detailseite von Unterbegriffen und auf der Detailseite des Begriffes selber angezeigt.
Begriff ausschließen	Mit dieser Meta-Relation kann die Wirkung der "wird angezeigt für"-Relationen für Unterbegriffe aufgehoben werden.

Das folgende Beispiel illustriert die Zuordnung:



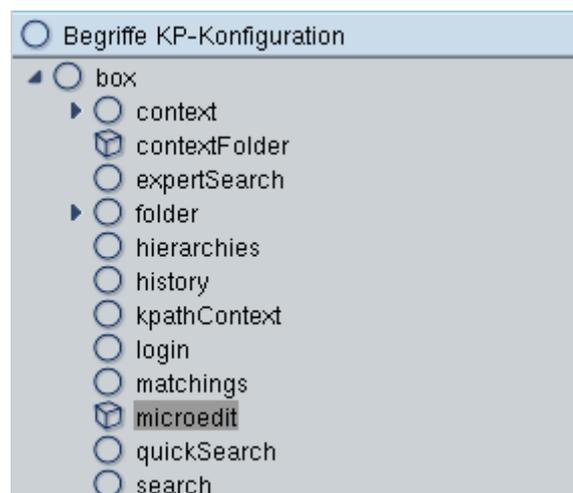
Anzeige von Boxen an Wissensnetz Begriffen

Hier wird eine Micro-Edit-Box mit dem Namen "Neue Box" definiert. Diese Box soll angezeigt werden, für Individuen des Begriffes "Aktivität". Dadurch wird die Box ebenfalls für Individuen aller Unterbegriffe (z.B. "Event") angezeigt.

Da eine Relation "Begriff ausschließen" zum Begriff "Projekt" geht, wird die Box für Individuen von "Projekt" nicht angezeigt, obwohl es sich hierbei um einen Unterbegriff von "Aktivität" handelt.

7.4.3.3 Konfiguration

Micro-Edit-Boxen werden konfiguriert, indem im Netz ein Individuum des Begriffes "microedit" angelegt wird. Verschiedene Attribute und Relationen steuern das Verhalten der Box:



Attribut/Relation	Typ	Beschreibung
Box bearbeitet Relation	Relation	Diese Relation verweist auf das Relationskonzept der Relation, die mit dieser Box bearbeitet werden soll.



orderKey	Ganzzahl	Dieses Attribut enthält die Position der Box im Vergleich zu anderen Boxen in dieser DisplayArea.
hideWhenEmpty	Boolesch	Wenn dieser Wert wahr ist, so wird die Box nur dann eingeblendet, wenn bereits Inhalt angezeigt würde. Ist die Box leer, so wird sie versteckt und kann über eine Auswahl über den Boxen wieder eingeblendet werden.
orderKPath	Zeichenkette	KPath nach dem die angezeigten Relationen sortiert werden
orderAscending	Boolesch	Gibt an, ob die durch orderKPath eingestellte Sortierung auf- oder absteigend sein soll
Eingabezeile verstecken	Boolesch	Gibt an, ob das Texteingabefeld für die Relationsszielsuche versteckt werden soll
editActivationKPath	Zeichenkette	KPath-Ausdruck mit den Ergebniswerten true / false. Dieser Ausdruck gibt an, ob die Eingabezeile angezeigt werden soll oder nicht. Ist das Ergebnis "true", so wird die Zeile angezeigt. Ist das Ergebnis "false", so wird die Zeile nicht angezeigt.
Angezeigte Relationsbeschreibung	Zeichenkette	Dies ist der Text, der als Tooltip für die Eingabezeile angezeigt wird. Ist dieses Attribut nicht gesetzt, dann wird stattdessen der Name der konfigurierten Relation angezeigt.
ID der Relationszielsuche	Zeichenkette	Dieses optionale Attribut dient dazu, eine spezielle Relationszielsuche zu verwenden. Die konfigurierte Suche wird verwendet, um die Vorschläge in der Autovervollständigung anzubieten. Ist dieses Attribut nicht konfiguriert, dann wird die voreingestellte Suche für die Suche der Ziele verwendet.
Max. Anzahl angezeigter Relationsziele im Autocomplete-Menü	Ganzzahl	Ab Version 3.8 Standardwert: 20 Gibt an, wieviele Relationsziele im Autocomplete-Menü angezeigt werden sollen. "0" bedeutet, dass keine Relationsziele nach einer Benutzereingabe vorgeschlagen werden.
Max. Anzahl angezeigter Zieltypen im Autocomplete-Menü	Ganzzahl	Ab Version 3.8 Standardwert: 20 Gibt an, wieviele Zieltypen angezeigt werden, die zum Anlegen neuer Individuen verwendet werden können. "0" bedeutet, dass keine Zieltypen vorgeschlagen werden.



Wie bei den anderen Boxen auch, werden Default-Werte vom Begriff und dem Box-Begriff selber übernommen.

Hinweis zu Version 3.7.0: Die Default-Werte sind noch nicht mit den spezifischen Konfigurationsmöglichkeiten konsolidiert, so werden "limit", "isLimited", "defaultCollapse", "defaultDisplayArea" und die Display-Area pro View-Id zwar von den default-Werten übernommen, können aber nicht pro Box selbst konfiguriert werden. Andererseits kann gibt es für "Eingabezeile verstecken" keinen Default-Wert. Diese Manko ist bekannt und wird voraussichtlich in Version 3.8 behoben.

7.4.3.4 Kontextsuchen integrieren.

Zusätzlich zur Darstellung der Relationen kann eine oder mehrere Expertensuchen zu Darstellung innerhalb der Box konfiguriert werden.



Dazu muss die Relation "**Box enthält Box**" zu einem Individuum des Begriffs "**contextFolder**" gezogen werden. Dort kann der Inhalt der integrierten Box konfiguriert werden:

At-tribute/Relation	Typ	Beschreibung
name (Namen-sattribut)	Zeichen-kette	Der angezeigte Titel der Kontextsuche.
folderEID	Zeichen-kette	Die externe ID des Suchordners
explainable	Boolesch	Gibt an, ob rechts die Glühlampe zum Erklären der Treffer angezeigt werden soll.
orderKey	Ganzzahl	Dieser Wert bestimmt die Sortierung der integrierten Boxen innerhalb der Micro-Edit-Box
orderKPath, orderAscending	Zeichen-kette, Boo-lesch	Dieser KPath bestimmt die Sortierung der angezeigten Element in der Kontext-Box

Der Suchordner, der über "**folderEID**" konfiguriert wird, kann nach den gleichen Regeln aufgebaut werden, wie andere Kontextsuchen auch.



7.4.4 Ergebnisse einer Textsuche in einer Kontextbox anzeigen

7.4.4.1 Beschreibung

Auf Detail- und Pflegeseiten können die Ergebnisse einer Suchanfrage an eine Textsuche in einer Kontextbox dargestellt werden.

In den Knowledge-Portals bis Version 3.7 können hierbei nur K-Infinity-Suchen aufgerufen werden. Details zur Konfiguration von K-Infinity-Suchen können dem K-Infinity-Handbuch (siehe Link rechts) entnommen werden. Ab KP-Version 3.8.0 können Suchen über alle Knowledge-Portal-Suchkomponenten ausgeführt werden. Dies hat den Vorteil, dass über Suchadapter in Java auch Suchmaschinen anderer Hersteller angebunden werden können (es stehen z.B. Suchadapter zur Anbindung des Apache-Nutch-Webcrawlers und zur Anbindung von Suchmaschinen über eine HTTP-REST-XML-Schnittstelle zur Verfügung). Details zur Suchkonfiguration mit Knowledge-Portal-Suchkomponenten sind im Kapitel 5.1 dieses Handbuchs beschrieben (siehe Link rechts).

Die folgenden Suchparameter werden abhängig von dem Begriff oder Individuum, in dessen Kontext die Box angezeigt wird, in die Suchanfrage eingegeben:

- **searchString**: Der Name des Kontext-Objektes wird als Suchausdruck in die Suchanfrage eingegeben. Dieser Parameter wird von den meisten K-Infinity-Suchen und von den Suchen anderer Suchkomponenten unterstützt.
- **initialTopics**: Bei Bestimmten K-Infinity-Suchen wie z.B. der semantische Suche oder der NextBot-Ähnlichkeitssuchen kann ein Wissensnetz-Objekt anstelle eines Suchstrings als Sucheingabe verwendet werden. Für diese Suchen wird das Objekt, in dessen Kontext die Box angezeigt wird, als Parameter in die Suchanfrage eingegeben
- **contextId-Variable** (ab KP-Version 3.8.0): Für die zusammenstellbaren Suchen in K-Infinity wird die ID des Wissensnetz-Objektes, in dessen Kontext die Box angezeigt wird, als Parameter mit dem Namen "contextId" in die Suchanfrage eingegeben. Der Parameter kann in der Suchkonfiguration der zusammenstellbaren Suche auf eine KScript-Variable abgebildet und dann z.B. in Expertensuchen oder anderen Suchbausteinen für kontextabhängige Operationen genutzt werden.

Das Ergebnis der Suche wird immer nach Trefferqualität sortiert in der Box angezeigt, d.h. die besser bewerteten Treffer erscheinen weiter vorn, die schlechter bewerteten Treffer weiter hinten in der Liste. Ab KP-Version 3.8.0 wird die Trefferbewertung direkt in der Box angezeigt, wobei beim Anklicken der Bewertung eine Erklärung des Treffers aufgerufen werden kann.

7.4.4.2 Definition im Wissensnetz

Für Detail- und Pflegeseiten wird eine separate Konfiguration unterstützt.

Die Konfiguration wird an einem Begriff vorgenommen. Die vorgenommene Konfiguration wird für alle Unterbegriffe und deren Individuen angewendet.

Anwendungsbereich	Typ	Interner Name
Detailseite	Zeichenkette	contextSearchName
Pflegeseite	Zeichenkette	contextSearchNameEdit

In Knowledge-Portals bis Version 3.7 muss als Attributwert die externe ID einer K-Infinity-

Suche angegeben werden. In Knowledge-Portals ab Version 3.8 muss die ID einer Knowledge-Portal-Suchkonfiguration angegeben werden. Sollen mehrere Suchen konfiguriert werden, so kann ein mehrzeiliger Attributwert angegeben werden, wobei pro Zeile die ID einer Suche bzw. Suchkonfiguration angegeben werden muss.

7.4.4.3 Titel der Kontext-Box festlegen

Wird keine spezielle Übersetzung konfiguriert, dann wird der interne Name der konfigurierten Suche als Titel der Box angezeigt.

Sollen die Titel übersetzt angezeigt werden, dann muss eine Übersetzung im Netz hinterlegt werden. Der Schlüssel setzt sich zusammen aus dem Prefix "context." und dem internen Namen der konfigurierten Suche.

Beispiel:

- Interner Name: "*kontextsuche*"
- Die Übersetzung muss für den Schlüssel "*context.kontextsuche*" angelegt werden.

7.4.5 Die Ähnliche-Themen-Kontextbox (Matchings)

Mit Version 3.7 des Knowledge-Portals wurde die Matchings-Box aus dem Portal entfernt. Stattdessen kann eine Kontextbox für eine einfache Suche vom Typ "Matching Suche" definiert werden.

7.5 HTML-Editor

7.5.1 Überblick

Der **HTML-Editor** kann sowohl direkt in der Detailseite des Topics verwendet werden, als auch in der Pflegeseite. Dies wird über Unterbegriffe des Begriffs "*renderers*" mit dem Namen "*renderer*" konfiguriert.



Ausschnitt des Netzes zum Begriff
renderers

HTML-Editor in der Detailansicht

Damit der HTML-Editor in der Detailansicht des Topics gezeigt wird müssen die folgenden Attribute/Werte bei einem Begriff "renderer" vorhanden sein:

```
componentClass_  
com.iviews.knowledgeportal.model.Attribute
```

```
renderMode:  
internalName_INTERNER_NAME
```



renderPage:

/WEB-INF/jsp/renderers/com/iviews/knowledgeportal/model/attribute_htmltext.jsp

Beim Wert des Attributs "*renderMode*" muss die Zeichenkette "*INTERNER_NAME*" durch den internen Namen des Attributs ersetzt werden, für das der HTML-Editor direkt in der Detailansicht des Begriffs oder Individuums aufgerufen werden soll.

Soll der HTML-Editor **über die gesamte Breite** der Topic-Anzeige dargestellt werden, so ist ein weiterer Begriff "*renderer*" mit den folgenden Werten erforderlich:

componentClass_

com.iviews.knowledgeportal.viewconfig.model.AttributeViewComponent

renderPage:

/WEB-

INF/jsp/renderers/com/iviews/knowledgeportal/viewconfig/model/attributeviewcomponent_htmltext.jsp

renderMode:

internalName_INTERNER_NAME

HTML-Editor in der Pflegemaske

Analog zur Anzeige des HTML-Editor in der Detailansicht des Topics werden auch für die Pflegemaske zwei Unterbegriffe "*renderer*" erzeugt, wobei einer die Verwendung des HTML-Editors aktiviert und der andere für die Nutzung der kompletten Breite des Pflegefensters konfiguriert werden kann.

Nutzung des HTML-Editor in der Pflegemaske

componentClass_

com.iviews.knowledgeportal.model.edit.KEditableStringAttribute

renderMode:

internalName_INTERNER_NAME

renderPage:

/WEB-

INF/jsp/renderers/com/iviews/knowledgeportal/model/edit/keitablestringattribute_htmltext.jsp

Nutzung der gesamten Breite:

componentClass_

com.iviews.knowledgeportal.viewconfig.model.edit.EditableAttributeViewComponent

renderPage:

/WEB-

INF/jsp/renderers/com/iviews/knowledgeportal/viewconfig/model/edit/editableattributeviewcomponent_htmltext.jsp

renderMode:

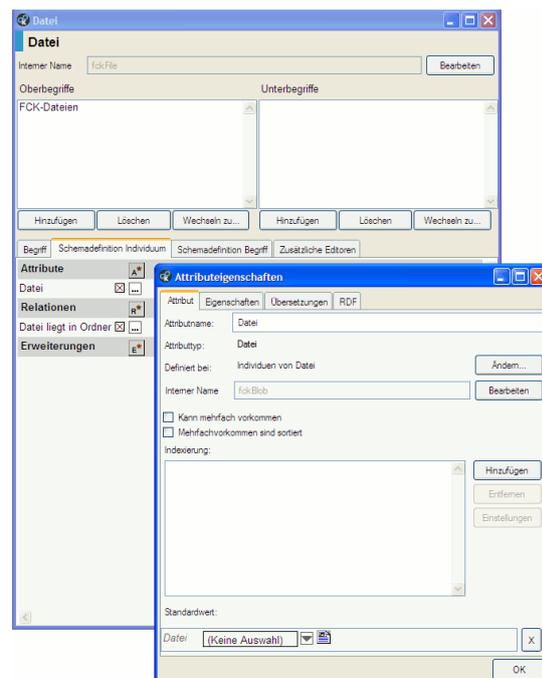
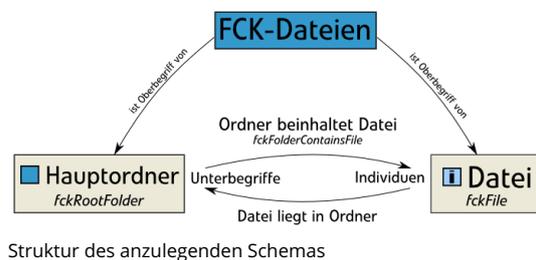
internalName_INTERNER_NAME

7.5.2 Bilder-Upload konfigurieren

Der FCK-Editor bietet die Möglichkeit, Bilder auf dem Server zu speichern und in das HTML-Dokument einzufügen. Um diese Möglichkeit zu benutzen müssen folgende Strukturen im Netz angelegt werden.

1. Ein neuer Hauptbegriff "FCK-Dateien" muss angelegt werden. In diesem Teilnetz werden die Ordner und die Dateien gespeichert
2. Ein neuer *individuenfähiger* Begriff "Datei" muss unter dem Hauptbegriff "FCK-Dateien" angelegt werden. Der interne Name dieses Begriffes muss auf "fckFile" gesetzt werden.
3. In der Schema-Definition für Individuen dieses Begriffes muss ein Datei-Attribut mit dem Namen "Datei" und dem internen Namen "fckBlob" angelegt werden.
4. Ein neuer Begriff "Hauptordner" muss unter dem Begriff "FCK-Dateien" angelegt werden. Dieser Begriff erhält den internen Namen "fckRootFolder"
5. In der Schema-Definition für Unterbegriffe von "Hauptordner" muss eine Relation "Datei liegt in Ordner" angelegt werden.
Das Ziel dieser Relation ist "Individuen von Datei" und die Rückrelation heißt "Ordner beinhaltet Datei" (interner Name "fckFolderContainsFile").

Die Struktur der Objekt-Definitionen sieht somit folgendermaßen aus:



Knowledge-Builder bei der Definition des Datei-Begriffs

7.6 Konfiguration der Kommentar-Funktion

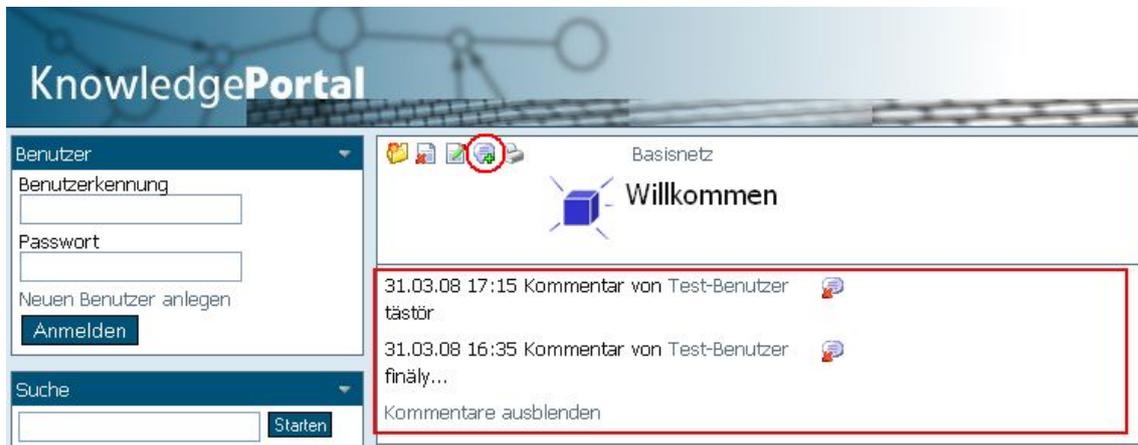
7.6.1 Überblick

Mit Hilfe der Kommentarfunktion können Nutzer des Knowledge-Portals Kommentare zu Wissensnetz-Objekten schreiben. Ist die Funktion korrekt konfiguriert, so erscheint



oben in der Icon-Leiste ein Kommentar-Symbol, über das ein Kommentar geschrieben werden kann.

Mit Hilfe des Rechtesystems kann festgelegt werden, welche Benutzer Kommentare schreiben und löschen dürfen und welche Benutzer Kommentare lesen dürfen.



Die Kommentare werden im Wissensnetz als Relation zwischen dem Benutzer und dem kommentierten Objekt gespeichert. Diese Relation hat Meta-Attribute für den Inhalt und Zeitpunkt des Kommentars.

7.6.2 Konfiguration der Kommentar-Funktion (bis KP 3.7)

Im Schema müssen die folgenden Relationen angelegt werden um die Kommentarfunktion nutzen zu können:

	Name	Interner Name	Definiert bei
Relation	kommentiert	com-ments	Individuen des Personenbegriffs
abstrakte Rück-Relation	wird kommentiert von (abstrakt)	commented-By	Individuen von Wurzelbegriff

Die abstrakte Rückrelation benötigt zwei konkrete Relationshälften als Unterrelationen. Diese brauchen keine internen Namen.

Name	Definiert bei	Sonstiges
Individuum wird kommentiert von	Individuen von Wurzelbegriff	Unterrelation von "wird kommentiert von (abstrakt)"
Begriff wird kommentiert von	Unterbegriffe von Wurzelbegriff	Unterrelation von "wird kommentiert von (abstrakt)"



Die abstrakte Relation "wird kommentiert von (abstrakt)" erhält die folgenden Meta-Attribute:

Attribut	Attribut-Typ	Interner Name
Kommentar	Zeichenkette	comment
Zeitpunkt des Kommentars	Datum und Uhrzeit	timestamp

Weiterhin müssen im Rechtesystem die folgenden Operation hinzugefügt werden:

Operationssymbol	Beschreibung
showComments	Kommentar lesen
createComment	Kommentar schreiben

7.6.3 Konfiguration der Kommentarfunktion ab KP 3.8

Die Knowledge-Portal-Version 3.8 bringt die folgenden Änderungen bei der Kommentarfunktion mit sich:

- Die Relationen und Attribute für die Kommentarfunktion (siehe vorangehendes Kapitel) werden beim Einrichten bzw. Upgrade der KP-Wissensnetz-Komponente im Admin-Tool automatisch angelegt, sofern diese nicht bereits im Wissensnetz vorhanden sind (es werden die internen Namen geprüft). Außerdem wird automatisch eine Objektliste mit dem Namen "kpComments" für die Kommentar-Relationen (wird kommentiert von) angelegt, die zur Anzeige der Kommentare dient.
- Kommentare werden nun nicht mehr als lange Liste angezeigt, sondern in einer tabellarischen Ansicht, die blätterbar, durchsuchbar und sortierbar ist. Dies verbessert die Übersicht und verringert die Ladezeit bei Objekten, zu denen viele Kommentare abgegeben wurden (siehe Bild unten).
- Die generelle Möglichkeit zum Kommentieren eines Topics bzw. zur Anzeige der Kommentare wird nun nicht mehr über die spezielle Rechtesystem-Operation "showComments" konfiguriert, sondern über das Element "Comments" in der ViewConfig. Beim Einrichten bzw. Upgrade der KP-Wissensnetz-Komponente im Admin-Tool wird in allen ViewConfigs automatisch ein Comments-Element eingefügt. Ist in der ViewConfig ein Header-Element vorhanden, so wird das Comments-Element hinter dem Header-Element eingefügt. Ist kein Header-Element vorhanden, so wird das Comments-Element als erstes Kind-Element unterhalb des ViewConfig-Elements eingefügt. Die Berechtigung zum Lesen von Kommentaren wird über die Standard-Operationen im Rechtesystem zum Lesen von Relationen geregelt. Details zum Comments-



Element sind in der Beschreibung der ViewConfig zu finden (siehe Verknüpfung in der rechten Spalte).

- Die Berechtigungen zum Anlegen und Löschen eines Kommentars (Kommentar-Relation) wird nicht mehr über die speziellen Rechtesystem-Operationen "createComment" und "deleteComment" ermittelt, sondern über die normalen Rechtesystem-Operationen für die Relationspflege (Relation-/Relationshälfte anlegen bzw. löschen). **Die Migration des Rechtesystems muss manuell durchgeführt werden.**

Es gibt 3 Kommentare zu diesem Thema. [Kommentare ausblenden](#) [Kommentar schreiben](#)

Datum	Kommentar	Autor	
Suchen in Datum...	Suchen in Kommentar...	Suchen in Autor...	
12.03.2012 14:20:09	Mein erster Kommentar!	Luick, Christof	✘
12.03.2012 14:20:58	Dieser Arbeitsbereich gefällt mir am besten!!!!	Luick, Christof	✘
12.03.2012 14:21:25	Und noch ein sehr sehr sehr sehr sehr sehr sehr sehr sehr langer Kommentar.	Luick, Christof	✘

1 bis 3 von 3 Einträgen

Erster Zurück 1 Nächster Letzter

7.7 Tooltips

(ab Version 3.8.5)

Auf Detail und Pflegeseiten können an verschiedenen Stellen Hilfetexte als Tooltip eingebildet werden.

Standard-Konfiguration

In der Voreinstellung des Knowledge-Portal gibt es ein Attribut "Tooltip" mit dem internen Namen "tooltipEdit", dass an allen Begriffen (an Attribut- und Relationsbegriffe sowie an normalen Begriffen) angelegt werden. Der Inhalt dieses Attribute wird jeweils als Tooltip angezeigt, wenn der Benutzer mit der Maus über das Label eines Attributes oder einer Relation fährt. Fährt der Benutzer über den Namen des angezeigten Objektes im Kopfbereich der Seite, wird der Inhalt des Attributes "Tooltip" angezeigt, dass am Begriff des Objektes vorhanden ist.

Änderung der Konfiguration

Über die Konfigurations-Option "tooltipKPath" kann ein alternativer KPath festgelegt werden um das Attribut aufzufinden, dessen Inhalt als Tooltip angezeigt werden soll. Folgende Anforderungen werden an den KPath gestellt:

- Die Eingabe ist das Konzept des angezeigten Objektes/Attributes/Relation.
- Das Ergebnis muss das Attribut (nicht der Attributwert) sein, da unter Umständen verschiedene Übersetzungen des Attributes geladen werden.

7.8 Collections - Wissensnetzelemente sammeln und teilen

Collections dienen wie auch die Kontextbox Favoriten dazu, Individuen aus dem Knowledge-Portal zu sammeln. Der Einsatzbereich von Collections und Favoriten unterscheidet sich in



folgenden Aspekten:

- **Die Strukturierung von Individuen in Gruppen:** Im Gegensatz zu der Kontextbox Favoriten kann jeder Nutzer mehrere Collections (Sammlungen von Objekten) gleichzeitig haben, jedoch zu einem bestimmten Zeitpunkt nur eine bearbeiten.
- **Das Teilen mit anderen Nutzern:** Collections können für die Zusammenarbeit, müssen aber nicht, mit anderen Nutzern geteilt werden. Die Kontextbox Favoriten ist dagegen immer privat.

Das Sammeln von Individuen

Collections sind Sammelmappen für Individuen. Im Gegensatz zu den Favoriten können Collections mit anderen Nutzern geteilt und zum gemeinsamen Arbeiten verwendet werden. Dafür können Collections anderen Mitarbeitern zugewiesen werden. Des Weiteren ist es möglich, mehrere Sammelmappen zu haben und damit Objekte aus verschiedenen Kontexten gleichzeitig zu verwalten.

Die aktive Collection wird i.d.R. im linken Interaktionsbereich angezeigt. Sie stellt die Drop-Zone der aktuellen Collection dar. Möchte man der Collection neue Einträge hinzufügen, zieht man das Icon des gewünschten Individuums in die Drop-Zone.



Hat man das Individuum-Icon in die Drop-Zone gezogen, öffnet sich ein Fenster. Dort kann man eine Bemerkung zu dem ausgewählten Individuum eintragen. Die Bemerkung wird an dem Eintrag in der Sammelmappe erstellt und nicht an dem Individuum, das man in die Collection gezogen hat. D.h. die Bemerkung existiert nur so lange wie die Collection existiert und wird auch nur in der Collection angezeigt.



Sammelmappe: PM Kranwagen



Bemerkung

14675 - Kranwagen basiert auf dem Produktmodell Kranwagen

Collect

Abbrechen

Durch Klick auf den Name der Collection Alles rund um den Kranwagen über der Drop-Zone wird die Detailseite der entsprechenden Collection aufgerufen. Auf der Detailseite einer Collection werden alle Einträge der Collection angezeigt, zusätzlich ist es möglich, eine Beschreibung für die Collection hinzuzufügen. Außerdem können die gesammelten Objekte angesehen, bearbeitet und gelöscht werden.

Collection

Alles rund um den Kranwagen



Klicken Sie doppelt auf dieses Feld, um den Text-Editor zu öffnen.

PV14675 - Kranwagen - made in China
Produktvariante für PV14675 Kranwagen
[..]
PM Kranwagen
PV14675 Kranwagen basiert auf dem Produktmodell Kranwagen
[..]
Projekt Kranwagen
Projekt zu PV14675 Kranwagen
[..]

Die Verwaltung von Collections

Die Verwaltung von eigenen Collections findet auf dem Reiter Collection der eigenen Personen-Detailseite statt. Dort werden alle Collections des jeweiligen Nutzers angezeigt.



Experte für (0)	Produktmanager für (0)	Produktentwickler für (0)	Freigaben (0)	Aufgaben (0)	Collection (2)			
+ Neue Collection erstellen				Erster	Zurück	1	Nächster	Letzter
Name	gehört Person	zugeordnete Personen	Collection Items					
Suchen in Name...	Suchen in gehört Person...	Suchen in zugeordnete Persc	Suchen in Collection Items...					
Alles rund um den Bagger	Schulze, Kathrin	Schnitter, Thorsten	70488 - Bagger,Produktcharge PV70488 - Bagger - made in Germany - 2805-131-15,PV70488 - Bagger - made in Germany	 				
Alles rund um den Kranwagen	Schulze, Kathrin		PM Kranwagen,Projekt Kranwagen,PV14675 - Kranwagen - made in China	 				
1 bis 2 von 2 Einträgen				Erster	Zurück	1	Nächster	Letzter

Mit dem Icon  neben der Collection kann diese aktiviert werden. Es wird immer nur die aktive Collection eines Nutzers im Drop-Down-Bereich angezeigt. Um eine Collection bearbeiten zu können, muss diese zuerst aktiviert werden.

Eine neue Collection kann ebenfalls auf dem Reiter Collection erstellt werden. Dafür muss der Name der neuen Collection eingegeben werden. Die neue Collection wird automatisch als aktive Collection hinzugefügt.

Kontextinformationen zu Collections

Auf der Detailseite einer Collection stehen verschiedene Kontextinformationen zur Verfügung.

- Collection gehört (1) 
- Collection ist Mitarbeiter zugewiesen (1) 
- Eigene Collections (2) 
 -  Alles rund um den Bagger 
 -  Alles rund um den Kranwagen 
- Zugewiesene Collections (1) 



- Collection gehört: Die angezeigte Collection gehört der in der Kontextbox angezeigten Person.
- Collection ist Mitarbeiter zugewiesen: Für die Zusammenarbeit kann die Collection mit anderen Personen geteilt werden, dafür kann die Kontextbox verwendet werden. Wurde die Collection einem Mitarbeiter zugewiesen, wird sie in der Box Zugewiesene Collections bei dem Mitarbeiter angezeigt.
- Eigene Collections: Alle eigenen Collections werden angezeigt. Auf der Personendetailseite können diese verwaltet werden.
- Zugewiesene Collections: Alle Collections, die der angemeldeten Person für eine Zusammenarbeit zugewiesen wurden. Die Collections wurden von anderen Personen erstellt, können aber weiter bearbeitet werden.

Collections mit anderen Nutzern teilen

Collections sind zuerst immer privat. Erst wenn eine Collection einem anderem Mitarbeiter zugewiesen wird, kann sie von ihm eingesehen und bearbeitet werden. Möchte man eine bestimmte Collection einem Mitarbeiter zuweisen, geht das mithilfe der Kontextboxen auf der Detailseite der Collection.

Collection einem Mitarbeiter zuweisen:

1. Die Detailseite der Collection aufrufen.
2. In der Kontextbox Collection ist Mitarbeiter zugewiesen die Person auswählen, mit der die Collection geteilt werden soll und bestätigen.

Eine zugewiesene Collection zum Bearbeiten öffnen:

1. Die Detailseite irgendeiner Collection aufrufen.
2. In der Kontextbox Zugewiesene Collections werden alle Collections von anderen Personen angezeigt, die für die Zusammenarbeit (mit der aktuell im Knowledge-Portal angemeldeten Person) geteilt wurden.
3. Wählt man eine der zugewiesenen Collections aus, werden die Objekte und Bemerkungen angezeigt. Zugewiesene Mitarbeiter einer Collection dürfen die Inhalte der Collections bearbeiten.

Je nach Konfiguration der Collections ist es möglich eine zugewiesene Collection zu aktivieren und dann weitere Individuen hinzuzufügen.

8 Pflegeseiten für Begriffe und Individuen

8.1 Erstellen einer Edit-Konfiguration

Die Edit-Konfigurationen werden im Wissensnetz konfiguriert. Dazu muss der Begriff "editConcept" als Unterbegriff des Konfigurations-Begriffs "concept" erstellt werden. Dieser dient als Oberbegriff für alle erstellten Pflegeseiten-Konfigurationen.

Um eine Konfiguration für eine bestimmte Layoutkategorie zu erstellen, muss ein Unterbegriff mit dem Namen der Layout-Kategorie erstellt werden, für die die Konfiguration angewendet werden soll. Für die Konfiguration des Personenbegriffs mit der Layout-Kategorie "person" würde also der Begriff "concept > editConcept > person" erstellt werden.

Die Konfiguration selbst wird in drei unterschiedlichen Textattributen an diesem Begriff gespeichert. Unterschieden wird hier zwischen den Attributen "concept", "instance" und "ex-



tension".

Die Konfigurationsdatei selbst wird in XML definiert und muss nach der DTD "editConfig.dtd" valide sein. Die DTD selbst finden Sie im Anhang dieses Dokuments.

concept

Diese Konfiguration wird für Begriffe angewendet. Beispielsweise würde der Begriff "Person" mit dieser Konfiguration dargestellt werden, Individuen von Person jedoch mit der Konfiguration aus dem Textattribut "instance".

Die Konfiguration für Begriffe hat folgende Besonderheiten:

- Das erste Konfigurationselement muss `<Header>` sein.
- Das Konfigurationselement `<AllExtensions>` ist nicht zulässig, da Begriffe nicht erweiterungsfähig sind.
- Das Element `<Extension>` kann nur in Verbindung mit dem Unterelement `<KPath>` verwendet werden, da Begriffe keine Erweiterungen haben können.

instance

Beinhaltet die Konfiguration der Individuen. Auch hier gelten besondere Einschränkungen

- Das erste Konfigurationselement muss `<Header>` sein

Die Konfiguration für Instanzen kann ebenso wie die View-Konfiguration an zwei Stellen im Knowledge-Builder bearbeitet werden. Einerseits am Konfigurationsbegriff selbst und andererseits auf dem Reiter *KP-Edit* im Begriffseditor des Begriffs, für den die Konfiguration gilt. Siehe dazu auch Kapitel "7.2.1 Erstellen einer View-Konfiguration".

extension

Die hier gespeicherte Konfiguration dient primär zum Darstellen von Erweiterungen. Daher bestehen für diese Konfiguration folgende Einschränkungen:

- Sie darf *kein* `<Header>` Element enthalten
- Sie darf *kein* `<Extension>` Element enthalten
- Sie darf *kein* `<AllExtensions>` Element enthalten

Sie kann jedoch verwendet werden, um andere Topics auf die Pflegeseite einzubinden. Hierfür wird die "extension" - Konfiguration des eingebundenen Topics verwendet.

Beispielsweise kann auf der Pflegeseite einer Firma der Ansprechpartner mit eingebunden und editiert werden. Dieser wird dann mit der Konfiguration "extension" für die layoutCategory "person" dargestellt.

8.1.1 Allgemeines Verhalten

Sofern keine spezielle EditConfig für einen Objekttyp festgelegt ist, werden auf einer Pflegeseite automatisch die folgenden Attribute und Relationen angezeigt:

- KP-Version 3.7 oder älter: Es werden automatisch alle Attribute und Relationen angezeigt, die einen internen Namen haben und die nicht über die Konfiguration für die versteckten Attribute ausgeblendet sind (siehe Kapitel "Ausblenden bestimmter Attribute und Relationen" bzw. Verknüpfung in der rechten Spalte)
- Ab KP-Version 3.8: Es werden automatisch alle Attribute und Relationen angezeigt, für welche die Rechtesystem-Operation "KP: Auf Pflegeseite anzeigen" erlaubt ist. Details zu



diesem Mechanismus sind ebenfalls im Kapitel "Ausblenden bestimmter Attribute und Relationen" (Verknüpfung in der rechten Spalte) zu finden.

Die übergreifende Konfiguration der Pflegeseite ermöglicht das Ändern des allgemeinen Verhaltens.

Konfigurationsschlüssel	Mögliche Werte	Standardwert	Beschreibung
editConfig.showConfirmOnCommit	true / false (boolesches Attribut)	true	<p>Ist diese Einstellung angeschaltet, so wird nach dem Speichern eine Bestätigung angezeigt. Diese signalisiert dem Benutzer, dass die Einstellungen korrekt gespeichert wurden.</p> <p>Der Inhalt der Bestätigung kann geändert werden. Die zugehörigen Übersetzungsschlüssel sind:</p> <ul style="list-style-type: none">• confirmCommit.title Standardwert: <i>Änderungen gespeichert</i>• confirmCommit.text Standardwert: <i>Alle Änderungen wurden erfolgreich gespeichert.</i>• confirmCommit.continue Standardtext: <i>Weiter</i>
editConfig.showConfirmOnCloseWindow	true / false (boolesches Attribut)	false	<p>Gibt an, ob beim Schliessen eines Edit-Fensters (klick auf das "x" oben rechts), der User gefragt werden soll, ob er das Fenster wirklich schliessen wollte. Die Abfrage kommt allerdings erst nachdem das Fenster geschlossen wurde.</p>

8.1.2 Allgemeines Aussehen

Konfiguration	Mögliche Werte	Standardwert	Beschreibung
---------------	----------------	--------------	--------------



topic-Form.toolbarAtTop		true	Gibt an, ob die Knöpfe am oberen Rand der Seite angezeigt werden sollen.
topicForm.toolbarAtBottom		false	Gibt an, ob die Knöpfe am unteren Rand der Seite angezeigt werden sollen. Sind "toolbarAtTop" und "toolbarAtBottom" gleichzeitig aktiviert, so wird die Werkzeugleiste sowohl oben als auch unten angezeigt.
topicForm.showApplyButton	true / false (boolsches Attribut)	true	Gibt an, ob der Knopf zum Übernehmen der Daten angezeigt werden soll. Das Übernehmen der Daten schickt Änderungen an den Server, lässt das Fenster jedoch geöffnet. Damit die Änderungen endgültig übernommen werden, müssen diese jedoch gespeichert werden.
topicForm.cancelAction	Zeichenkette	cancel-Window	<i>Nur für Experten</i> Gibt an, welche Aktion durch Betätigen des Knopfes "Abbrechen" durchgeführt werden soll.
topicForm.useChangeFileApplet	true / false (boolsches Attribut)	true	Gibt an, ob das ChangeFileApplet angezeigt werden soll.
topicForm.useMultiFileUpload	true / false (boolsches Attribut)	true	Gibt an, ob das MultiFileUpload-Applet angezeigt werden soll.
topicForm.autoCreateWithSingleTarget	true / false (boolsches Attribut)	true	Hiermit kann festgelegt werden, ob nach der Relationszielsuche bei nur einem möglichen Relationsziel automatisch eine Relation angelegt werden soll.
topicForm.relationTargetSearchPageSize		20	<i>Nicht mehr unterstützt.</i>
topicForm.possibleTargetDomainsLimit		1000	Die maximale Anzahl der auf dem Reiter "Relationsziel anlegen" angezeigten Objekte im Dialog für die Relationszielauswahl, kann hiermit festgelegt werden.



topicForm.session KeepAliveInterval	Ganzzahl (Einheit: Minuten)	5	Gibt an, in welchen Zeitabständen (in Minuten) bei einer geöffneten Pflege-seite ein HTTP-Request gesendet werden soll, damit die Web-Session nicht abläuft. Wird 0 als Wert für den Keep-Alive-Intervall angegeben, so wird der Keep-Alive-Request deaktiviert. Der Standardwert für diese Einstellung beträgt 5 Minuten.
--	-----------------------------------	---	--

8.1.3 Attribute

Attribute für den Tag <Attribute>:

Attribut	Typ	Standard	Beschreibung
id	(string)		Die ID, die auf der generierten Seite als ID am Attributcontainer generiert wird.
styleClass	(string)		Der Wert wird in das Attribut class am Tag des Attributcontainers angebracht.
renderMode	(string)		Der RenderMode für die Darstellung des Attributes.
sortKPath	(string)		Bestimmt die Sortierung der ermittelten Attribute.
sortAscending	true/false		Gibt an, ob die Sortierung aufsteigend oder absteigend erfolgen soll.
metasMode	off / lazy / direct	off	Bestimmt die Anzeige von Metaeigenschaften für die einzelnen Attribute.

Unterelemente <Label> oder <LabelKey>

Als Label wird ohne explizite Angabe der Name des Attributes benutzt. Optional kann hier durch das Unterelement <Label> ein fester String gesetzt werden, oder durch die Angabe von <LabelKey> ein Schlüssel für eine Übersetzung angegeben werden. (Diese muss in der msg.properties definiert werden!)

Unterelement <Tooltip> oder <TooltipKey>

Konfiguration des Tooltips, der bei den Attributen angezeigt wird.

Unterelement <Valuetip> oder <ValuetipKey>

Konfiguration des Textes, der in der Eingabezeile eines Attributes angezeigt wird, wenn noch kein eigener Text eingegeben wurde.

Unterelement <InternalName> / <KPath>

Welches Attribut angesprochen werden soll, wird über eines dieser Unterelemente definiert.



Daher ist die Angabe des Unterelements `<InternalName>` oder `<KPath>` zwingend erforderlich.

Unterelement `<RenderHint>`

Über dieses Konfigurationselement können dem Renderer, der HTML-Quelltext generiert, verschiedenen Parameter übergeben werden.

Name	Mögliche Werte	Standardwert	Beschreibung
acceptedFile-Extensions	(String)		Nur bei Datei-Attributen (ab KP 3.9.29): Kommaseparierte Liste von Dateierweiterungen (z.B. "jpg,png") zur Einschränkung der Dateiauswahl im File-Upload-Applet (wirkt sich nicht auf das normale Dateiauswahl-Formularfeld aus).
autoSubmit	true/false	false	Nur bei Auswahlattributen (ab KP 3.9.6): Wenn dieser Render-Hint mit dem Wert "true" für ein Auswahlattribut gesetzt ist, so wird das Formular automatisch abgesendet, wenn ein Wert mit der Maus ausgewählt wird oder wenn ein Wert mit der Tastatur ausgewählt und das Eingabefeld verlassen wird (z.B. mittels der Tab-Taste).
collapseMessageKey	(String)	collection.collapseLink	
expandMessageKey	(String)	collection.expandLink	
hideLabel	true/false	false	ermöglicht das Ausblenden der Gruppenbeschreibung
isLimited	true/false	false	true -> Beschränkt die Anzahl der angezeigten Attribute
itemClass	(String)		Styleclass der einzelnen Attribute
itemSeparator	(String)		Trennzeichen, welches bei der Anzeige mehrerer Attribute zwischen den einzelnen Attributwerten angezeigt werden soll.
itemMode	(String)		Bestimmt den RenderMode für die einzelnen Attribute. Je nach Attributtyp können dies unterschiedliche Werte sein:
	<i>radio</i> oder <i>checkbox</i>	<i>radio</i>	Nur bei booleschen Attributen: Bestimmt ob das Attribut als Radiogroup oder Checkbox angezeigt werden soll



iterateFromIndex	(Integer)	0	
iterateToIndex	(Integer)	100000	
limit	(Integer)		Anzahl der anzuzeigenden Attribute
maxFileSize	(Long)		Nur bei Datei-Attributen (ab KP 3.9.29): Maximale Dateigröße als Einschränkung für die Dateiauswahl im File-Upload-Applet (wirkt sich nicht auf das normale Dateiauswahl-Formularfeld aus).
maxNumFiles	(Integer)		Nur bei Datei-Attributen (ab KP 3.9.29): Beschränkung der Anzahl der hochgeladenen Dateien. Wird dieser Render-Hint auf einen Wert größer als 0 gesetzt, so wird auf der Pflegeseite ein Hinweistext eingeblendet (Message-Key: edit.uploadMaxNumFiles) und das Dateiauswahl-Formularfeld (bzw. das Icon oder der Button zum Öffnen des File-Upload-Applets) wird deaktiviert, sobald die angegebene Anzahl von Dateien hochgeladen wurde. Die jeweils verbleibende Anzahl hochladbarer Dateien wird auch im File-Upload-Applet berücksichtigt, so dass der Benutzer auch mit dem Applet nicht mehr Dateien hochladen kann, als erlaubt.
required	true/false	false	Wird dieser Render-Hint auf den Wert "true" gesetzt, so wird das entsprechende Formularfeld mit einer Markierung versehen ("*") und der Benutzer muss einen Wert in dem Feld eingeben, bevor das Formular abgesendet (Klick auf Übernehmen/Speichern) oder ein anderer Reiter ausgewählt werden kann.
	Sprachcode (de, en, ...)		Ab KP 3.9.28: Wenn für das Attribut die Anzeige mehrere Eingabefelder zum Pflegen von verschiedenen Übersetzungen aktiviert ist (Attribut "lang"), dann kann als Wert dieses Render-Hints auch ein Sprachcode (z.B. "de" oder "en") angegeben werden, um festzulegen, dass der Benutzer nur in dem Eingabefeld für eine bestimmte Übersetzung einen Wert eintragen muss.



showApplet-Button	true/false	false	Nur für Datei-Attribute (ab KP 3.9.29): Wird dieser Render-Hint auf "true" gesetzt und ist das File-Upload-Applet aktiviert, so wird anstatt des normalen Dateiauswahl-Formularfeldes ein Button angezeigt, der beim Anklicken das File-Upload-Applet zur Auswahl und zum Hochladen von Dateien öffnet. Dies kann dann nützlich sein, wenn beim Upload bestimmte Einschränkungen gelten sollen, die sich nur mittels des File-Upload-Applets einschränken lassen (z.B. maximale Dateigröße oder Einschränkung auf bestimmte Dateitypen). Siehe hierzu auch "maxFileSize" und "acceptedFileExtensions" in dieser Tabelle.
showExistingUploadTags	true/false	true	Gibt an, ob für bereits existierende Datei-Attribute ein Dateiauswahl-Formularfeld zum Austauschen der hinterlegten Datei angezeigt werden soll (true) oder nicht (false).
showTooltip	true/false	true	Ist der Wert 'false', wird kein Tooltip angezeigt.
showWhen-Empty	true/false	false	false -> Blendet die Gruppe aus wenn kein Attribut existiert.
thumbnail-Width	(Integer)		Nur für Datei-Attribute (ab KP 3.9.29): Breite eines Vorschaubildes, welches für existierende Dateiattribute auf der Pflege-seite angezeigt werden soll, wenn es sich bei der hinterlegten Datei um eine Bilddatei handelt (Dateien mit Erweiterung "jpg", "jpeg", "png", "bmp" und "gif"). Dieser Render-Hint wird nur berücksichtigt, wenn auch die Höhe des Vorschaubildes über den Render-Hint "thumbnailHeight" definiert wurde.
thumbnail-Height	(Integer)		Nur für Datei-Attribute (ab KP 3.9.29): Höhe eines Vorschaubildes, welches für existierende Dateiattribute auf der Pflegeseite angezeigt werden soll. Dieser Render-Hint wird nur ausgewertet, wenn es sich bei der hinterlegten Datei um eine Bilddatei handelt (Dateien mit Erweiterung "jpg", "jpeg", "png", "bmp" und "gif") und wenn auch die Breite des Vorschaubildes über den Render-Hint "thumbnailWidth" definiert wurde.



thumbnailErrorPath	(String)	gfx/ image- Gallery/ not- Found.png	Nur für Datei-Attribute (ab KP 3.9.29): Pfad einer Bilddatei, die auf der Pflege- seite angezeigt werden soll, wenn der Download-Request für ein Vorschaubild in der angegebenen Größe (siehe Render- Hints "thumbnailWidth" und "thumbnail- Height") den HTTP-Response-Code "404 - Not found" zurückliefert.
uploadLimit- Bytes	(Long)		Nur für Datei-Attribute (ab KP 3.9.29): Wird dieser Render-Hint konfiguriert, so wird die Dateiauswahl im File-Upload- Applet auf das angegebene Datenvolu- men eingeschränkt. Auch wenn dieser Render-Hint konfig- uriert ist, kann der Benutzer beliebig viele Upload-Vorgänge durchführen, wenn nicht die maximale Dateianzahl limitiert ist (siehe Render-Hint "maxNumFiles"). Der Render-Hint wirkt sich nicht auf das normale Dateiauswahl-Formularfeld aus!

HINWEIS: Die Reihenfolge der Unterelemente muss in der XML-Definition mit der hier vorgegebene übereinstimmen.

Unterelement <ComponentActionTransformers>

Definiert eine Liste von Action-Transformern, die zur Erzeugung von Actions für den Attribut-
container verwendet werden sollen. Details zu den verfügbaren Action-Transformern sind
im Kapitel "Referenz" zu finden.

Unterelement <ItemActionTransformers>

Definiert eine Liste von Action-Transformern, die zur Erzeugung von Actions für alle angezeigten
Attribute ausgeführt werden sollen. Details zu den verfügbaren Action-Transformern sind
im Kapitel "Referenz" zu finden.

8.1.4 Relation

Attribute für den Tag <Relation>:

Attribut	Typ	Stan- dard	Beschreibung
id	(string)		Die ID, die auf der generierten Seite als ID am Relationscontainer generiert wird.
styleClass	(string)		Der Wert wird in das Attribut class am Tag des Relationscontainers angebracht.



renderMode	(string)		Der RenderMode für die Darstellung der Relationen.
sortKPath	(string)		Bestimmt die Sortierung der ermittelten Relationen.
sortAscending	true/false		Gibt an, ob die Sortierung aufsteigend oder absteigend erfolgen soll.
targetSearch-Name	(string)	relation-Target-Search	Interner Name einer Textsuche, die für die Bestimmung der Relationsziele dient.
autocomple-tion	true/false	true	Gibt an, ob die Autovervollständigung beim Anlegen einer neuen Relation aktiviert ist oder nicht.
autocomplete-SearchName	(string)		Interner Name einer Textsuche, die zur Bestimmung der Relationsziele verwendet wird. Ist kein Wert angegeben, so wird die Konfiguration von "targetSearchName" verwendet.
autocomple-tionLimit	(Zahl)	8	Die Anzahl der angezeigten Einträge in der Liste der bestehenden Relationsziele. Der Wert "0" schaltet die Anzeige der Relationsziele ab.
autocomple-tionDomain-Limit	(Zahl)	0	Die Anzahl der eingeblendeten Zieltypen, von denen neue Ziele angelegt werden können. Der Wert "0" besagt, dass keine Zieltypen angezeigt werden.
autocomple-tionMinLength	(Zahl)	3	Die minimale Anzahl Buchstaben, die eingegeben werden müssen, bevor Vorschläge angezeigt werden.
autocomple-tionDialogLink	true / false	false	Ist diese Einstellung aktiviert, so wird am Ende der Liste ein Link zum Öffnen der erweiterten Relationszielauswahl angeboten.
metasMode	off / lazy / direct	off	Bestimmt die Anzeige von Metaeigenschaften für die einzelnen Relationen.

Unterelemente <Label> oder <LabelKey>

Als Label wird ohne explizite Angabe der Name der Relation benutzt. Optional kann hier durch das Unterelement <Label> ein fester String gesetzt werden, oder durch die Angabe von <LabelKey> ein Schlüssel für eine Übersetzung angegeben werden. (Diese muss in der msg.properties definiert werden!)

Unterelement <Tooltip> oder <TooltipKey>

Konfiguration des Tooltips, der bei den Relationen angezeigt wird.

Unterelement <Valuetip> oder <ValuetipKey>

Konfiguration des Textes, der in der Eingabezeile einer Relation angezeigt wird, wenn noch



kein eigener Text eingegeben wurde.

Unterelement <InternalName> / <KPath>

Welche Relation angesprochen werden soll, wird über eines dieser Unterelemente definiert. Daher ist die Angabe des Unterelements <InternalName> oder <KPath> zwingend erforderlich.

Unterelement <RenderHint>

Über dieses Konfigurationselement können dem Renderer, der HTML-Quelltext generiert, verschiedenen Parameter übergeben werden.

Name	Possible	Default	Beschreibung
hideDeleteButton	true/false	false	false -> Blendet die Gruppe aus wenn kein Attribut existiert.
msg.edit.back	(String)	Key: <i>edit.back</i> Wert: Zurück	
msg.edit.createTarget	(String)	Key: <i>edit.createTarget</i> Wert: Ziel erzeugen	
msg.edit.findTargets	(String)	Key: <i>edit.findTargets</i> Wert: Relationsziel(e) suchen...	
msg.edit.findTargetsFor	(String)	Key: <i>edit.findTargetsFor</i> Wert: Relationsziel für "{1}" auswählen	
msg.edit.noPossibleTargets	(String)	Key: <i>edit.noPossibleTargets</i> Wert: Keine möglichen Ziele gefunden.	
msg.edit.setNewTarget	(String)	Key: <i>edit.setNewTarget</i> Wert: Begriff/Indiv. anlegen	
msg.edit.setNewTargetNoDomain	(String)	Key: <i>edit.setNewTargetNoDomain</i> Wert: Bitte wählen Sie die Art des neu anzulegenden Ziels aus.	
msg.edit.setNewTargetNoName	(String)	Key: <i>edit.setNewTargetNoName</i> Wert: Bitte geben Sie einen Namen für das neu anzulegende Ziel an.	
msg.edit.setTargets	(String)	Key: <i>edit.setTargets</i> Wert: Relationsziel(e) auswählen	



msg.edit.setTargets.hits	(String)	Key: <i>edit.setTargets.hits</i> Wert: Treffer	
msg.edit.setTargets.jumpToPage	(String)	Key: <i>edit.setTargets.jumpToPage</i> Wert: Springe zu Seite	
msg.edit.setTargets.nextPage	(String)	Key: <i>edit.setTargets.nextPage</i> Wert: Nächste Seite	
msg.edit.setTargets.noHits	(String)	Key: <i>edit.setTargets.noHits</i> Wert: Keine	
msg.edit.setTargets.page	(String)	Key: <i>edit.setTargets.page</i> Wert: Seite	
msg.edit.setTargets.pageOf	(String)	Key: <i>edit.setTargets.pageOf</i> Wert: von	
msg.edit.setTargets.prevPage	(String)	Key: <i>edit.setTargets.prevPage</i> Wert: Vorige Seite	
msg.error.setTarget.noTargetSelected	(String)	Key: <i>edit.setTargets.noTargetsSelected</i> Wert: Bitte wählen Sie mindestens ein Relationsziel aus!	
msg.topic.createInstance	(String)	Key: <i>edit.topic.createInstance</i> Wert: Neues Individuum erzeugen	
msg.topic.createSubconcept	(String)	Key: <i>edit.topic.createSubconcept</i> Wert: Unterbegriff erzeugen	
required (ab KP 3.9.50)	true/false	false	Ist der Wert 'true', so wird beim Absenden des Formulars (Button "Übernehmen" bzw. "Speichern") überprüft, ob eine Relation des angegebenen Typs existiert und ggf. eine Warnhinweis an dem entsprechenden Eingabefeld angezeigt.



showEmptyMessage	true/false	true	Ist der Wert 'false', so wird die Meldung 'Keine Einträge gefunden' nicht mehr angezeigt.
showTargetTooltips	true/false	false	Zusätzliches Stylesheet das für die Items verwendet werden soll
targetSearchName			ermöglicht das Ausblenden der Gruppenbeschreibung

HINWEIS: Die Reihenfolge der Unterelemente muss in der XML-Definition mit der hier vorgegebene übereinstimmen.

Unterelement <ComponentActionTransformers>

Definiert eine Liste von Action-Transformern, die zur Erzeugung von Actions für den Relation-
scontainer verwendet werden sollen. Details zu den verfügbaren Action-Transformern sind
im Kapitel "Referenz" zu finden.

Unterelement <ItemActionTransformers>

Definiert eine Liste von Action-Transformern, die zur Erzeugung von Actions für alle angezeigten
Relationen ausgeführt werden sollen. Details zu den verfügbaren Action-Transformern sind
im Kapitel "Referenz" zu finden.

8.1.5 RelationDropdown (ab Version 3.8.0)

Beschreibung

Mit Hilfe des Tags <RelationDropdown> können Relationen zu vorher festgelegten Zielen
angelegt werden. Diese Ziele werden in einer aufklappbaren Liste angezeigt ("dropdown").

Bestehende Relationen werden, wie beim Tag <Relation>, unterhalb der Auswahlbox aufgelistet.



Beispiel

```
<RelationDropdown>
  <Label>Dropdown-Liste</Label>
  <InternalName>myRelationInternalName</InternalName>
  <TargetKPath>
    // $myConceptInternalName$/allInstances()
  </TargetKPath>
  <TargetExpertQuery>myExpertQueryEID</TargetExpertQuery>
  <TargetGroupKPath sortKPath="label()" sortAscending="false">
    concept()/label()
  </TargetGroupKPath>
</RelationDropdown>
```

Unterelemente

Die folgenden Elemente werden genau so wie bei <Relation> konfiguriert.

- Label
- LabelKey
- LabelKPath
- Tooltip
- TooltipKey
- InternalName
- RenderHint



- ComponentActionTransformers
- ItemActionTransformers

Spezielle Unterelemente

Element	Typ	Beschreibung
TargetKPath	(string)	KPath-Ausdruck, der die Topics der Auswahlliste festlegt.
TargetExpertQuery	(string)	<p>Externe ID einer Expertensuche. Das Ergebnis der Suche ist die Ausgangsmenge der möglichen Ziele. Aus dieser werden nur die Elemente ausgewählt, die zur Relation und zur Rechtekonfiguration des angemeldeten Benutzers passen.</p> <p>Sind sowohl KPath als auch Expertensuche konfiguriert, so wird die Schnittmenge von beiden verwendet.</p>
TargetGroupKPath	(string)	<p>Optionaler KPath-Ausdruck, mit dem die Relationsziele gruppiert dargestellt werden können.</p> <p>Bei aktivierter Gruppierung werden Gruppennamen als nicht auswählbare, hervorgehobene Elemente oberhalb der zugehörigen Gruppenelemente angezeigt.</p> <p>Attribute am Tag <TargetGroupKPath>:</p> <ul style="list-style-type: none"> • sortKPath: Optionaler KPath-Ausdruck, der die Sortierung der Gruppen untereinander angibt. Standardwert: <i>label()</i> • sortAscending: Gibt an, ob die Sortierung der Gruppen aufsteigend erfolgen soll. Standardwert: <i>true</i>
OtherTargetsOption (ab Version 3.9.37)	(leer)	<p>Ist dieses Element vorhanden, wird dem Dropdown ein weiterer Eintrag hinzugefügt der bei der Auswahl eine erweiterte Relationszielsuche öffnet. Die erweiterte Relationszielsuche ist identisch zum Dialog des <Relation> Tags. Dabei gelten auch die gleichen RenderHints.</p> <p>Attribute:</p> <ul style="list-style-type: none"> • targetSearchName Name der erweiterten Relationszielsuche die verwendet werden soll • labelKey Message Key der im Dropdown als zusätzliche Auswahl angezeigt wird



Attribute für das Tag <RelationDropdown>:

Attribut	Typ	Standard	Beschreibung
id	(string)		Die ID, die auf der generierten Seite als ID am Relationscontainer generiert wird.
styleClass	(string)		Der Wert wird in das HTML-Attribut "class" am Tag des Relationscontainers angebracht.
renderMode	(string)		Der RenderMode für die Darstellung der Relationen.
sortKPath	(string)		Bestimmt die Sortierung der ermittelten Relationen. Die Menge der möglichen Ziele wird mit diesem KPath-Ausdruck sortiert.
sortAscending	true/false		Gibt an, ob die Sortierung aufsteigend oder absteigend erfolgen soll.
metasMode	off / lazy / direct	off	Bestimmt die Anzeige von Metaeigenschaften für die einzelnen Relationen.

Render-Hints für <RelationDropdown>

Die folgenden Render-Hints werden beim Rendering der RelationDropdown-Komponente ausgewertet:

Attribut	Typ	Standard	Beschreibung
hideLabel	true/false	false	Gibt an, ob die Beschriftung der Komponente (definiert durch die XML-Elemente "Label", "LabelKey" und/oder "LabelKPath" in der Edit-Config) angezeigt werden soll (false), oder nicht (true).
required (ab KP 3.9.50)	true/false	false	Ist der Wert 'true', so wird beim Absenden des Formulars (Button "Übernehmen" bzw. "Speichern") überprüft, ob eine Relation des angegebenen Typs existiert und ggf. eine Warnhinweis an dem entsprechenden Eingabefeld angezeigt.



showEmptyMessage	true/false	true	Für Relationen, die mehrfach vorkommen können: Gibt an, ob ein Text angezeigt werden soll (Message-Key "collection.empty"), wenn noch keine Relationen vom konfigurierten Konzept am aktuell bearbeiteten Topic existieren.
showTooltip	true/false	true	Dieser Render-Hint gibt an, ob für diese Komponente das konfigurierte Tooltip (definiert durch XML-Element "Tooltip" in der Edit-Config) angezeigt werden soll (, oder nicht.
showWhenEmpty	true/false	false	Gibt an, ob die Komponente auch dann angezeigt werden soll, wenn keine möglichen und auch keine existierenden Relationsziele vorhanden sind.

8.1.6 Relation Wizard (ab Version 3.9)

Beschreibung

Mit Hilfe des Tags `<RelationWizard>` wird das Anlegen von Relationen über eine gefilterte Ansicht ermöglicht. Die Ziele können in einer konfigurierbaren, tabellarischen Ansicht durchgesehen werden. Dabei ist es möglich, die Zielmenge durch Filtereinstellungen einzuschränken.

Der Einsatz von `<RelationWizard>` eignet sich besonders für eine große Menge an möglichen Relationszielen.

Beispiel

```
<RelationWizard>  
  <InternalName>bearbeitet_Auftrag</InternalName>  
</RelationWizard>
```

Diese Konfiguration legt fest, dass für die Relation mit dem internen Namen "bearbeitet_auftrag" die Auswahl per RelationWizard erfolgt. Es wird die Objektlistenkonfiguration verwendet, die automatisch über den Typ der möglichen Relationsziele ermittelt werden kann.

Unterelemente von `<RelationWizard>`

Die folgenden Elemente werden an einem RelationWizard Tag unterstützt. Ist keine gesonderte Erklärung angefügt, so werden diese Elemente genau so wie bei `<Relation>` konfiguriert.

- Label
- LabelKey
- LabelKPath
- Tooltip



- TooltipKey
- TargetKPath
Die Ergebnisse dieses KPath-Ausdrucks definieren die maximale Menge der angezeigten Tabellenelemente. Topics, die als Relationsziel nicht in Frage kommen, werden ignoriert.
- InternalName
- RenderHint
- ComponentActionTransformers
- ItemActionTransformers

Attribute von <RelationWizard>

Attribut	Typ	Standard	Beschreibung
tableName	(string)		Optional. Der Name der Objektlistenkonfiguration, die verwendet werden soll. Die Liste wird ausgehend vom Typ des Relationsziels an diesem Begriff und allen Oberbegriffen gesucht. Hat die Relation mehrere Zieltypen, so wird das Konzept als Startpunkt verwendet, das alle Konzepte unter sich vereinigt. Wird kein Wert angegeben, so wird versucht, die am besten passende Konfiguration automatisch zu ermitteln.
alternativeTableName	(string)		Optional. Wird angegeben, wenn statt der Hauptkonfiguration eine alternative Konfiguration verwendet werden soll.
sortColumns	(string)		Die Spalten, die initial sortiert werden sollen. Gleiche Bedeutung wie die Eigenschaft beim ViewConfig Tag <ObjectList>.
pageSize	Ganzzahl	10	Die Seitengröße der Tabelle. Wird nichts angegeben, so wird ein Menü mit einer Auswahl an möglichen Seitengrößen eingeblendet.
initialLoad	true / false	true	Gibt an, ob die Daten der Tabelle schon beim ersten Öffnen angezeigt werden sollen oder erst nach Anfrage durch den Benutzer.



searchable	true / false	true	Gibt an, ob die Tabelle über Filtereingaben im Tabellenkopf verfügen soll oder nicht.
autoSearchOnTab	true / false	true	Gibt ab, ob das Verlassen einer Filtereingabe per Tabulatortaste einen Suchvorgang auslösen soll oder nicht.
autoSearchMinLength	Ganzzahl	3	Die Mindestmenge an Buchstaben, die in einer Filtereingabe eingegeben sein muss, damit ein automatische Suchvorgang ausgelöst wird.
multiselect	true / false	false	Aktiviert die Mehrfachauswahl von Einträgen, um mehr als eine Relationen innerhalb einer RelationWizard-Sitzung ziehen zu können.
allItemsLabel (ab KP 5.0.1)	(string)	relationWizard.button.allItems	Message-Key für Button zur Verknüpfung mit allen Zielen (bei Relationen die mehrfach vorkommen können)
cancelButtonLabel (ab KP 5.0.1)	(string)	button.cancel	Message-Key für Abbrechen-Button
createButtonLabel (ab KP 5.0.1)	(string)	button.create	Message-Key für Button zur Verknüpfung mit ausgewähltem Ziel
selectedItemsLabel (ab KP 5.0.1)	(string)	relationWizard.button.selectedItems	Message-Key für Button zur Verknüpfung mit ausgewählten Zielen (bei Relationen die mehrfach vorkommen können)

Render-Hints von <RelationWizard>

Name	Possible	Default	Beschreibung
hideLabel	true/false	false	Gibt an, ob die Beschriftung des Formularfeldes zum Aufruf des Relation-Wizard ausgeblendet werden soll (true) oder nicht (false).



required (ab KP 3.9.50)	true/false	false	Ist der Wert 'true', so wird beim Absenden des Formulars (Button "Übernehmen" bzw. "Speichern") überprüft, ob eine Relation des angegebenen Typs existiert und ggf. eine Warnhinweis an dem entsprechenden Eingabefeld angezeigt.
showEmptyMessage	true/false	true	Ist der Wert 'false', so wird die Meldung 'Keine Einträge gefunden' nicht mehr angezeigt.
showTooltip	true/false	true	Gibt an, ob der Tooltip mit dem über die Attribute "Tooltip" oder "TooltipKey" konfigurierten Text am Eingabefeld angezeigt werden soll (true) oder nicht (false).

8.1.7 AllExtensions

Der Tag <AllExtensions> zeigt alle Erweiterungen des aktuellen Individuums an und erlaubt das Anlegen weiterer Erweiterungen.

Attribute für den Tag <AllExtensions>:

Attribut	Typ	Standard	Beschreibung
styleClass	(string)		Der Wert wird in das Attribut class am Tag des Relationscontainers angebracht.
renderMode	(string)		Der RenderMode für die Darstellung der Relationen.
sortKPath	(string)		Bestimmt die Sortierung der ermittelten Relationen.
sortAscending	true/false		Gibt an, ob die Sortierung aufsteigend oder absteigend erfolgen soll.



layoutCategory	(string)		Die Layout-Kategorie der Erweiterungen wird ignoriert. Die Konfiguration für die Darstellung der Erweiterung wird über die angegebene Layout-Kategorie geladen.
showChildren-Collapsed	true/false	false	Gibt an, ob die einzelnen Erweiterungen zunächst eingeklappt angezeigt werden sollen.
expand-FirstChild	true/false	false	Gibt an, ob die erste Erweiterung ausgeklappt werden soll.
createOnSelect	true/false	false	"true" verursacht folgendes Verhalten: 1) Der Knopf zum Anlegen neuer Erweiterungen wird ausgeblendet. 2) Wird im Menü ein Erweiterungsbegriff ausgewählt, so wird sofort eine Erweiterung dieses Typs angelegt.

Unterelement <RenderHint>

Über dieses Konfigurationselement können dem Renderer, der HTML-Quelltext generiert, verschiedenen Parameter übergeben werden.

Name	Possible	Default	Beschreibung
hideDeleteButton	true/false	false	false -> Blendet die Gruppe aus wenn kein Attribut existiert.
required (ab KP 3.9.50)	true/false	false	Ist der Wert 'true', so wird beim Absenden des Formulars (Button "Übernehmen" bzw. "Speichern") überprüft, ob eine Relation des angegebenen Typs existiert und ggf. eine Warnhinweis an dem entsprechenden Eingabefeld angezeigt. Ist der Wert 'false' (Standardwert), so kann das Formular auch abgesendet werden, ohne dass eine Relation des entsprechenden Typs existiert.



showEmptyMessage	true/false	true	Ist der Wert 'false', so wird die Meldung 'Keine Einträge gefunden' nicht mehr angezeigt.
showTargetTooltips	true/false	false	Zusätzliches Stylesheet das für die Items verwendet werden soll
targetSearchName			ermöglicht das Ausblenden der Gruppenbeschreibung

Unterelement <ComponentActionTransformers>

Konfiguration von Actions, die direkt im Header des AllExtensions-Elements angezeigt werden. Details zu den verfügbaren Action-Transformern sind im Kapitel "Referenz" zu finden.

Unterelement <SubcomponentActionTransformers>

Konfiguration von Actions, die an jeder Extension angezeigt werden. Details zu den verfügbaren Action-Transformern sind im Kapitel "Referenz" zu finden.

HINWEIS: Die Reihenfolge der Unterelemente muss in der XML-Definition mit der hier vorgegebene übereinstimmen.

8.2 Sperren

Allgemeines

Ab Version 3.8 werden Sperren unterstützt.

Sperren verhindern, dass ein Topic von mehreren Benutzern gleichzeitig bearbeitet werden kann. Öffnet ein Anwender die Pflegeseite eines Topics, so wird dieses für nachfolgende Änderungen durch andere Personen gesperrt.

Versucht ein Anwender, ein gesperrtes Topic zu bearbeiten, so wird im eine entsprechende Meldung angezeigt, die ihn über den Grund der Sperrung informiert.

Sperren werden an diesen Stellen überprüft:

- Öffnen der Pflegeseite
- Löschen eines Topics
- Ziehen einer Relation per MicroEdit-Box
- Relationszielsuche beim Anlegen von Relationen (MicroEdit und Pflegeseiten)
- Löschen von Relationen in MicroEdit-Boxen

Konfiguration in K-Infinity

Sperren können pro Wissensnetz an- und ausgeschaltet werden.

Im Knowledge-Builder können Sperren von Administratoren für ein Netz an- und ausgeschaltet werden. Die Einstellung befindet sich unter *Datei > Einstellungen* im Reiter *Sperren*.

Sind Sperren aktiviert, so werden hier alle aktuell gesperrten Topics aufgelistet. Falls Sperren offengeblieben sind, bspw. durch kritische Fehler, so können diese hier entfernt werden, um



Benutzern das weitere Arbeiten mit den betroffenen Topics zu ermöglichen.

Das Knowledge-Portal enthält jedoch bereits einige Vorkehrungen, um im Fehlerfalle Sperren korrekt zu entfernen.

Konfiguration im Knowledge-Portal

Im Portal selbst ist keine weitere Konfiguration für die Verwendung von Sperren notwendig. Die Aktivierung im Knowledge-Builder wird vom Portal ausgelesen und angewendet.

8.3 "find or create" Funktionalität

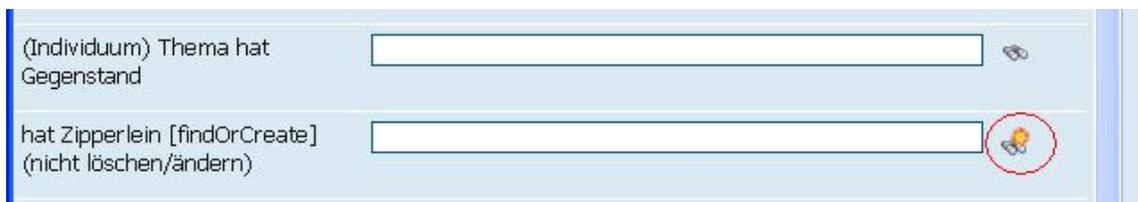
Die "find or create" Funktionalität ermöglicht das automatische Erzeugen von noch nicht vorhandenen Relationszielen direkt aus der Pflegeansicht heraus. Es können dabei sowohl Individuen als auch Unterbegriffe erzeugt werden, abhängig von der Netzkonfiguration.

Es existieren zwei Modi: **normal** und **brainstorming**.

Normal-Modus: nach Eingabe eines nicht vorhandenen Relationsziels wird eine neue Seite angezeigt, auf der der Benutzer die Möglichkeit hat, das Ziel anzulegen oder eventuell Schreibfehler zu korrigieren und eine neue Relationszielsuche zu starten.

Brainstorming-Modus: das Relationsziel wird sofort erzeugt, falls es nicht bereits existiert.

Ob die Funktionalität zu Verfügung steht, wird durch ein spezielles Icon angezeigt.



8.3.1 Netzkonfiguration

Um die "find or create" Funktionalität verwenden zu können, müssen im Begriffseditor des Relationsbegriffs "Benutzerrelation" unter "Schemadefinition Begriff" folgende Attribute angelegt werden:

Interner Name	Typ
findOrCreateInstance	Boolean
brainStormingMode	Boolean

An der gleichen Stelle folgende Relation anlegen:



Interner Name	Quelle	Ziel
findOrCreateTargetConcept	Unterbegriffe von Benutzerrelation (Begriff)	Unterbegriffe von Wurzelbegriff (Begriff)

Um die Funktionalität für eine bestimmte Relation zu aktivieren, muss zunächst festgelegt werden, welcher Begriff für das Anlegen des Relationsziels verwendet werden soll. Dazu im Begriffseditor der Relation unter "Begriff" die Relation mit dem internen Namen "findOrCreateTargetConcept" hinzufügen und als Ziel den Begriff auswählen, von dem Unterbegriffe oder Individuen erzeugt werden sollen.

Über "findOrCreateInstance" wird festgelegt, ob ein Individuum oder ein Unterbegriff erzeugt wird. true -> Individuum, false -> Unterbegriff.

Über "brainStormingMode" wird der Brainstorming-Modus an- bzw. ausgeschaltet.

8.3.2 Anwendungskonfiguration

In der names.xml müssen folgende Definitionen hinterlegt sein:

```
<findOrCreate>
  <kPathCreateInstance>
    concept()/@$findOrCreateInstance$/rawValue()
  </kPathCreateInstance>
  <kPathBrainStormingMode>
    concept()/@$brainStormingMode$/rawValue()
  </kPathBrainStormingMode>
  <kPathTargetConcept>
    concept()/~$findOrCreateTargetConcept$/target()
  </kPathTargetConcept>
</findOrCreate>
```

Diese Einstellung sollte per default vorhanden sein.



8.4 Hochladen von Dateien mit dem Multi-Upload-Applet

Wenn auf den Knowledge-Portal-Pflegeseiten viele Dateien hochgeladen werden müssen, kann das sog. Multi-Upload-Applet aktiviert werden. Das Multi-Upload-Applet ermöglicht es den Anwendern mehrere Dateien "auf einen Rutsch" hochzuladen. Außerdem ermöglicht das Applet auch den Upload von sehr großen Dateien, die nicht in einem einzelnen HTTP-Post-Request zum Webserver übertragen werden können.

Das Multi-Upload-Applet kann aktiviert werden, indem der Konfigurationsschlüssel "kpConfig.topicForm.useMultiFileUpload" in der Knowledge-Portal-Konfiguration auf den Wert "true" gesetzt wird. Um den Konfigurationsschlüssel in der KP-Konfiguration im Wissensnetz anzulegen, bitte folgendermaßen vorgehen:

1. Unterbegriff mit Namen "topicForm" als Unterbegriff von KP-Konfiguration anlegen
2. In der Schemadefinition Begriff am Begriff "topicForm" ein Attribut vom Typ Boolean mit dem Namen "useMultiFileUpload" anlegen
3. Am Begriff "topicForm" das Attribut "useMultiFileUpload" anlegen und die Checkbox an dem neu angelegten Attribut anhaken.
4. Mit dem Tomcat-Manager einen Reload der Knowledge-Portal-Web-Anwendung durchführen oder den Tomcat neu starten

Ist das Multi-Upload-Applet aktiviert, so wird auf der Pflegeseite rechts neben Datei-Attributen das Icon  angezeigt sofern:

- das entsprechende Attribut mehrfach vorkommen darf
- der Benutzer das Recht zum Anlegen des entsprechenden Attributs hat (siehe dazu K-Infinity-Handbuch Stichwort "Rechtesystem").

Ein Klick auf das Icon öffnet das Multi-Upload-Applet.

8.5 Bearbeiten von Dateien mit dem Change-File-Applet

Mit dem Change-File-Applet können Dokumente, die im Wissensnetz abgespeichert sind, auf einfache Weise bearbeitet werden, wobei das Applet die folgenden Arbeitsschritte, die normalerweise per Hand ausgeführt werden müssten, übernimmt:

- Herunterladen und Abspeichern der Datei aus dem Wissensnetz
- Öffnen der Datei
- Überschreiben der Datei im Wissensnetz nach dem Bearbeiten mit der geänderten Version

Das Applet kann aktiviert werden, indem der Konfigurationsschlüssel "kpConfig.topicForm.useChangeFileApplet" in der Knowledge-Portal-Konfiguration auf den Wert "true" gesetzt wird. Um das Applet über die Knowledge-Portal-Konfiguration im Wissensnetz zu aktivieren, muss folgendermaßen vorgegangen werden:



1. Unterbegriff mit Namen "topicForm" als Unterbegriff von KP-Konfiguration anlegen
2. In der Schemadefinition Begriff am Begriff "topicForm" ein Attribut vom Typ Boolean mit dem Namen "useChangeFileApplet" anlegen
3. Am Begriff "topicForm" das Attribut "useChangeFileApplet" anlegen und die Checkbox an dem neu angelegten Attribut anhaken.
4. Mit dem Tomcat-Manager einen Reload der Knowledge-Portal-Web-Anwendung durchführen oder den Tomcat neu starten

Ist das Change-File-Applet aktiviert, so wird auf der Pflegeseite rechts neben jedem Datei-Attribut das Icon  angezeigt, sofern das Rechtesystem dem Benutzer die Bearbeitung erlaubt.

Ein Klick auf das Icon startet das Change-File-Applet, welches sofort die jeweilige Datei auf den Computer des Benutzers herunterlädt und das Dokument mit der zum Dateityp passenden Applikation öffnet. Hat der Anwender alle Änderungen an der Datei vollzogen und abgespeichert, kann er die geänderte Datei über ein vom Change-File-Applet geöffnetes Dialogfenster wieder ins Wissensnetz hochladen. Die alte Version der Datei im Wissensnetz wird hierbei überschrieben.

8.6 Tooltips

9 Die Collections

Collections dienen wie auch die Favoriten dazu, Objekte aus dem Trace System zu sammeln. Der Einsatzbereich von Collections und Favoriten unterscheidet sich in folgenden Aspekten:

- Die Strukturierung von Objekten in Gruppen
- Das Teilen mit anderen Nutzern

9.1 Das Sammeln von Objekten

Collections sind **Sammelmappen** für Objekte. Im Gegensatz zu den Favoriten können Collections mit anderen Nutzern geteilt und zum gemeinsamen Arbeiten verwendet werden. Dafür können Collections anderen Mitarbeitern zugewiesen werden. Des Weiteren ist es möglich, mehrere Collections zu haben.

Die aktive Collection wird im linken Interaktionsbereich angezeigt. Sie stellt die Drop-Zone der aktuellen Collection dar. Möchte man der Collection neue Objekte hinzufügen, zieht man das Icon des gewünschten Objekts in die Drop-Zone.



Produkt-Variante

PV14675 - Kranwagen - I

Benutzer

Alles rund um den Kranwagen

Projekt Kranwagen

Details Freigabe Stückliste Chargen (1) Prozesse

Bild

Hat man das Objekt-Icon in die Drop-Zone gezogen, öffnet sich ein Fenster. Dort kann man eine Bemerkung zu dem ausgewählten Objekt eintragen. Die Bemerkung wird an dem Objekt in der Sammelmappe erstellt und nicht an dem Objekt, das man in die Collection gezogen hat.

Sammelmappe: PM Kranwagen



Bemerkung

14675 - Kranwagen basiert auf dem Produktmodell Kranwagen

Collect

Abbrechen

Durch Klick auf den Collectionname *Alles rund um den Kranwagen* über der Drop-Zone wird die Detailseite der entsprechenden Collection aufgerufen. Auf der Detailseite einer Collection werden alle Einträge der Collection angezeigt, zusätzlich ist es möglich, eine Beschreibung für die Collection hinzuzufügen. Außerdem können die gesammelten Objekte angesehen, bearbeitet und gelöscht werden.



Collection

Alles rund um den Kranwagen



Klicken Sie doppelt auf dieses Feld, um den Text-Editor zu öffnen.

PV14675 - Kranwagen - made in China
Produktvariante für PV14675 Kranwagen
[..]
PM Kranwagen
PV14675 Kranwagen basiert auf dem Produktmodell Kranwagen
[..]
Projekt Kranwagen
Projekt zu PV14675 Kranwagen
[..]

9.2 Die Verwaltung von Collections

Die Verwaltung von eigenen Collections findet auf dem Reiter *Collection* der Personen-Detailseite statt. Dort werden alle Collections des Nutzers angezeigt.

Experte für (0)	Produktmanager für (0)	Produktentwickler für (0)	Freigaben (0)	Aufgaben (0)	Collection (2)			
+ Neue Collection erstellen				Erster	Zurück	1	Nächster	Letzter
Name	gehört Person	zugeordnete Personen	Collection Items					
Suchen in Name...	Suchen in gehört Person...	Suchen in zugeordnete Persc...	Suchen in Collection Items...					
Alles rund um den Bagger	Schulze, Kathrin	Schnitter, Thorsten	70488 - Bagger,Produktcharge PV70488 - Bagger - made in Germany - 2805-131-15,PV70488 - Bagger - made in Germany					
Alles rund um den Kranwagen	Schulze, Kathrin		PM Kranwagen,Projekt Kranwagen,PV14675 - Kranwagen - made in China					
1 bis 2 von 2 Einträgen				Erster	Zurück	1	Nächster	Letzter

Mit dem Icon neben der Collection kann diese aktiviert werden. Es wird immer nur die aktive Collection eines Nutzers im Drop-Down-Bereich angezeigt. Um eine Collection bearbeiten zu können, muss diese zuerst aktiviert werden.

9.3 Kontextinformationen zu Collections

Auf der Detailseite einer Collection stehen verschiedene Kontextinformationen zur Verfügung:



Collection gehört (1) ▶

Collection ist Mitarbeiter zugewiesen (1) ▶

Eigene Collections (2) ▼

 **Alles rund um den Bagger** 

 **Alles rund um den Kranwagen** 

Zugewiesene Collections (1) ▶

Collection gehört

Die angezeigte Collection gehört der in der Kontextbox angezeigten Person.

Collection ist Mitarbeiter zugewiesen

Für die Zusammenarbeit kann die Collection mit anderen Personen geteilt werden, dafür kann die Kontextbox verwendet werden. Wurde die Collection einem Mitarbeiter zugewiesen, wird sie in der Box *Zugewiesene Collections* bei dem Mitarbeiter angezeigt.

Eigene Collections

Alle eigenen Collections werden angezeigt. Auf der Personendetailseite können diese verwaltet werden.

Zugewiesene Collections

Alle Collections, die der angemeldeten Person für eine Zusammenarbeit zugewiesen wurden. Die Collections wurden von anderen Personen erstellt, können aber weiter bearbeitet werden.

10 Ereignisprotokollierung

10.1 Überblick

Die Ereignisprotokollierung im Knowledge-Portal zeichnet zwei unterschiedliche Arten von Ereignissen auf:

- Suchanfragen
- Aufruf von Detailseiten zu Begriffen/Individuen

Für Suchanfragen werden die folgenden Informationen protokolliert:



- Zeitstempel
- Name der Suche
- Query String
- Anzahl der Treffer

Beim Aufrufen von Detailseiten werden nur ein Zeitstempel und die interne ID des angezeigten Topics (Domain Model ID, auch DMID genannt) gespeichert.

Die Ereignisse werden in Tabellenform protokolliert, damit das Protokoll leichter in Tabellenkalkulationsprogrammen wie z.B. Microsoft Excel weiterverarbeitet werden kann.

Standardmäßig werden die Ereignisse im CSV-Format protokolliert und direkt im Tomcat-Logs-Verzeichnis gespeichert (CSV-Dateien können direkt in Excel importiert werden). Wenn eine einfachere Auswertung der Protokolle direkt im Knowledge-Portal möglich sein soll, kann wahlweise aber auch die Protokollierung in eine Datenbank aktiviert werden. In diesem Fall werden die Ereignisse in einer SQL-Datenbank gespeichert, die direkt in das Knowledge-Portal eingebettet ist.

10.2 Protokollierung in CSV-Dateien

Bei dieser Form der Protokollierung werden die Ereignisse in dem textbasierten Tabellenformat CSV (Character separated values) abgespeichert.

Suchereignisse werden in die Datei "kp_search.log" protokolliert und die Aufrufe von Detailseiten in die Datei "kp_navigation.log". Die Protokolldateien werden standardmäßig im Tomcat-Logs-Verzeichnis gespeichert.

Beide Protokolldateien werden pro Kalenderwoche gerollt, d.h. wenn ein neues Ereignis in einer anderen Kalenderwoche stattfindet als das zuletzt protokollierte Ereignis, so wird die Protokolldatei zunächst umbenannt und der Dateiname mit Jahr und Nummer der aktuellen Kalenderwoche als Zusatz versehen (z.B. "kp_navigation.log.2008-10"). Das neue Ereignis wird dann wieder in eine leere "kp_navigation.log" oder "kp_search.log" protokolliert. Dadurch werden die Protokolldateien nicht so groß und man kann leichter auf die Protokoll für einen bestimmten Zeitraum zugreifen.

CSV-Dateien können direkt in Excel geöffnet werden, sofern die Datei die Endung ".csv" hat. Um die Log-Dateien aus dem Knowledge-Portal in Excel zu öffnen muss also an den Dateinamen noch die Endung ".csv" angehängt werden.

Zur Aktivierung der Ereignisprotokollierung in CSV-Dateien müssen die folgenden Schritte ausgeführt werden:

1. Anlegen des Begriffs "eventLog" als Unterbegriff von KP-Konfiguration (im Wissensnetz mit dem Knowledge-Builder)
2. Definieren des Attributs "logger" am Begriff "eventLog" (Zeichenketten-Attribut; in der Schemadefinition auf Begriffsseite)
3. Anlegen des Attributs "logger" am Begriff "eventLog". Attributwert:
com.iviews.knowledgeportal.logging.CommonsLogger
4. Neustart des Tomcat-Webservers



10.3 Protokollierung in eine Datenbank

10.3.1 Einführung

Durch die Aktivierung der Datenbank-Protokollierung werden die Ereignisse in einer SQL-Datenbank gespeichert. Standardmäßig wird als Datenbank eine direkt im Knowledge-Portal integrierte SQL-Datenbank verwendet.

Mit der Aktivierung der Datenbank-Protokollierung erhält man die Möglichkeit, direkt im Knowledge-Portal-Web-Frotend vordefinierte Auswertungen aufzurufen.

Die Auswertungen können über eine Box aufgerufen werden, die im Knowledge-Portal ganz unten in der linken Spalte angezeigt wird. Da diese Auswertungen nur für Administratoren gedacht sind, wird die Anzeige der Box über eine benutzerdefinierte Operation namens "can-QueryEventLog" im K-Infinity-Rechtesystem gesteuert. Bei aktiviertem Rechtesystem muss daher u.U. eine Regel definiert werden, damit die Box angezeigt wird. Bei deaktiviertem Rechtesystem wird die Box immer angezeigt.

Eine Auswertung ist eine in der Knowledge-Portal-Konfiguration hinterlegte SQL-Anweisung, die sich auf Tabellen aus der Protokollierungsdatenbank beziehen muss. In die SQL-Anweisung können Platzhalter eingefügt werden, die bei der Ausführung der Auswertung über ein automatisch generiertes Eingabeformular vom Benutzer abgefragt werden.

Standardmäßig werden die folgenden Auswertungen angeboten:

- Top X Begriffe: Die am häufigsten geöffneten Begriffe
- Top X Suchen: Die am häufigsten gestellten Suchanfragen
- Null-Suchen: Suchanfragen die keine Treffer geliefert haben

10.3.2 Aktivieren der Datenbank-Protokollierung (bis KP 3.5)

Zur Aktivierung der Datenbank-Protokollierung müssen die folgenden Schritte ausgeführt werden:

1. Anlegen des Begriffs "eventLog" als Unterbegriff von KP-Konfiguration (im Wissensnetz mit dem Knowledge-Builder)
2. Definieren des Attributs "logger" am Begriff "eventLog" (Zeichenketten-Attribut; in der Schemadefinition auf Begriffsseite)
3. Anlegen des Attributs "logger" am Begriff "eventLog". Attributwert:
com.iviews.knowledgeportal.logging.jdbc.JDBCLogger
4. Neustart des Tomcat-Webservers



10.3.3 Aktivieren der Datenbank-Protokollierung (ab KP 3.6)

Um die Ereignisprotokollierung in eine Datenbank im KP ab Version 3.6 zu aktivieren, muss im Attribut "componentsXml" in der KP-Konfiguration das Element "JDBCEventLogger" (Unterlement von "ApplicationComponents > EventLogging") eingetragen werden.

Unterhalb des JDBCEventLogger-Elements können mit den Elementen "queries" und "query" parametrisierbare SQL-Abfragen definiert werden, die für berechtigte Benutzer in der Box "Benutzerauswertungen" aufrufbar sind.

Standardmäßig sind im KP die folgenden Benutzerauswertungen konfiguriert:

```
<JDBCEventLogger>
  <queries>
    <query name="topXsearch">
      <parameter name="from" type="timestamp"/>
      <parameter name="to" type="timestamp"/>
      <parameter name="limit" type="number"
        defaultValue="10"/>
      <sql>
        SELECT count(query) AS ccount, query, search_name
          FROM search_events
          WHERE timestamp >= '${from}' AND timestamp &lt;= '${to}'
          GROUP BY query,search_name
          ORDER BY ccount DESC, query ASC
      </sql>
    </query>
    <query name="topXclicks">
      <parameter name="from" type="timestamp"/>
      <parameter name="to" type="timestamp"/>
      <parameter name="limit" type="number" defaultValue="10"/>
      <sql>
        SELECT count(object_dmid) AS ccount, object_dmid, object_label
          FROM navigate_events
          WHERE timestamp >= '${from}' AND timestamp &lt;= '${to}'
          GROUP BY object_dmid,object_label
          ORDER BY ccount DESC, object_label ASC
      </sql>
    </query>
    <query name="zeroSearches">
      <parameter name="from" type="timestamp"/>
      <parameter name="to" type="timestamp"/>
      <parameter name="limit" type="number" defaultValue="10"/>
      <sql>
        SELECT *
          FROM search_events
          WHERE result_size=0 and timestamp >= '${from}'
            AND timestamp &lt;= '${to}'
      </sql>
    </query>
  </queries>
</JDBCEventLogger>
```

Der Parameter "limit" dient (falls definiert) zur Beschränkung der Zeilenanzahl in der Ergeb-



nistabelle. Der Parameter spielt eine Sonderrolle, indem er nicht einfach in das SQL-Statement eingesetzt wird, sondern stattdessen vom KP in den JDBC-Aufrufen zum Abfragen der Ergebnistabelle eingesetzt wird.

10.3.4 Datenbank-Schema

Hier ein Überblick über die Datenbank-Tabellen, in denen die KP-Ereignisse protokolliert werden (Stand: KP 3.8.14):

Tabelle "SEARCH_EVENTS"

Spaltenname	SQL-Typ (aus CREATE statement)	Beschreibung
id	INT primary key generated always as identity	ID des Ereignisses
timestamp	TIMESTAMP	Zeitpunkt des Ereignisses
search_name	VARCHAR(32672)	Name der Suchkonfiguration
query	VARCHAR(32672)	Sucheingabe
result_size	INTEGER	Anzahl der Treffer
user_name	VARCHAR(64)	Name des Benutzers, der die Aktion ausgelöst hat (wird nur protokolliert, wenn das Attribut "logUser" am EventLogger-Element in der componentsXml den Wert "true" hat).
milliseconds	INTEGER	Dauer der Suche in Millisekunden

Tabelle "NAVIGATE_EVENTS"

Spaltenname	SQL-Typ (aus CREATE statement)	Beschreibung
id	INT primary key generated always as identity	ID des Ereignisses
timestamp	TIMESTAMP	Zeitpunkt des Ereignisses
object_dmid	VARCHAR(255)	K-Infinity Domain Model ID des Objektes, dessen Detailseite aufgerufen wurde.



object_label	VARCHAR(32672)	Name des Objektes, dessen Detailseite aufgerufen wurde
user_name	VARCHAR(64)	Name des Benutzers, der die Aktion ausgelöst hat (wird nur protokolliert, wenn das Attribut "logUser" am EventLogger-Element in der componentsXml den Wert "true" hat).

Tabelle "USER_LOGINS"

Spaltenname	SQL-Typ (aus CREATE statement)	Beschreibung
id	INT primary key generated always as identity	ID des Ereignisses
timestamp	TIMESTAMP	Zeitpunkt des Ereignisses
user_dmid	VARCHAR(64)	K-Infinity Domain Model ID des Benutzers, der sich angemeldet hat
user_id	VARCHAR(64)	Login-Name des Benutzers, der sich angemeldet hat.
user_name	VARCHAR(64)	Name des Benutzers, der sich angemeldet hat.

Tabelle "TOPIC_CHANGED_EVENTS"

Spaltenname	SQL-Typ (aus CREATE statement)	Beschreibung
id	INT primary key generated always as identity	ID des Ereignisses
timestamp	TIMESTAMP	Zeitpunkt des Ereignisses
object_dmid	VARCHAR(255)	K-Infinity Domain Model ID des Objekts, das verändert wurde



source	VARCHAR(255)	Bezeichner für die Quelle des Ereignisses (siehe unten)
user_name	VARCHAR(64)	Name des Benutzers, der das Objekt verändert hat.

Folgende Werte können in der Source-Spalte vorkommen:

- editPage: Das Ereignis wurde beim Speichern einer Pflegeseite ausgelöst.

Tabelle "NEW_TOPIC_EVENTS"

Spaltenname	SQL-Typ (aus CREATE statement)	Beschreibung
id	INT primary key generated always as identity	ID des Ereignisses
timestamp	TIMESTAMP	Zeitpunkt des Ereignisses
object_dmid	VARCHAR(255)	K-Infinity Domain Model ID des Objekts, das neu angelegt wurde.
source	VARCHAR(255)	Bezeichner für die Quelle des Ereignisses (siehe unten)
user_name	VARCHAR(64)	Name des Benutzers, der das Objekt angelegt hat.

Folgende Werte können in der Source-Spalte vorkommen:

- editPage: Das Ereignis wurde beim Speichern einer Pflegeseite ausgelöst.

11 Problemlösung

11.1 Quick Search

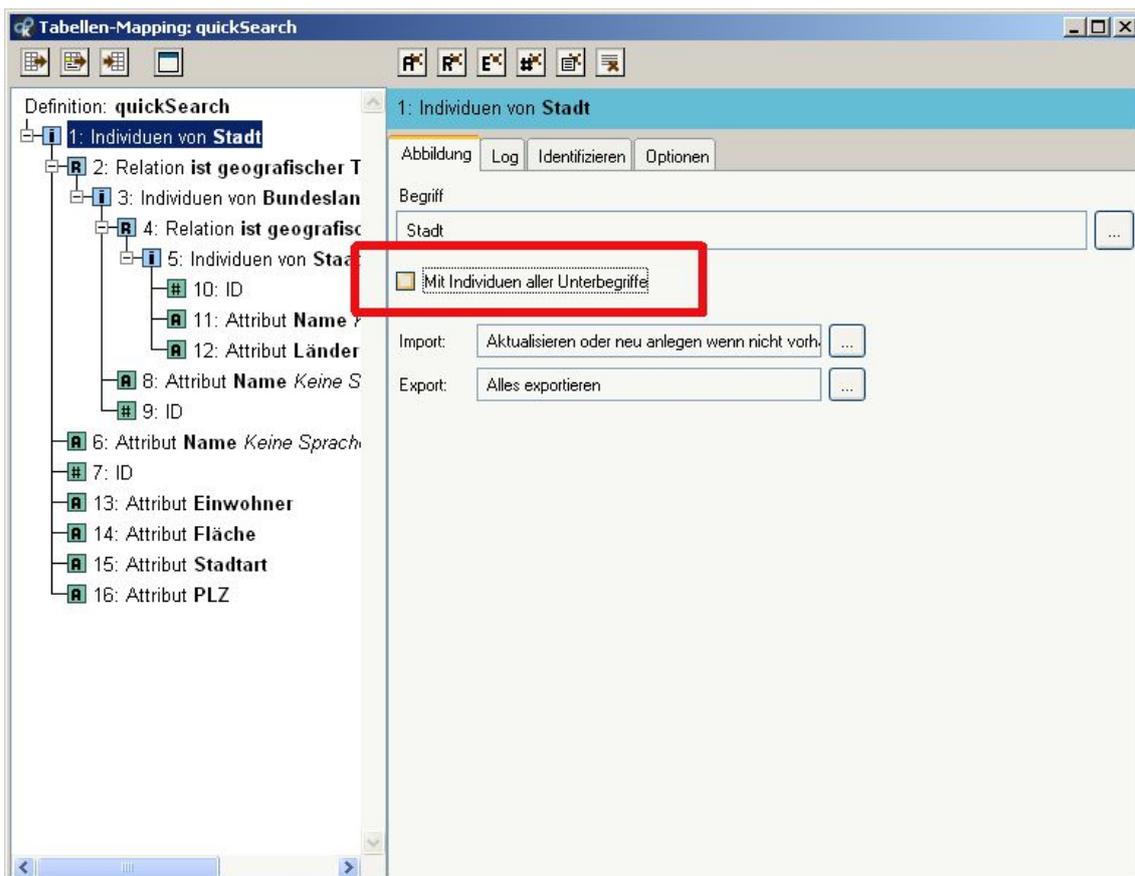
Problem:

Schnellsuche findet keine Ergebnisse.

Mögliche Ursachen:



- Für bestimmte Parameter der Quick-Search-Expertensuche ist kein Standardwert eingestellt (z.B. "*" um nach beliebigen Werten zu suchen). Wenn in einem Formularfeld der Quick-Search im Web-Frontend kein Wert angezeigt wird, so wird auch tatsächlich nach Attributen gesucht, die einen Leerstring als Wert enthalten.
- Im Quick-Search-Export-Mapping ist bei einer Individuen-Abbildung das Häkchen "Mit Individuen aller Unterbegriffe" nicht gesetzt. Dadurch können die von der Expertensuche gefundenen Individuen nicht exportiert werden.



11.2 Topic-Detailseite

11.2.1 Es werden keine Super-/Subkonzepte und/oder Individuen angezeigt

Möglicherweise fehlen in der ViewConfig für das betreffende Konzept oder Individuum die Elemente zur Anzeige von Super-/Subkonzepten und Individuen. Fügen Sie das folgende XML-Fragment unterhalb des <ViewConfig>-Elements ein, um das Problem zu lösen:

```
<Header>  
  <Superconcepts/>  
  <Subconcepts/>  
  <Instances/>  
</Header>
```



Das Header-Element muss das erste Element innerhalb des ViewConfig-Elements sein, d.h. es dürfen keine anderen Elemente wie Attribute, Relation, AllAttributes, etc. davor stehen.

Nach einem Neustart des Knowledge-Portals oder einem VolumeConfig-Reload sollten die Super-/Subkonzepte und Instanzen dann angezeigt werden.

11.2.2 Es werden keine Attribute angezeigt

Wenn auf der Pflege- oder auf der Detailseite keine Attribute angezeigt werden, diese aber definitiv vorhanden sind (d.h. die Attribute und Relationen im Knowledge-Builder angezeigt werden), so hat dies in der Regel eine der folgenden Ursachen:

- Es ist eine leere ViewConfig (für Detailseiten) bzw. eine leere EditConfig (für Pflegeseiten) für das betreffende Objekt bzw. den betreffenden Objekttyp (z.B. Person, Projekt, etc.) konfiguriert. Bitte überprüfen Sie in diesem Fall die EditConfig bzw. ViewConfig in der KP-Konfiguration im Wissensnetz.
- Die Eigenschaften des betreffenden Objekts werden versteckt. Die Konfiguration der versteckten Eigenschaften wird im Kapitel "Ausblenden bestimmter Attribute und Relationen" beschrieben (siehe Verweis in der rechten Spalte).

12 Referenz

12.1 XML-Suchkonfiguration (searchConfig.dtd)

12.1.1 Element "SearchConfiguration"

Beschreibung	Das Element SearchConfiguration definiert eine Suchkonfiguration.
Mögliche Unterelemente	<ul style="list-style-type: none">• SearchDescriptions

Die folgenden Attribute können an diesem Elementen gesetzt werden:

Attributname	Beschreibung	Attributtyp
id	ID der Suche, für welche diese Suchkonfiguration gelten soll. Im Knowledge-Portal ist als Wert für dieses Attribut derzeit nur "defaultSearch" (Standardsuche über die Box "Suche") möglich.	Zeichenkette, Pflichtfeld



12.1.2 Element "SearchDescriptions"

Beschreibung	Das Element SearchDescriptions definiert die Menge von Teilsuchen, die ausgeführt werden sollen, wenn eine Suchanfrage mit dieser Suchkonfiguration gestartet wird. Die Teilsuchen müssen mittels Search-Elementen definiert werden, die in dieses SearchDescriptions-Element geschachtelt werden.
Mögliches Unterelement von	<ul style="list-style-type: none">• SearchConfiguration
Mögliche Unterelemente	<ul style="list-style-type: none">• Search

12.1.3 Element "Search"

Beschreibung	Das Element Search definiert eine Teilsuche innerhalb einer Suchkonfiguration. Die Parameter zur Ausführung dieser Teilsuche werden im XML als Attribute und Unterelemente des jeweiligen Search-Elements angegeben. Wird eine Suchanfrage gestellt, so wird für jede Teilsuche das Suchergebnis ausgerechnet und auf der Suchergebnisseite angezeigt.
Mögliches Unterelement von	<ul style="list-style-type: none">• SearchConfiguration
Mögliche Unterelemente	<ul style="list-style-type: none">• QueryParams• SortOption

Die folgenden Attribute können an einem Search-Element gesetzt werden:

Attributname	Beschreibung	Attributtyp
id	ID der jeweiligen Teilsuche. Die ID muss innerhalb der jeweiligen Suchkonfiguration eindeutig sein.	Zeichenkette, Pflichtfeld



componentId	ID des Suchadapters, das zur Ausführung dieser Teilsuche verwendet werden soll. Standardmäßig ist im Knowledge-Portal nur der Suchadapter für K-Infinity- bzw. NextBot-Suchen unter der ID "kinfinity" konfiguriert.	Zeichenkette, Pflichtfeld
loadLazy	<p>Wird als Attributwert "true" angegeben, so wird das Ergebnis für diese Teilsuche in einem separaten Thread (d.h. im Hintergrund) berechnet. Dies hat zur Folge, dass die Suchergebnisseite für die übergeordnete Suche angezeigt werden kann, auch wenn das Suchergebnis dieser Teilsuche noch nicht vorliegt. In diesem Fall wird in dem Bereich auf der Suchergebnisseite, in welchem die Ergebnisse dieser Teilsuche erscheinen sollten, ein Hinweis angezeigt, dass das Suchergebnis noch berechnet wird. Sobald die Ergebnisse vorliegen, wird der Hinweis durch die Auflistung der Ergebnisse ersetzt. Diese Einstellung bietet sich für Teilsuchen an, die bekanntermaßen längere Zeit zur Beantwortung von Suchanfragen benötigen (z.B. NextBot-Suchen).</p> <p>Wird als Attributwert "false" angegeben, so wird die Suchergebnisseite erst angezeigt, wenn das Suchergebnis für diese Teilsuche vorliegt.</p>	Boolesch, optional, Standardwert: false
optional	<p>Gibt an, ob diese Teilsuche bei der Beantwortung einer Suchanfrage ignoriert werden soll (Wert "true"), wenn der angegebene Suchadapter (Attribut "componentId") diese Teilsuchkonfiguration als ungültig betrachtet. Beispiel: In einer Suchkonfiguration wird eine Teilsuche mit dem K-Infinity-Suchadapter definiert, die auf eine K-Infinity-Suche mit dem Namen "xxx" abgebildet ist (siehe Beschreibung des KInfinity-Suchadapters bzw. des Elements "QueryParam"). Wird für diese Teilsuchkonfiguration das Attribut "optional" auf true gesetzt, so prüft der K-Infinity-Suchadapter bereits beim Laden der Suchkonfiguration, ob eine Suche namens "xxx" auch im Wissensnetz definiert ist. Ist dies nicht der Fall, wird die Teilsuche bei Suchanfragen komplett ignoriert (d.h. es erscheint auf der Suchergebnisseite auch kein Bereich zur Darstellung der Teilsuchergebnisse). Wird das Attribut "optional" auf false gesetzt, so prüft der Suchadapter nicht, ob die Suche im Wissensnetz auch wirklich definiert ist. Auf der Suchergebnisseite erscheint in diesem Fall immer ein Bereich zur Anzeige der Teilsuchergebnisse und darin - falls die Suche im Wissensnetz nicht definiert ist - eine Fehlermeldung.</p>	Boolesch, optional, Standardwert: false



12.1.4 Element "QueryParams"

Beschreibung	Das Element "QueryParams" definiert Suchparameter, die in der Suchanfrage für eine Teilsuche enthalten sein sollen. Vor jeder Ausführung einer Teilsuche wird die Liste der Unterelemente des QueryParams-Elements durchlaufen und eine Menge von Suchparametern berechnet. Aus der Menge der Suchparameter wird dann eine Suchanfrage gebildet, die dann von dem für die Teilsuche angegebenen Suchadapter beantwortet wird.
Unterelement von	<ul style="list-style-type: none">• Search
Mögliche Unterelemente	<ul style="list-style-type: none">• ComputedQueryParam• UserQueryParams• QueryParam

12.1.5 Element "ComputedQueryParam"

Beschreibung	Mit dem Element ComputedQueryParam kann eine Java-Klasse definiert werden, die zur dynamischen Berechnung von Suchparametern verwendet werden soll (ein sog. Query-Parameter-Computer). Die Java-Klasse muss dazu das Interface "com.iviews.searchmanager.model.QueryParamComputer" aus dem Search-Manager-Framework und damit die Methode "computeQueryParams" implementieren. Im Programmcode haben Entwickler die Möglichkeit, auf die Suchanfrage des Benutzers, die Suchkonfiguration und die Suchergebnisse von anderen Teilsuchen zuzugreifen. Damit ergeben sich vielfältige Möglichkeiten zur dynamischen Erzeugung und Ergänzung von Suchanfragen. Um von der Suchkonfiguration aus Einfluss auf die Berechnung der Suchparameter nehmen zu können, ist es möglich, Property-Elemente in dieses Element zu schachteln (siehe Beispiel). Im Knowledge-Portal sind einige Query-Parameter-Computer standardmäßig enthalten. Diese werden im Kapitel "Query-Parameter-Computer" beschrieben.
Mögliches Unterelement von	<ul style="list-style-type: none">• Search
Mögliche Unterelemente	<ul style="list-style-type: none">• Property



Die folgenden Attribute können an diesem Elementen gesetzt werden:

Attributname	Beschreibung	Attributtyp
computerClass	Vollqualifizierender Name der Java-Klasse, die das Interface "com.iviews.searchmanager.model.QueryParamComputer" implementieren muss.	Zeichenkette, Pflichtfeld

Beispiel:

```
<ComputedQueryParam  
  computerClass=  
    "com.iviews.knowledgeportal.newsearch.KTopicHitsQueryParamComputer">  
  <Property name="topicsSearchDescription" value="semanticSearch" />  
  <Property name="name" value="filterTopics" />  
</ComputedQueryParam>
```

Im Beispiel wird ein Query-Param-Computer für eine Teilsuche definiert und zwei Eigenschaften "topicsSearchDescription" und "name" für diesen gesetzt.

12.1.6 Element "QueryParam"

Beschreibung	Das Element "QueryParam" kann dazu verwendet werden, einen bestimmten Suchparameter für die Suchanfrage einer Teilsuche zu definieren. Hierzu muss der Name und der Wert des Suchparameters als Attribut angegeben werden.
Mögliches Unterelement von	<ul style="list-style-type: none">• QueryParams
Mögliche Unterelemente	

Die folgenden Attribute können an diesem Elementen gesetzt werden:

Attributname	Beschreibung	Attributtyp
--------------	--------------	-------------



name	Name des Suchparameters, der zur Suchanfrage der Teilsuche hinzugefügt werden soll. Die möglichen Werte für dieses Attribut hängen davon ab, welcher Suchadapter für die jeweilige Teilsuche konfiguriert ist (die Suchadapter unterstützen jeweils unterschiedliche Parameter). Details hierüber sind in der Referenz für Suchadapter zu finden.	Zeichenkette, Pflichtfeld
value	Wert des Suchparameters. Die möglichen Werte sind abhängig vom Suchparameter und sind in der Referenz der Suchadapter beschrieben.	Zeichenkette, Pflichtfeld

Beispiel:

```
<QueryParam name="searchName" value="trigramSearch" />
```

Im Beispiel wird der Parameter "searchName" mit dem Wert "trigramSearch" für die Suchanfrage der Teilsuche definiert.

12.1.7 Element "SortOption"

Beschreibung	Das Element SortOption definiert ein mögliches Sortierkriterium für die Treffer einer Teilsuche. Dazu wird ein Name (Attribut id) auf eine vom Suchadapter vorgegebene Beschreibung des jeweiligen Sortierkriteriums (Attribut sortKey) abgebildet. Das Sortierkriterium kann dann in anderen Teilen der Knowledge-Portal-Konfiguration (z.B. in der Konfiguration der Suchergebnisseiten) über die ID angegeben werden. Teilweise unterstützen die Suchadapter standardmäßig Sortierkriterien mit bestimmten Namen, so dass diese nicht mehr extra definiert werden müssen (siehe Referenz der Suchadapter).
Mögliches Unterelement von	<ul style="list-style-type: none">• Search

Die folgenden Attribute können an diesem Elementen gesetzt werden:

Attributname	Beschreibung	Attributtyp
id	Name des Sortierkriteriums zur Verwendung in der Knowledge-Portal-Konfiguration.	Zeichenkette, Pflichtfeld



sortKey	Beschreibung des Sortierkriteriums. Die Syntax der Beschreibung ist in der Referenz der Suchadapter beschrieben.	Zeichenkette, Pflichtfeld
---------	--	---------------------------

Beispiel:

```
<SortOption id="postalCode" sortKey="@postalCode$" />
```

Im Beispiel wird das Sortierkriterium "postalCode" auf einen für den K-Infinity-Suchadapter spezifischen Ausdruck abgebildet, nach dem der Suchadapter die Treffer dann sortieren kann.

12.1.8 Element "UserQueryParams"

Beschreibung	Das Element "UserQueryParams" sorgt dafür, dass alle Suchparameter, die in der Suchanfrage des Benutzers enthalten sind, auch in die Suchanfrage für diese Teilsuche einfließen. Normalerweise werden die Suchparameter aus der Benutzeranfrage nicht automatisch in die Suchanfrage der Teilsuche aufgenommen. Es werden nur solche Suchparameter aus der Suchanfrage des Benutzers übernommen, die auch von dem für diese Teilsuche verwendeten Suchadapter unterstützt werden.
Mögliches Unterelement von	<ul style="list-style-type: none">• QueryParams

Beispiel:

```
<UserQueryParams/>
```

12.1.9 Element "Property"



Beschreibung	Mit dem Element "Property" können Eigenschaften für ComputedQueryParams (siehe übergeordnetes Element) in der Suchkonfiguration definiert werden. Vor der Ausführung einer Teilsuche wird für jeden in der Konfiguration angegebenen ComputedQueryParam eine Instanz der angegebenen Java-Klasse erzeugt und die in den Property-Elementen definierten Name-Wert-Paare als Bean-Properties an dieser Instanz gesetzt.
Mögliches Unterelement von	<ul style="list-style-type: none">• ComputedQueryParam

Die folgenden Attribute können an diesem Elementen gesetzt werden:

Attributname	Beschreibung	Attributtyp
name	Name einer Bean-Property, welche für Instanzen eines Query-Parameter-Computers gesetzt werden kann.	Zeichenkette, Pflichtfeld
value	Wert der Bean-Property. Sofern die Bean-Property nicht vom Typ Zeichenkette ist, konvertiert das Knowledge-Portal den Zeichenkettenwert automatisch in den passenden Typ (z.B. Ganzzahl).	Zeichenkette, Pflichtfeld

Beispiel:

```
<Property name="name" value="filterTopics" />
```

Im Beispiel wird die Eigenschaft "name" eines Query-Parameter-Computers auf den Wert "filterTopics" gesetzt.

12.2 XML Suchergebnisseiten-Konfiguration

12.2.1 Element "Search"

Beschreibung	Das Element "Search" definiert die Konfiguration einer Suchergebnisseite.
Mögliche Unterelemen	<ul style="list-style-type: none">• ResultViewConfig



Das Element unterstützt keine Attribute.

12.2.2 Element ResultViewConfig

Beschreibung	Das Element "ResultViewConfig" definiert das Layout einer Suchergebnisseite für eine spezielle Suchmanager-Konfiguration.
Mögliche Unterelemente	<ul style="list-style-type: none">• TabGroup• TwoColumnContainer• Actions• RenderHint

Die folgenden Attribute können an diesem Elementen gesetzt werden:

Attributname	Beschreibung	Attributtyp
id	ID der Suchkonfiguration, für die die Layoutkonfiguration gelten soll.	Zeichenkette, Pflichtfeld

12.2.3 Element TabGroup

Beschreibung	Das Element "TabGroup" ist das gruppierende Element für die einzelnen Tabs einer Suchergebnisseite.
Mögliche Unterelemente	<ul style="list-style-type: none">• Tab

Die folgenden Attribute können an diesem Elementen gesetzt werden:



Attributname	Beschreibung	Attributtyp
renderMode	Definiert den RenderMode, mit der die TabGroup gerendert wird. Dieses Attribut ist nur für Anpassungsprojekte interessant.	Zeichenkette, optionales Feld
styleClass	Die CSS Klasse für das TabGroup Element auf der erzeugten HTML Seite.	Zeichenkette, optionales Feld
styleId	Die CSS ID für das TabGroup Element auf der erzeugten HTML Seite.	Zeichenkette, optionales Feld

12.2.4 Element Tab

Beschreibung	Das Element Tab definiert einen einzelnen Reiter innerhalb des Element TabGroup. Ein Reiter enthält eine eigenständige Layoutkonfiguration.
Mögliche Unterelemente	<ul style="list-style-type: none">• TwoColumnContainer

Die folgenden Attribute können an diesem Elementen gesetzt werden:

Attributname	Beschreibung	Attributtyp
renderMode		Zeichenkette, optional
styleClass		Zeichenkette, optional
styleId		Zeichenkette, optional

12.2.5 Element TwoColumnContainer



Beschreibung	Das Element "TwoColumnContainer" ist das gruppierende Element für die Spalten des Suchergebnislayouts.
Mögliche Unterelemente	<ul style="list-style-type: none">• TwoColumnRow• ColumnRow

Die folgenden Attribute können an diesem Elementen gesetzt werden:

Attributname	Beschreibung	Attributtyp
renderMode		Zeichenkette, optional
styleClass		Zeichenkette, optional
styleId		Zeichenkette, optional

12.2.6 Element ColumnRow

Beschreibung	Das Element "ColumnRow" definiert eine neue Zeile innerhalb des Suchlayouts. Die eingefügte Zeile ist einspaltig, d.h. die enthaltene Ergebnisliste nimmt die volle Breite der Seite bzw. des Tabs ein.
Mögliche Unterelemente	<ul style="list-style-type: none">• ResultHeader• SearchPartResult

Das Element unterstützt keine eigenen Attribute.

12.2.7 Element TwoColumnRow



Beschreibung	<p>Das Element "TwoColumnRow" definiert eine neue Zeile innerhalb des Suchlayouts. Die neu eingefügte Zeile ist zweispaltig, d.h. links und rechts können separate Suchergebnislisten stehen.</p> <p>Wird nur eine Suchergebnisliste als Unterelement konfiguriert, so nimmt dieses die volle Breite ein, d.h. "TwoColumnRow" verhält sich dann wie eine normale "ColumnRow".</p>
Mögliche Unterelemente	<ul style="list-style-type: none">• LeftColumn• RightColumn

Es werden keine eigenen Attribute am Element unterstützt.

12.2.8 Element LeftColumn

Beschreibung	Das Element "LeftColumn" definiert das Layout der linken Spalte einer zweispaltigen Zeile (TwoColumnRow).
Mögliche Unterelemente	<ul style="list-style-type: none">• SearchPartResult

Das Element unterstützt keine eigenen Attribute.

12.2.9 Element RightColumn

Beschreibung	Das Element "RightColumn" definiert die rechte Spalte einer zweispaltigen Zeile (TwoColumnRow).
Mögliche Unterelemente	<ul style="list-style-type: none">• SearchPartResult



Das Element unterstützt keine Attribute.

12.2.10 Element SearchPartResult

Beschreibung	Das Element "SearchPartResult" zeigt das Ergebnis einer einzelnen Suche. Die Darstellung des Suchergebnisses kann mit diesem Element konfiguriert werden.
Mögliche Unterelemente	<ul style="list-style-type: none">• Actions• GroupActions• HitActions• Label• LabelKey• RenderHint• SortOption

Die folgenden Attribute können an diesem Elementen gesetzt werden:

Attributname	Beschreibung	Attributtyp
hitsPageSize	Gibt an, wieviele Treffer der Suche pro Seite dargestellt werden.	Ganzzahl, optional, Standardwert: 20
groupsPageSize	Wenn die Treffer der Suche gruppiert angezeigt werden, dann kann mit diesem Wert konfiguriert werden, wieviele Treffer pro Gruppe angezeigt werden. Ist die Trefferanzahl innerhalb einer Gruppe größer als der hiermit konfigurierte Wert, so werden die Treffer der Gruppe seitenweise angezeigt.	Ganzzahl, optional, Standardwert: 5
hitsPerGroupSize		Ganzzahl, optional, Standardwert: -1



initialGrouping	Gibt an, ob die Suchergebnisse der Suche initial gruppiert oder ungruppiert dargestellt werden sollen.	Boolean, optional Standardwert: true
-----------------	--	--------------------------------------

12.2.11 Element ResultHeader

Beschreibung	Das Element "ResultHeader" definiert den Kopfbereich der Darstellung eines einzelnen Suchergebnisses.
Mögliche Unterelemente	<ul style="list-style-type: none">• Actions

Die folgenden Attribute können an diesem Elementen gesetzt werden:

Attributname	Beschreibung	Attributtyp
id	ID der Suche, für welche diese Suchkonfiguration gelten soll. Im Knowledge-Portal ist als Wert für dieses Attribut derzeit nur "defaultSearch" (Standardsuche über die Box "Suche") möglich.	Zeichenkette, Pflichtfeld

12.2.12 Elemente "Actions" / "GroupActions" / "HitActions"

Beschreibung	Die Elemente Actions, GroupActions und HitActions haben die gleichen möglichen Einstellungen und unterscheiden sich nur im Namen des Elements.
Mögliche Unterelemente	<ul style="list-style-type: none">• CommentSearchResultAction• PrintSearchResultAction



Ein Actions-Element hat keine eigenen Attribute.

Die möglichen Unterelemente sind:

CommentSearchResultAction

Mit dieser Action kann ein Benutzer einen Kommentar zur aktuellen Suche abgeben. Werden beispielsweise keine Ergebnisse gefunden, so kann er dies anmerken. Die Anmerkungen des Benutzers werden als Individuen eines konfigurierten Konzepts im Netz abgelegt.

Konfigurationsattribute:

- *conceptId*: Der interne Name des Konzepts, von dem ein neues Individuum angelegt werden soll.
- *queryAttId*: Der interne Name des Attributs an einem Individuum des konfigurierten Konzepts, in dem der Suchbegriff abgespeichert werden soll.

PrintSearchResultAction

Mit dieser Action wird dem Benutzer die Möglichkeit gegeben, ein Suchergebnis auszudrucken.

12.2.13 Element "SortOption"

Beschreibung	Das Element "SortOption" dient dazu, die Knöpfe zum Nachsortieren eines Suchergebnisses zu konfigurieren.
Mögliche Unterelemente	<ul style="list-style-type: none">• SortCriteria

Die folgenden Attribute können an diesem Elementen gesetzt werden:

Attributname	Beschreibung	Attributtyp
id	Eindeutige ID der Sortieroption. Die ID muss pro Element <SearchPartResult> eindeutige	Zeichenkette, Pflichtfeld

12.2.14 Element "SortCriteria"



Beschreibung	Spezifiziert eine Möglichkeit, das Suchergebnis zu sortieren.
Mögliche Unterelemente	

Die folgenden Attribute können an diesem Elementen gesetzt werden:

Attributname	Beschreibung	Attributtyp
id	Beschreibt ein Sortierkriterium. Der Wert muss mit der ID einer SortOption in der Suchkonfiguration (siehe Search-Manager) übereinstimmen.	Zeichenkette, Pflichtfeld
orderDescending	Gibt an, ob aufsteigend oder absteigend sortiert werden soll. Wird dieses Attribut nicht angegeben, dann wird aufsteigend sortiert.	Boolean, optional, Standardwert: false

12.3 Suchadapter

12.3.1 K-Infinity-Suchadapter (kinfinity)

Unterstützte Suchparameter

Der K-Infinity-Suchadapter bildet die Suchen des Knowledge-Portals auf die in einem K-Infinity-Wissensnetz definierten Suchkonfigurationen ab. Der Suchadapter ist im Knowledge-Portal unter der ID "kinfinity" registriert. Es werden die die folgenden Suchparameter unterstützt:

Name	Beschreibung	Datentyp / Ab Version
-------------	---------------------	------------------------------



containerFilter	Parameter zur Einschränkung von Nextbot Repositories. Dieser Parameter wird erst ab Bridges der Version 3.2 mit Build-Datum > 27.10.2010 ausgewertet. Wird dieser Parameter nicht gesetzt, so werden grundsätzlich alle indextierten Repositories und darin enthaltenen Verzeichnisse durchsucht. Sollen nur bestimmte Repositories/Verzeichnisse durchsucht werden, so kann mit diesem Parameter eine Liste von NextBot-internen Locators (URI) angegeben werden (Liste von Strings). Durch die Einschränkung auf ein Verzeichnis bzw. Repository werden trotzdem noch alle Dateien in Unterverzeichnissen desselben durchsucht.	Liste von Zeichenketten (Nextbot-Locator URI)
filterTopics	Mit diesem Parameter kann eine Menge von Topics spezifiziert werden, die von der Treffermenge abgezogen werden sollen, bevor das Teilsuchergebnis zurückgeliefert wird.	Liste von Zeichenketten (DMIDs); 3.4.x
fromDateFilter	Wird dieser Parameter in einer Suchanfrage gesetzt, so wird bei der Ausführung von NextBot-Suchen der Parameter "fileDateMinFilter" auf den entsprechenden Datumswert gesetzt. Dies bewirkt, dass nur Dokumententreffer zurückgeliefert werden, bei denen das Metadatum "lastmodified" >= dem angegebenen Datum ist.	Datum; 3.4.x
groupHits	Wird dieser Parameter gesetzt, so werden die Treffer, die Topics im Wissensnetz repräsentieren, nach ihrer Layout-Kategorie gruppiert.	Boolesch (false/true); 3.4.x
groupKPath	KPath-Ausdruck der bei aktivierter Gruppierung (siehe oben) ausgewertet wird, um das Gruppierungskriterium zu ermitteln. Standardwert: topic()/conceptOrSelf()/attributeInherited('layoutCategory')/valueString()	Zeichenkette (KPath-Ausdruck)
groupOrderAscending	Gibt an, ob die Gruppen nach dem Wert des Sortierkriteriums (siehe unten) aufsteigend (true) oder absteigend (false) sortiert werden sollen.	Boolean; Standardwert: true



groupOrderKPath	KPath-Ausdruck, der bei aktivierter Treffergruppierung auf das jeweilige Gruppierungskriterium (z.B. ein Wissensnetzobjekt) angewendet wird, um das Sortierkriterium für eine Gruppe zu bestimmen. Wird diesem Parameter kein KPath-Ausdruck zugeordnet (Standardwert), so werden die Gruppen direkt nach dem Gruppierungskriterium (z.B. Layout-Kategorie) sortiert.	Zeichenkette (KPath-Ausdruck); Standardwerte: nicht definiert
initialTopics	Mit diesem Parameter kann eine Menge von Topics spezifiziert werden, die als Wert für den gleichnamigen Parameter einer K-Infinity-Suchen eingegeben werden soll. So können K-Infinity-Suchen wie z.B. die semantische Suche aufgerufen werden, die als Eingabe eine Menge von Topics akzeptieren (z.B. die semantische Suche).	Liste von Zeichenketten (DMIDs); 3.4.x
kemInitialTopicList	Dieser Parameter hat die gleiche Funktion wie der Parameter "initialTopics", allerdings muss die Topic-Liste in Form einer KEMListEditors angegeben werden. Dieser Parameter wird vom Suchadapter gegenüber dem Parameter "initialTopics" bevorzugt.	KEMListEditor; 3.4.x
kemSession	KEMSession, in der die K-Infinity-Suche ausgeführt werden soll. Falls der Parameter nicht definiert ist, wird vom Suchadapter eine neue KEMReadSession zur Ausführung der Suche geöffnet.	KEMSession; 3.4.x



queryExpression	<p>Dieser Parameter wird auf den gleichnamigen Parameter einer Suche vom Typ "Nextbot Dokumentensuche über Suchausdruck" abgebildet. Der Parameter dient dazu, eine Suchanfrage mit Gruppen von gewichteten Termen und Phrasen zu formulieren (dies kann mit einem einfachen Suchstring nicht formuliert werden). Als Wert für diesen Parameter muss ein Datenstruktur in Form von ineinander geschachtelten Java-List-Objekten angegeben werden. Der Aufbau dieser Datenstruktur wird in der folgenden Spezifikation unter der Rubrik "Query Language" beschrieben:</p>  <p>Query-Syntax.txt</p> <p>Dieser Parameter wird ab der KP-Version 3.9.16 durch den Parameter "useNextbotQuerySyntax" (Wert "list") ersetzt!</p>	List-Objekt; 3.4.x - 3.9.15
searchExpression	Zerlegte Suchanfrage in Form eines Objektes der Klasse "AbstractExpression" (Search-Manager-Bibliothek).	Objekt; 3.9.16
searchName	Name der K-Infinity-/NextBot-Suche, die ausgeführt werden soll. Wird auf den gleichnamigen K-Infinity-Suchparameter abgebildet.	Zeichenkette; 3.4.x
searchString	Suchanfrage in Form einer Zeichenkette. Dieser Parameter wird direkt auf den gleichnamigen Parameter der K-Infinity-Suche abgebildet, wenn der Parameter "useExpressionQuerySyntax" mit dem Wert "true" in der Suchanfrage enthalten ist. Ansonsten wird der Wert des Parameters als Suchausdruck in der K-Infinity-Suchsyntax ausgewertet und in ausgewerteter Form als Wert des Parameters "searchExpression" der Suchanfrage hinzugefügt.	Zeichenkette; 3.4.x



searchTree	<p>Zerlegte Suchanfrage in Form eines Objekts der Klasse com.iviews. searchmanager.model.SearchTree. Dieser Parameter wird mit der Knowledge-Portal-Version 3.9.16 durch den Parameter "searchExpression" ersetzt.</p>	SearchTree-Objekt; 3.4.x
sortOrder	<p>Sortierreihenfolge der Treffer. Die Sortierreihenfolge kann in Form einer Liste von SortOption-Objekten angegeben werden. Mehrere SortOption-Objekte führen zu einer mehrstufigen Sortierung. Jedes SortOption-Objekt enthält die ID des Sortierkriteriums sowie die Sortierreihenfolge. Die möglichen Werte für IDs von Sortierkriterien ergeben sich aus der Tabelle unten und aus den in der Suchkonfiguration definierten Sortierkriterien (siehe Element SortOption in der Referenz zur XML-Suchkonfiguration). Alternativ kann diese Liste auch als Zeichenkette angegeben werden und zwar in der folgenden Form (ohne Zeilenumbruch):</p> <pre>[isOrderDescending1]:[sortCriteriald1], [isOrderDescending2]:[sortCriteriald2], ... [isOrderDescendingN]:[sortCriterialdN]</pre> <p>isOrderDescendingX muss hierbei durch "false" (aufsteigend) oder "true" (absteigend) ersetzt werden, sortCriterialdX muss durch die ID des Sortierkriteriums ersetzt werden.</p>	Liste von SortOption-Objekten oder Zeichenkette; 3.5.0
toDateFilter	<p>Wird dieser Parameter in einer Suchanfrage gesetzt, so wird bei der Ausführung von NextBot-Suchen der Parameter "fileDateMaxFilter" auf den entsprechenden Datumswert gesetzt. Dies bewirkt, dass nur Dokumententreffer zurückgeliefert werden, bei denen das Metadatum "lastmodified" <= dem angegebenen Datum ist (siehe auch fromDateFilter).</p>	Datum; 3.4.x



useExpressionQuerySyntax	Gibt an, dass der searchString-Parameter in der Suchanfrage der Syntax des K-Infinity-Suchtyps "Volltextsuche über Suchausdruck" entspricht. Ist der Wert dieses Parameters true, so wird der Parameter "useNextbotQuerySyntax" nicht mehr berücksichtigt.	Boolesch; 3.5.0
useNextbotQuerySyntax	Dieser Parameter gibt an, dass die Suchanfrage des Benutzers in die NextBot-Suchsyntax übersetzt werden soll. Dieser Parameter muss für normale Nextbot-Volltextsuchen auf den Wert "true" gesetzt werden, und für normale K-Infinity-Suchen, die einen Suchstring erwarten, auf den Wert "false" (Standardwert). Ab der Knowledge-Portal-Version 3.9.16 kann das Knowledge-Portal die Suchanfrage des Benutzers auch in die erweiterte Suchsyntax der "NextBot-Volltextsuche über Suchausdruck" übersetzen. Hierfür muss dieser Parameter auf den Wert "list" gesetzt werden.	Von Version 3.4.0 - 3.9.15 vom Typ Boolean (false/true), in Versionen > 3.9.15 ein Enum mit den möglichen Werten false, true und list. Standardwert: false
userId	Ist dieser Parameter in der Suchanfrage gesetzt, so wird die Suchanfrage im Kontext des angegebenen Benutzers ausgeführt. Dieser Parameter wird nicht berücksichtigt, wenn der Parameter "kemSession" in der Suchanfrage enthalten ist (was im Knowledge-Portal aus Sicherheitsgründen immer der Fall ist).	Zeichenkette; 3.4.x
variables	Dictionary, welches an K-Infinity-Suchen vom Typ "Zusammenstellbare Suchen" (aka Floyd-Suchen) übergeben werden kann. Jedes Key-Value-Paar im Dictionary kann als Variable in der Suche ausgewertet werden (Key -> Variablenname, Value -> Wert).	Dictionary / Map; 3.7.0

Sortierkriterien

Der Suchadapter interpretiert bei der Angabe von Sortierkriterien (SortOption-Element in der Suchkonfiguration) die als sortKey angegebenen Ausdrücke als KPath-Ausdrücke, d.h. diese müssen der KPath-Syntax entsprechen. Details zu KPath sind im KScript-Handbuch nachzulesen.

Der Suchadapter unterstützt standardmäßig die folgenden Sortierkriterien:

Name	Beschreibung
------	--------------



application.title = K-Infinity Knowledge-Portal

attribute.boolean.value.false = Nein
attribute.boolean.value.true = Ja
attribute.notDisplayable = Attributwert nicht darstellbar
attribute.notEditable = Attributwert nicht editierbar

beanrenderers.error = Anzeigefehler

box.eventLogQueries.title = Benutzeraktivitäten
box.expertSearch.title = Werkzeuge
box.hide = Ordner schließen
box.hierarchies.title = Navigation
box.history.title = Verlauf
box.login.title = Benutzer
box.matchings.title = Ähnliche Themen
box.quickSearch.title = Schnellsuche
box.search.title = Suche
box.show = Ordner öffnen

button.cancel = Abbrechen
button.ok = OK

collection.collapseLink = [Weniger anzeigen]
collection.empty = Keine Einträge.
collection.expandLink = [{0} mehr anzeigen]

comments.commentBy = Kommentar von {0}
comments.confirmDelete = Wollen Sie den Kommentar wirklich löschen?
comments.delete = Kommentar löschen
comments.hideComments = Kommentare ausblenden
comments.noComment = Es liegen noch keine Kommentare zu diesem Thema vor.
comments.numComments = Es gibt {0} Kommentare zu diesem Thema.
comments.showComments = Anzeigen
comments.writeComment = Kommentar abgeben

common.object = Objekt
common.objects = Objekte

component.search.default.name = Standardsuche
component.search.websearch.name = Websuche
component.search.empty = Keine Suchergebnisse gefunden

confirmCommit.text = Alle Änderungen wurden erfolgreich gespeichert.
confirmCommit.title = Änderungen gespeichert
confirmCommit.continue = Weiter

context.authors = Autoren
context.organisations = Organisationen

contextSearch.empty = Keine Treffer gefunden!
contextSearch.showAll = Alles zeigen
contextSearch.showLimited = Nur {0} Einträge zeigen



```
dateFilter.error.from          = Das 'von'-Datum konnte nicht zerlegt
    werden: '{0}'.
dateFilter.error.to           = Das 'bis'-Datum konnte nicht zerlegt
    werden: '{0}'.
dateFilter.error.toBeforeFrom = Das 'von'-Datum liegt vor dem 'bis'-Datum.
dateFilter.from               = Von:
    # Man beachte, dass der DateFilter nur auf Tag-Genauigkeit ausgelegt ist, nicht
    # auch Minuten oder Stunden,
    # auch, wenn das Pattern hier verändert wird.
dateFilter.inputPattern       = dd.MM.yyyy
dateFilter.inputPattern.explain = (Datumsformat TT.MM.JJJJ)
dateFilter.title              = Dokumente mit Änderungsdatum\
dateFilter.to                 = Bis:

demote.confirm                = Möchten Sie das ausgewählte Dokument wirklich herabstufen?
demote.select.title           = Demote - Relation wählen

document.unnamed              = Ohne Name

edit.apply                    = Übernehmen
edit.back                     = Zurück
edit.cancel                   = Abbrechen
edit.cannotCreateAttribute    = Kein Recht zum Erzeugen dieses Attributes.
##edit.chooseTarget           = Bitte wählen Sie das Ziel der Verknüpfung
edit.confirmDelete            = Möchten Sie dieses Objekt wirklich löschen?
edit.confirmDelete.attribute  = Möchten Sie dieses Attribut wirklich löschen?
edit.confirmDelete.relation   = Möchten Sie diese Relation wirklich löschen?
edit.confirmDeleteExt         = Möchten Sie diese Erweiterung wirklich löschen?
edit.content                  = Inhalt:
edit.create                   = Erzeugen
edit.createTarget             = Ziel erzeugen
edit.delete                   = Löschen
edit.edit                     = Bearbeiten
edit.emptyTargetTopic         = Bitte geben Sie einen Suchbegriff ein
edit.existing.ext             = Erweiterung: {0}
edit.existing.title           = Bearbeiten - {0}
edit.extensions               = Mögliche Erweiterungen
edit.filterTargets            = Ziele filtern
edit.findOrCreateTargets      = Objekte verknüpfen oder Verknüpfungsziel erzeugen
edit.findTargets              = Relationsziel(e) suchen...
edit.findTargetsFor           = Relationsziel für "{1}" auswählen
edit.multiupload.title        = Multi-Upload...
edit.multiupload.dialogTitle  = Datei-Upload {0}
edit.changeFile.title         = Datei bearbeiten - {0}
edit.name                     = Name:
edit.new                      = Neu
edit.new.ext                  = Neue Erweiterung
edit.new.title                = Neu anlegen - {0}
edit.noPossibleTargets        = Keine möglichen Ziele gefunden.
##edit.possibleTargets        = aus. Mögliche Ziele sind:
edit.required.marker          = *
edit.required.hintMessage     = Pflichtfelder sind mit einem * markiert.
```



edit.save = Speichern
edit.setNewTarget = Begriff/Indiv. anlegen
edit.setNewTargetButton = Relationsziel anlegen
edit.setNewTargetNoDomain = Bitte wählen Sie die Art des neu anzulegenden Ziels aus.
edit.setNewTargetNoName = Bitte geben Sie einen Namen für das neu anzulegende Ziel an.
edit.setNewTargetTitle = Neues Relationsziel anlegen
edit.setNewTargetType = Type
edit.setNewTargetName = Name
edit.setTargets = Relationsziel(e) auswählen
edit.setTargets.noHits = Keine
edit.setTargets.hits = Treffer
edit.setTargets.page = Seite
edit.setTargets.pageOf = von
edit.setTargets.jumpToPage = Springe zu Seite
edit.setTargets.prevPage = Vorige Seite
edit.setTargets.nextPage = Nächste Seite
edit.selectExtConcept = Bitte wählen Sie einen Typ
edit.selectCoreTopic = Kernobjekt auswählen
edit.noPossibleCoreTopics = Keine möglichen Kernobjekte gefunden
edit.createExtensionNoCore = Bitte wählen Sie das Kernobjekt für die neue Erweiterung
edit.findCoreTopic = Mögliche Kernobjekte suchen...
edit.createExtensionTarget = Erweiterung anlegen

edit.title = Begriff/Indiv. bearbeiten
edit.title.existing = {0}
edit.title.newConcept = Neuer Unterbegriff
edit.title.newInstance = Neue Instanz von {0}
edit.top = Seitenanfang

empty = leer

error.topicForm.emptyField = Bitte geben Sie einen Wert für dieses Feld ein!
error.setTarget.noTargetSelected = Bitte wählen Sie mindestens ein Relationsziel aus!

error.exit = Zurück zur Startseite
error.header = Fehler
error.iframes = Ihr Browser kann keine eingebetteten Frames darstellen...
error.noCookies.message = Um das Knowledge-Portal benutzen zu können, muss Ihr Browser Cookies akzeptieren. Dies ist momentan nicht der Fall. Bitte aktivieren Sie Cookies in Ihren Browsereinstellungen und laden Sie dann die Seite neu\!
error.noCookies.title = Cookies sind nicht aktiviert!
error.noDocHit = Dieser Treffer wurde nicht als Dokument erkannt. Dies kann daran liegen, dass der Nextbot-Index asynchron ist.
error.noJS.message = Um das Knowledge-Portal benutzen zu können, muss JavaScript in Ihrem Browser aktiviert sein. Dies ist momentan nicht der Fall. Bitte aktivieren Sie JavaScript und laden Sie dann die Seite neu.
error.noJS.title = JavaScript ist nicht aktiviert!
error.search.KEMActionError = Fehler beim Suchen: {0}



error.search.KTooManyHitsError = Zu viele Treffer!
error.search.primaryDocs = Fehler beim Suchen der primären Dokumente\!
error.title = Fehler
error.unavailable = Die Applikation konnte nicht initialisiert werden,
bitte wenden Sie sich an den Systemadministrator!
Weitere Informationen zu
diesem Fehler wurden im Anwendungsprotokoll verzeichnet.
error.unauthorized = Login fehlgeschlagen. Bitte wenden Sie sich an den
Systemadministrator.
Weitere Informationen zu diesem Fehler wurden im
Anwendungsprotokoll verzeichnet.
error.unexpected = Es ist leider ein Fehler aufgetreten: {0}
error.catched = Es ist leider ein Fehler aufgetreten: {0}
error.wrongUser = Das abgefragte Objekt ist nicht dem eingeloggten
Benutzer zugeordnet.

#errors.header = <h3>Fehler:</h3>
#errors.footer = <hr>Es ist leider ein Fehler aufgetreten. Wir bedanken uns
für Ihr Verständnis!
Auf Wiedersehen!

error.edit.noAction = Es wurde keine Folge-Action angegeben.
error.deleteAction = Objekt konnte nicht gelöscht werden: {0}
error.editTopicAction = Objekt konnte nicht bearbeitet werden: {0}
error.genericAction = Session konnte nicht initialisiert werden: {0}
error.commitAction = Session konnte nicht committed werden: {0}
error.typeDispatcherAction = TypeDispatcher Fehler: {0}
error.noKObjectEditor = Root-Editor ist kein KObject-Editor
error.noRootEditor = Request enthält keinen Root-Editor
error.noListEditor = Root-Editor ist kein KEMListEditor, sondern: {0}
error.noIndex = Request enthält keinen Index
error.indexNoNumber = Index ist keine Zahl: {0}
error.indexNegativ = Index ist negativ: {0}
error.login = Fehler beim Login von Benutzer '{1}'
an Volume '{0}': {2}
error.userNotFound = Benutzer '{1}' in Volume '{0}' nicht gefunden
error.noUniqueUser = Benutzer '{1}' in Volume '{0}' ist nicht eindeutig
error.wrongPassword = Das Passwort für Benutzer '{1}' ist falsch
error.userLocked = Der Benutzer '{1}' wurde gesperrt
error.fault = Es ist ein Systemfehler (KEM) aufgetreten: {0}
error.nnSelectTarget = Das ausgewählte Ziel ist für die Relation nicht gültig: {0}
error.maxFileSize = Die maximale Dateigröße wurde überschritten

error.infocart.send = Fehler beim Senden des Warenkorbs: {0}

errors.validate.header = Fault error header
errors.validate.footer = Fault error footer
error.validate.defaultMessage = fault default error message: {0}, {1}
error.validate.3 = Fault error code 3 message: {0}, {1}

error.populate = Dieser Wert konnte nicht gesetzt werden: {0}

error.kpath = Der KPath-Ausdruck <code>{0}</code> konnte nicht ausgeführt
werden: {1}

error.skat.defaultMessage = Skat default error message: {0}, {1}



```
error.skat.3          = <li>Skat error code 3 message: {0}, {1}</li>

login.noUsername      = Es wurde keine Benutzerkennung angegeben
login.noPassword      = Es wurde kein Passwort angegeben
login.noAuthServer    = Authentifikations-Server nicht erreichbar. Bitte versuchen
                        Sie es später noch einmal...
error.noAuthServer    = Authentifikations-Server nicht erreichbar. Bitte versuchen
                        Sie es später noch einmal...

required.name=Name   wird benötigt

newAccount.error.wrongPassword = Passwort und Bestätigung stimmen nicht überein
newAccount.error.userExists    = Ein Benutzer mit dieser Benutzerkennung existiert
                                bereits
newAccount.error.noPassword    = Bitte ein Passwort eingeben
newAccount.error.notAllowed    = Gast-Benutzer dürfen keine neuen Benutzer anlegen

eventLogQueries.error.date     = Bitte geben Sie ein Datum im Format "{0}" ein
eventLogQueries.error.empty    = Bitte geben Sie einen Wert für diesen Parameter ein
eventLogQueries.error.number   = Bitte geben Sie eine Zahl ein

eventLogQueries.execute        = Herunterladen
eventLogQueries.title          = Benutzeraktivitäten
eventLogQueries.topXclicks     = Top-Begriffe
eventLogQueries.topXclicks.from = Von Datum
eventLogQueries.topXclicks.limit = Anzahl Topics
eventLogQueries.topXclicks.to   = Bis Datum
eventLogQueries.topXsearch     = Top-Suchen
eventLogQueries.topXsearch.from = Von Datum
eventLogQueries.topXsearch.limit = Anzahl Topics
eventLogQueries.topXsearch.to   = Bis Datum
eventLogQueries.zeroSearches   = Erfolgreiche Suchen
eventLogQueries.zeroSearches.from = Von Datum
eventLogQueries.zeroSearches.to   = Bis Datum

expertSearch.column.enabled    = Aktiviert?
expertSearch.column.parameter  = Parameter
expertSearch.column.value     = Wert
expertSearch.noParameterDefined = Keine Suchparameter definiert.
expertSearch.parameters       = Suchparameter
expertSearch.results          = Suchergebnisse ({0})
expertSearch.subTitle         = Suche &bdquo;{0}&rdquo;
expertSearch.title            = Expertensuche

explain.title = Erkl\u00E4ren

explanation.attribute.introduction = Direkt gefunden:
explanation.attribute.label        = {0} ({1}%):
explanation.attribute.label.unknown = Unbek. Attribut
explanation.attribute.value        = {0}
explanation.relation               = {0} ({1}%):
explanation.relation.introduction  = Semantisch gefunden über\
explanation.title                  = Erklärung
```



filesize.unknown = k.A.

folder.quickCreate = Neu anlegen...
folder.expertQuery.title = Expertensuche

fulltextHits = Dokumentensuche
fulltextHits.loaded = Dokumentensuche ({0})

fulltextHits.group.email = E-Mails
fulltextHits.group.external = Externe Treffer
fulltextHits.group.internal = Interne Treffer
fulltextHits.sortCriteria = Dokumente sortieren nach:
fulltextHits.sortCriteria.date = Datum
fulltextHits.sortCriteria.quality = Qualität
fulltextHits.submit = Suchen

fulltextSearch.hideExtendedOptions = Erweiterte Optionen ausblenden
fulltextSearch.showExtendedOptions = Erweiterte Optionen einblenden

glasspane.pleaseWait = Bitte warten...

history.empty = Keine Einträge
history.title = Verlauf

#htmlarea.link = In HTML-Editor bearbeiten
htmlarea.link = Bearbeiten
htmlarea.createLink = Attribut anlegen

infocart.add = Element dem Infokorb hinzufügen
infocart.concepts = Begriffe
infocart.elements = Elemente:
infocart.explanation = Zeigt den Status des Infokobs an.
infocart.files = Dateien
infocart.infocart = Infokorb
infocart.mailto = Senden an:
infocart.send = Senden
infocart.show = Anzeigen
infocart.successfullySent = Infokorb erfolgreich gesendet an:
infocart.title = Infokorb
infocart.urls = URLs

langswitch.explanation = Wählen sie eine Sprache
langswitch.select = Wählen
langswitch.title = Sprachauswahl

lastModified.unknown = k.A.

login.couldNotLogin = Anmeldung fehlgeschlagen!
login.currentUser.label = Aktueller Benutzer
login.guestLogin = Als Gast anmelden
login.id = Benutzerkennung
login.login = Anmelden



login.logout = Abmelden
login.pageTitle = Anmelden
login.passwd = Passwort

mail.attachments.empty = Kein Anhang.
mail.attachments.title = Anhänge
mail.emptySubject = Kein Betreff
mail.from = Von
mail.htmlContent.empty = Diese Mail enthält keine HTML-Inhalte.
mail.sender = Absender
mail.sentDate = Datum
mail.textContent.empty = Diese Mail enthält keinen Text.
mail.title = E-Mail anzeigen
mail.to = An
mail.unknownValue = k.A.

multiupload.button = Multiupload

netnav.showIn = zeige in Net-Navigator
netnav.title = Net-Navigator

newAccount.commitLabel = Benutzer anlegen
newAccount.fullName = Voller Name
newAccount.linkLabel = Neuen Benutzer anlegen
newAccount.passwdConfirm = Passwort bestätigen

outlook.open.title = Diese Mail in MS-Outlook öffnen

primaryDocs = Zentrale Dokumente ({0})

primaryDocs.confirmDemote = Möchten Sie dieses Dokument wirklich abstufen?
primaryDocs.empty = Keine Treffer
primaryDocs.new = Neues Zentraldokument erzeugen

primaryDocsNT = Zentrale Dokumente (nicht trainierend)

printVersion.printLink = Druck starten

promote.selectConcept.title = Promote - Konzept des neuen Dokumentindividuums
wählen
promote.selectRelation.noRelation = Das Dokument konnte nicht promotet werden.
Möglicherweise handelt es sich bereits um ein Zentraldokument dieses Themas.
promote.selectRelation.title = Promote - Relation auswählen

promoteDocumentForm.cancel = Abbrechen
promoteDocumentForm.concept = Bitte wählen Sie das Konzept für das neue
Dokumentenindividuum aus\
promoteDocumentForm.concept.select = Bitte wählen Sie ein Konzept\
promoteDocumentForm.submit = OK
promoteDocumentForm.title = Zentrales Dokument hinzufügen

quickSearch.param.ignoreChoice = - ignorieren -



quicklinks.newPerson = Neue Person
quicklinks.title = Quicklinks
quicklinks.welcomePage = Startseite

relatedFulltextHits.empty = Keine Treffer

renderer.relation.targetNotReadable = Relationsziel nicht sichtbar

results.docs.title = Dokument-Treffer
results.empty = Keine Treffer gefunden!
results.semantic.title = Semantische Treffer
results.trigram.title = Meinten Sie...
results.websearch.title = Treffer auf Websites

search.all = Alle Suchen
search.exp.explanation = Hier werden alle Unterordner des Ordners mit
der externen ID 'web' aufgelistet
search.exp.list.backButtonLabel = < Zurück
search.exp.list.leadingText = Wählen Sie eine Anfrage aus
search.exp.list.title = vordefinierte Suchanfragen
search.exp.param.parameter = Parameter für
search.exp.title = Expertensuchen
search.fileExtensionFilter.all = Kein Filter
search.fileExtensionFilter.title = Dateiendung
search.for = zu
search.found = gefunden
search.gif = gfx/suchen.gif
search.next = Weiter
search.person.company = Firma
search.person.name = Name
search.person.phone.mobile = Telefon Mobil
search.person.phone.office = Telefon Firma
search.person.surname = Vorname
search.reset = Zurücksetzen
search.results = Suchergebnisse
search.results.subTitle = Suche nach „{0}”
search.start = Starten
search.text.title = Suchbegriff eingeben
search.title = Suche
search.wait = Suche wird ausgeführt

searchPart.semanticSearch.header = Semantische Suche
searchPart.trigramSearch.header = Meinten Sie...?
searchPart.fulltextSearch.header = Dokumentensuche

secondaryDocs = Sekundärdokumente

selectBox.noSelection = Bitte auswählen

similarDocs = Ähnliche Dokumente
similarDocs.loaded = Ähnliche Dokumente ({0})



similarDocs.empty = Die Ähnlichkeitssuche benötigt mindestens {0} zentrale Dokumente.
similarDocs.emptyButWellTrained = Die Ähnlichkeitssuche hatte keine Ergebnisse.

singleupload.button = Durchsuchen...

snippet.unknown = Keine Vorschau verfügbar

table.search = Durchsuchen:

timeout.text = Ihre Benutzersitzung wurde beendet. Dies kann z.B. passieren, wenn Sie längere Zeit (länger als {0} Minuten) keine Aktionen im Web-Frontend durchgeführt haben.
timeout.title = Sitzung wurde beendet.

topic.createInstance = Neues Individuum erzeugen
topic.createSubconcept = Unterbegriff erzeugen
topic.createExtension = Typ der Erweiterung
topic.delete = Begriff löschen
topic.deleteConfirm = Möchten Sie diesen Begriff wirklich löschen?
topic.downloadDoc = Dokument herunterladen
topic.editConcept = Begriff bearbeiten
topic.editInstance = Individuum bearbeiten
topic.superconcept = Oberbegriff
topic.concept = Begriff
topic.instances = Individuen
topic.print = Drucken
topic.subthemes = Unterthemen
topic.subconcepts = Unterthemen
topic.theme = Thema
topic.unnamed = Ohne Name

translations.de = Deutsch
translations.en = English
translations.nl = Nederlands

userSettings.changePassword = Passwort ändern
userSettings.editProfile = bearbeiten
userSettings.profile = Profil\
userSettings.showProfile = anzeigen

viewtopic.concept = Konzept
viewtopic.confirmDelete = Möchten Sie dieses Objekt wirklich löschen?
viewtopic.delete = Löschen
viewtopic.docs = Keine Nebendokumente? Interner Name der Relation
'ist Thema von': themaVon
viewtopic.document = Dokument
viewtopic.extension = Erweiterung
viewtopic.instance = Individuum
viewtopic.mainDocs = Keine Hauptdokumente? Interner Name der Relation
'ist Hauptthema von': hauptthemaVon
viewtopic.maintain = pflegen



```
viewtopic.newInstance    = neues Individuum
viewtopic.newSubtopic   = neuer Unterbegriff

window.close            = Fenster schließen
window.close.confirm    = Wollen Sie dieses Fenster wirklich schließen?

action.expandContainer.title = Details anzeigen
action.collapseContainer.title = Details ausblenden
```

12.5 Actions und Action-Transformer

12.5.1 Relation erzeugen (CreateRelationAction)

Erzeugt eine neue Relation, wobei die Relationsquelle und das Relationsziel durch die Auswertung von konfigurierbaren KPath-Ausdrücken bestimmt wird. Die KPath-Ausdrücke werden ausgehend von dem Objekt ausgewertet, für welches diese Action konfiguriert wird.

Beispiel:

```
<ActionTransformer
  class="
  com.iviews.knowledgeportal.actions.transformer.
    ConfiguredSourceTargetActionTransformer">
  <Property name="actionClassname"
    value="com.iviews.knowledgeportal.actions.action.CreateRelationAction" />
  <Property name="actionId" value="addFavorite" />
  <Property name="sourcePath" value="userInstance()/id()" />
  <Property name="targetPath" value="id()" />
  <Property name="actionForward" value="script_reload" />
  <Property name="actionType" value="SCRIPT" />
  <ActionProperties>
    <Property name="newRelationInternalName" value="hasFavorite" />
  </ActionProperties>
</ActionTransformer>
```

12.5.2 Relation mit neuem Relationsziel erzeugen (CreateNewTopicAction)

Verknüpft das Objekt, für welches diese Action definiert ist, mit einem neu erzeugten Objekt. Der Typ des Objektes muss mittels der Action-Property "targetConceptName" als interner Name eines Begriffs angegeben werden. Die für die Verknüpfung zu verwendende Relation kann in der Action-Property "relationConceptName" (interner Name) angegeben werden. Soll als Relationsziel ein Individuum erzeugt werden, so muss für die Action-Property "createInstance" der Wert "true" gesetzt werden, soll ein Unterbegriff erzeugt werden, so muss "false" als Wert gesetzt werden.

Beispiel:

```
<ActionTransformer
  class="com.iviews.knowledgeportal.actions.transformer.IdActionTransformer">
```



```
<Property name="actionClassname"
  value="com.iviews.knowledgeportal.actions.action.CreateNewTopicAction" />
<Property name="actionForward"
  value="editRootInfo" />
<ActionProperties>
  <Property name="relationConceptName" value="projectHasMember" />
  <Property name="targetConceptName" value="Person" />
  <Property name="createInstance" value="true" />
</ActionProperties>
</ActionTransformer>
```

12.5.3 Objekt löschen (DeleteTopicAction)

Löscht das Objekt, an dem diese Action konfiguriert ist. Das Icon zur Ausführung dieser Action wird nur dann angezeigt, wenn das Rechtesystem die Löschen-Operation auch erlaubt. Vor dem Löschen wird eine Sicherheitsabfrage durchgeführt.

Beispiel:

```
<ActionTransformer class="
  com.iviews.knowledgeportal.actions.transformer.DeleteTopicIdActionTransformer">
  <Property name="actionClassname"
    value="com.iviews.knowledgeportal.actions.action.DeleteTopicAction" />
  <Property name="actionForward" value="script_reload"/>
</ActionTransformer>
```

12.5.4 KScript-Report ausführen (DownloadReportAction)

Führt einen im Wissensnetz definierten KScript-Report aus und liefert den resultierenden Text als Datei-Download im Browser zurück.

Beispiel:

```
<ActionTransformer
  class="com.iviews.knowledgeportal.actions.transformer.IdActionTransformer">
  <Property name="actionClassname"
    value="com.iviews.knowledgeportal.actions.action.DownloadReportAction" />
  <Property name="actionId" value="downloadVCard" />
  <ActionProperties>
    <Property name="folderName" value="kpReports" />
    <Property name="reportName" value="generateVCard" />
    <Property name="contentType" value="text/x-vcard" />
    <Property name="fileName" value="vcard.vcf" />
  </ActionProperties>
</ActionTransformer>
```



12.5.5 Report ausführen und Blob herunterladen (AjaxReportAndDownloadAction / ab KP 3.8)

Die AjaxReportAndDownloadAction erlaubt die Ausführung eines KScript-Reports zur Aktualisierung bzw. Erzeugung und den anschließenden Download eines Blob-Attributes.

Der Report muss als Ausgabe die DMID des erzeugten bzw. aktualisierten Blob-Attributes zurückliefern. Nach der Ausführung des Reports wird dann zu einer BlobDownloadAction weitergeleitet, die eine HTTP-Download-Response mit den Binärdaten des entsprechenden Blob-Attributes erzeugt.

Um zu bestimmen, welches Objekt als Eingabeobjekt bei der Ausführung des Reports gesetzt werden soll, muss bei der Erzeugung der Action die ID-Eigenschaft auf den entsprechenden Wert gesetzt werden (siehe unten). Dazu muss zur Erzeugung der Action der IdActionTransformer verwendet werden, der im Normalfall die DMID des Topics aus dem Kontext der Action-Erzeugung als ID setzt.

Hier eine Beispielkonfiguration:

```
<ActionTransformer
  class="com.iviews.knowledgeportal.actions.transformer.IdActionTransformer">
  <Property name="actionClassname"
    value="com.iviews.knowledgeportal.actions.action.AjaxReportAndDownloadAction"/>
  <Property name="actionType" value="DIRECT"/>
  <Property name="actionId" value="downloadDraft"/>
  <ActionProperties>
    <Property name="folderName" value="sbbReports"/>
    <Property name="reportName" value="generateSiDispoDraft"/>
    <Property name="withStartObjects" value="true"/>

    <Property name="readOnly" value="false"/>
  </ActionProperties>
</ActionTransformer>
```

Die folgenden Action-Properties können für diese Action konfiguriert werden:

Name	Beschreibung
id	DMID eines Topics im Wissensnetz, welches als Eingabeobjekt bei der Ausführung des KScript-Reports gesetzt werden soll. Diese Eigenschaft wird normalerweise vom IdActionTransformer gesetzt und sollte daher im Regelfall nicht explizit als ActionProperty angegeben werden.
folderName	External-ID des Report-Ordnern im Wissensnetz, der den auszuführenden Report enthält (Pflichtfeld)
reportName	Name des auszuführenden Reports (Pflichtfeld)
withStartObjects	Gibt an, ob das Topic aus dem Kontext der Action-Erzeugung als Start-Objekt an den Report übergeben werden soll Standardwert: true.



readOnly	Gibt an, ob der Report innerhalb einer Lese- (true) oder einer Schreibtransaktion (false) ausgeführt werden soll. Standardwert: false.
----------	--

12.5.6 Blob herunterladen (DownloadBlobAction / ab KP 3.8)

Die DownloadBlobAction generiert eine HTTP-Response zum Download der Binärdaten eines Blob-Attributes.

Um zu bestimmen, für welches Blob-Attribut die Download-Response erzeugt werden soll, muss bei der Erzeugung der Action die ID-Eigenschaft auf den entsprechenden Wert (DMID des Attributes) gesetzt werden. Zur Erzeugung der Action kann im Normalfall der IdActionTransformer verwendet werden, der die DMID des Objektes aus dem Kontext der Action-Erzeugung als ID setzt (Vorsicht: Dies kann je nach Stelle der Action-Konfiguration auch ein Individuum oder ein Konzept sein).

Das folgende Konfigurationsbeispiel erzeugt eine Download-Response für ein bestimmtes Attribut (myAttribute) an einem Topic, so dass man die Action in den ComponentActionTransformers einer ViewConfig definieren könnte:

```
<ActionTransformer
  class="com.iviews.knowledgeportal.actions.transformer.IdActionTransformer">
  <Property name="actionClassname"
    value="com.iviews.knowledgeportal.actions.action.DownloadBlobAction"/>
  <Property name="actionType" value="DIRECT"/>
  <Property name="actionId" value="download"/>
  <Property name="kpath" value="@$report$/id()/singleElement()"/>
  <ActionProperties>
    <Property name="fileName" value="report.pdf"/>
    <Property name="contentDisposition" value="INLINE"/>
  </ActionProperties>
</ActionTransformer>
```

In der folgenden Tabelle sind die ActionProperties beschrieben, welche für die Download-BlobAction konfiguriert werden können:

Name	Beschreibung
id	ID des Blob-Attributes dessen Binärdaten heruntergeladen werden sollen (Pflichtfeld). Die ID wird normalerweise von einem IdActionTransformer dynamisch bestimmt und gesetzt und muss daher nicht explizit angegeben werden.
fileName	Name der heruntergeladenen Datei. Standardwert: Dateiname aus dem Blob-Attribut im Wissensnetz.



contentDisposition	Gibt an, ob eine HTTP-Download-Reponse vom Typ "ATTACHMENT" (Speichern der Datei auf der Festplatte) oder "INLINE" (Öffnen und Anzeigen der Datei im Browser) erzeugt wird. Standardwert: ATTACHMENT.
--------------------	---

12.5.7 KPath-Ausdruck ausführen (ab 3.8.5)

Die KPathEvalAction kann einen beliebigen KPath ausführen und das Ergebnis weiterverwenden. Damit können beispielsweise KScript-Funktionen ausgeführt werden oder Behaviour-Calls aufgerufen werden.

Beispiel:

```
<ActionTransformer
  class="com.iviews.knowledgeportal.actions.transformer.
    KPathEvalActionTransformer">
  <Property name="actionType" value="SCRIPT" />
  <Property name="actionId" value="openConcept" />
  <Property name="actionForward" value="kpathEval/showResultDmid" />
  <ActionProperties>
    <Property name="kpath" value="script->aFunction()" />
    <Property name="edit" value="true" />
  </ActionProperties>
</ActionTransformer>
```

Dieses Beispiel führt die Funktion "aFunction()" im Registrierten Script "script" aus. Es wird erwartet, dass diese Funktion eine DMID zurückliefert, welche dann als Detailseite geöffnet wird.

Beispiel 2:

Die KPathEvalAction kann auch in Suchergebnisseiten (ResultViewConfig) eingebunden werden. Hier ein Beispiel, wie eine Action zum Aufruf der Detailseite des Topics für Attributtreffer (z.B. auf BLOB-Attributen bei der Lucene-Volltextsuche) konfiguriert werden kann:

```
<SearchPartResult id="fulltextSearch">
  <RenderHint name="showDocMetas" value="false" />
  <HitActions>
    <KPathEvalAction kpath="topic()/topic()/id()"
      actionId="showTopic"
      forward="kpathEval/showResultDmid"
      type="SCRIPT" />
  </HitActions>
</SearchPartResult>
```

ActionProperties:



Name	Beschreibung
kpath	Auszuführender Kpath: Eingabe des Pfades ist das aktuelle Objekt. Die Ausgabe wird im "actionForward" verwendet.
edit	true: KPath wird in einer Edit-Session ausgeführt und kann das Wissensnetz modifizieren. false: KPath liest nur und wird in einer Read-Session ausgeführt.

ActionForwards

Name (kpathEval/...)	action-Type	Erwartete Rückgabe	ab	Beschreibung
showResultDmid	SCRIPT	Dmid	3.8.5	Öffnen des entsprechenden Objektes in einer Detailseite

12.5.8 KPath-Ausdruck ausführen und Session offenhalten (ab 3.8.6)

Die KPathRootEditorAction führt einen KPath in einer Edit-Session aus ohne die Session danach zu schließen. Sie kann dazu verwendet werden um zum Beispiel Objekte per KScript zu erzeugen und sie danach direkt zu editieren.

Konfiguration

```
<ActionTransformer
  class="com.iviews.knowledgeportal.actions.transformer.
    KPathRootEditorActionTransformer">
  <Property name="actionForward" value="editRootInfo"/>
  <Property name="actionId" value="cloneInstance"/>
  <Property name="actionType" value="SCRIPT"/>
  <ActionProperties>
    <Property name="kpath" value="clone->cloneInstance('Copy of ')/>
    [ <Property name="showLastFromHistory" value="true" /> ]
    [ <Property name="layoutCategory" value="layoutCategory " /> ]
  </ActionProperties>
</ActionTransformer>
```

ActionProperty	Benötigt	Beschreibung
kpath	ja	Auszuführender KPath. Dieser KPath kann KScript-Funktionen aufrufen, die Objekte anlegen, modifizieren oder löschen.



showLast-FromHistory	nein	Wird z.B. vom Forward "editRootInfo" verwendet um zu definieren ob nach dem Schließen des Pflegefenster das bearbeitete Objekt ("false") oder das zuletzt betrachtete Objekt ("true") angezeigt werden soll.
layoutCategory	nein	Wird z.B. vom Forward "editRootInfo" verwendet um eine alternative Layout-Kategorie zu spezifizieren, die für die Edit-Config der Pflegeseite verwendet werden soll.

Beispiel: Klonen von Individuen

Die oben dargestellte Konfiguration kann zum Klonen von Individuen verwendet werden, wenn folgendes KScript als Registriertes Skript "clone" im Wissensnetz angelegt wird.

```
<Script xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="http://i-views.de/xml/ns/kscript">

  <!--
    Input: Instance
    Creates a copy of the instance, with the same attributes and relations
    and extensions if possible.
    If parts of the topic cannot be replicated properly, they are just silently
    ignored.
    Return value: The cloned instance

    Note that concepts cannot be cloned with this function.
  ->
  <Function name="cloneInstance" arguments="prefix">
    <SetVariable variable="source" value="." />
    <CreateInstance concept="var('source')/concept()"
      name="{var('prefix') + var('source')/name()}">
      <SetVariableValue variable="target" value="." />
    </CreateInstance>

    <Call function="copyProperties">
      <parameter name="source" value="var('source')" />
      <parameter name="target" value="var('target')" />
    </Call>

    <Path path="var('source')">
      <Extensions>
        <Each>
          <SetVariable variable="extension" value="." />
          <CreateExtension concept="var(extension)/concept()"
            core="var('target')">
            <Call function="copyProperties">
              <parameter name="source" value="var('extension')" />
              <parameter name="target" value="." />
            </Call>
          </CreateExtension>
        </Each>
      </Extensions>
    </Path>
  </Function>
</Script>
```



```
</Path>
  <Return value="var(target)" />
</Function>

<Function name="copyProperties" arguments="source, target">
  <Call function="copyAttributes">
    <parameter name="source" value="var('source')" />
    <parameter name="target" value="var('target')" />
  </Call>
  <Call function="copyRelations">
    <parameter name="source" value="var('source')" />
    <parameter name="target" value="var('target')" />
  </Call>
</Function>

<Function name="copyAttributes" arguments="source, target">
  <Path path="var('source')">
    <Attributes includeExtensions="false">
      <Each>
        <Try>
          <do>
            <CopyAttribute targetPath="var('target')"
              modifyExisting="false" withMetas="true"
              property="." />
          </do>
        </do>
      </catch></catch>
    </Try>
  </Each>
</Attributes>
</Path>
</Function>

<Function name="copyRelations" arguments="source, target">
  <Path path="var('source')">
    <Relations includeExtensions="false">
      <Each>
        <Try>
          <do>
            <CopyRelation targetPath="var('target')"
              modifyExisting="false" withMetas="true"
              property="." />
          </do>
        </do>
      </catch></catch>
    </Try>
  </Each>
</Relations>
</Path>

</Function>
</Script>
```



12.5.9 Mailto-Links erzeugen - EmailLinkIdActionTransformer (ab KP 3.8.8)

Dieser Action-Transformer dient dazu, Mailto-Links im Web-Frontend zu erzeugen, die beim Anklicken dazu führen, dass das E-Mail-Programm auf dem Client-Rechner gestartet und der Dialog zum Erstellen einer neuen E-Mail-Nachricht angezeigt wird.

Der Transformer kann so konfiguriert werden, dass die Felder zur Eingabe der Empfängeradresse ("to"), Empfängeradresse in Kopie ("cc"), Betreff ("subject") und Nachrichtentext ("body") mit Inhalten aus dem Wissensnetz vorbefüllt werden, so dass der Benutzer diese Felder nicht mehr ausfüllen oder zumindest nur noch nachbearbeiten muss.

Hier ein Überblick über die Eigenschaften des Transformers:

Name	Beschreibung
toKPath	KPath-Ausdruck zur Bestimmung des Inhaltes des Feldes "to" der neuen E-Mail (Empfängeradresse). Defaultwert: "" (KPath-Ausdruck für Leerstring)
ccKPath	KPath-Ausdruck zur Bestimmung des Inhaltes des Feldes "cc" der neuen E-Mail (Empfängeradresse in Kopie). Defaultwert: "" (KPath-Ausdruck für Leerstring)
subjectKPath	KPath-Ausdruck zur Bestimmung des Inhaltes des Feldes "subject" der neuen E-Mail (Betreff).
bodyKPath	KPath-Ausdruck zur Bestimmung des Inhaltes des Feldes "body" der neuen E-Mail (Nachrichtentext).

Transformer-Eigenschaften können in der Action-Config (siehe unten) mit dem <Property>-Element gesetzt werden. Es **muss** für jede der oben genannten Eigenschaften ein gültiger KPath-Ausdruck hinterlegt sein, andernfalls wird beim Aufruf des Transformers ein Fehler geworfen (IllegalStateException).

Das folgende Beispiel konfiguriert einen EmailLinkIdActionTransformer, wobei die Inhalt für das Feld "subject" und "body" aus einem Attribut im Wissensnetz geladen werden, wobei der Platzhalter "<1s>" durch die ID (DMID) und der Platzhalter "<2s>" durch den Namen des Topics ersetzt werden, in dessen Kontext die KPath-Ausdrücke ausgewertet werden (Details zur Funktion "expand" in der KPath/KScript-Dokumentation):

```
<ActionTransformer
class="com.iviews.knowledgeportal.actions.transformer.EmailLinkIdActionTransformer">
  <Property name="actionId" value="mailElement"/>
  <Property name="subjectKPath"
    value="expand(//\$email$/@$subject$/singleElement()/valueString(),id(),name())"/>
  <Property name="bodyKPath"
    value="expand(//\$email$/@$body$/singleElement(), id(), name())"/>
</ActionTransformer>
```



12.5.10 Objektliste als Excel-Datei herunterladen

Ab Version 3.9.7 bzw. 3.8.33 des Knowledge-Portal können Excel-Dateien aus einer Vorlage und einer Objektliste generiert werden.

Voraussetzung für das Funktionieren der Action ist:

1. eine in der ViewConfig konfigurierte Objektliste,
2. eine Excel-Druckvorlage für den Begriff der Objektliste
3. eine konfigurierte Rest-Bridge, diese benötigt einen aktivierten baseService und trustedLoginEnabled=true

Konfiguration

Die folgende Konfiguration erzeugt eine einfache Objektliste mit einen Knopf für den Excel-Download

```
<ObjectList conceptInternalName="person" tableName="tableName" forInstances="true" inheritanceDis
  <ComponentActionTransformers>
    <ActionTransformer class="com.iviews.knowledgeportal.actions.transformer.objectlist.Object
      <ActionProperties>
        <Property name="template" value="templateLocator" />
      </ActionProperties>
    </ActionTransformer>
  </ComponentActionTransformers>
</ObjectList>
```

ActionProperties:

Name	Beschreibung	benötigt
template	Referenz auf die zu verwendende Druckvorlage. Die Property muss eines der Formate haben, die als Referenz in RestBridge-URLs verwendet werden können (siehe REST-Bridge, Vordefiniere Services, Allgemeines)	ja
filename	Dateiname, der beim Herunterladen der Datei zum Speichern angeboten werden soll (default: "download.xlsx")	nein

12.5.11 Action Render Modes

Actions können im Knowledge-Portal auf unterschiedliche Arten dargestellt werden. Die Darstellung wird über die Action Property renderMode konfiguriert.



```
<ActionTransformer
  class="com.iviews.knowledgeportal.actions.transformer.IdActionTransformer">
  <Property name="actionId" value="meineaction" />
  <ActionProperties>
    <Property name="renderMode" value="button" />
  </ActionProperties>
</ActionTransformer>
```

Folgende Render-Modes sind im KP verfügbar:

Render-Modus	Beschreibung
 <i>icon</i>	Stellt die Action als konfigurierbares Icon dar. Für die Konfiguration des Icons s. Kapitel (Standard-Konfiguration von Icons". Als Schlüssel wird dabei <i>action.<ACTIONID></i> verwendet.
 <i>button</i>	Stellt die Action als Button dar. Für die Konfiguration des angezeigten Textes s. Kapitel "Texte konfigurieren". Als Schlüssel wird dabei <i>action.<ACTIONID>.title</i> für das title Attribut und <i>action.<ACTIONID>.text</i> für die Beschriftung verwendet.
 <i>texticon</i>	Stellt die Action Icon mit Text dar. Text, Titel und Icon werden wie bei den Render-Modus <i>icon</i> und <i>button</i> konfiguriert. Typischerweise sollte das CSS noch entsprechend angepasst werden. Dabei sollte z.B. (ab KP 3.9.16) Schrift zur Iconfarbe gewählt werden, oder alle Actions auf die gleiche Größe eingestellt werden. Anpassungen können über die CSS Klasse <i>texticon</i> vorgenommen werden: <pre>.texticon, .texticon:active, .texticon:focus { padding-left: 26px; line-height: 24px; background-repeat: no-repeat; border: 1px solid transparent; background-color: transparent; white-space: nowrap; text-align: left; outline: none; } .texticon:hover { text-decoration: underline; border: 1px solid transparent; background-color: transparent; }</pre>



13 Administration

13.1 Rechtekonfiguration

Verschiedene Aktionen im Knowledge-Portal benötigen das Benutzerrecht "kpAdmin". Somit kann mittels des Rechtesystems konfiguriert werden, welche Benutzer administrative Aufgaben im Portal durchführen können.

Anlegen

Das Recht wird automatisch durch das Einspielen der Knowledge-Portal Komponente im Rechtssystem angelegt.

Konfiguration

Die Konfiguration erfolgt auf Benutzerebene. Mittels des Rechtssystemparameters "Benutzer" kann somit einfach gesteuert werden, wer die Berechtigung zu administrativen Aufgaben hat.

13.2 Portalstatus

Version: Ab KP 3.8

Ab der Version 3.8 ist es möglich, wichtige Informationen über das Portal zur Laufzeit abzufragen.

Unter der Adresse `/portalStatus.skat` können diese von berechtigten Benutzern abgefragt werden. Es sind nur Anwender berechtigt, die über das Benutzerrecht "kpAdmin" verfügen.

Aufgelistet werden:

- Knowledge Portal Version: Die Version des Portals und das Profil, das zum erstellen des Paketes verwendet wurde.
- Volume-Name: Der Name des konfigurierten Netzes
- K-Infinity core Version: Die Hauptversion von K-Infinity, bspw. 3.2.0 .
- Java KEM API Version: Die Version der von Java verwendeten KEM-Schnittstelle.
- Konfigurierte Bridges: Die Liste der konfigurierten Bridges, inkl. Port und aktueller Zustand der Verbindung.
- Start des Portals, als Tag und Datum
- Laufzeit: Die Zeit, die seit dem Start vergangen ist.

14 Kochbuch

In diesem Kapitel werden Anforderungen angesprochen, die häufig auftreten, aber komplexere Anpassungen in der Konfiguration erfordern.

14.1 Zeitpunkt des letzten Logins in einem Attribut merken

Manchmal ist es nötig, eine Expertensuche als Kontextbox zu konfigurieren, die das Datum des letzten Logins des Benutzers verwenden. Dazu muss man das Datum in einem Attribut merken, damit es dann in einer Expertensuche verwendet werden kann.



Attribut mit Login-Trigger befüllen

Als erstes muss in der Schema-Definition für Individuen von Web-Nutzer ein Attribut "Letztes Login" mit dem internen Namen "login" angelegt werden.

Dann muss der Login-Trigger in der Login-Komponente konfiguriert werden. Ab Knowledge-Portal 3.6 geschieht dies im Attribut "componentsXml" am KP-Konfigurations-Begriff. Das auszuführende KScript muss HTML-escaped werden. Daher ist es ratsam, den eigentlichen Code in ein Registriertes Skript auszulagern:

```
<StandardLogin afterUserLoggedIn="&lt;Script&gt;&lt;CallScript  
  script="&quot;afterUserLoggedIn&quot; /&gt;&lt;/Script&gt;" ... />
```

Das Registrierte Skript "afterUserLoggedIn" muss dann folgenden Inhalt haben:

```
<Script>  
  <Do>  
    <CreateAttribute modifyExisting="true"  
      name="$lastLogin$"  
      value="currentTimestamp()" />  
  </Do>  
</Script>
```

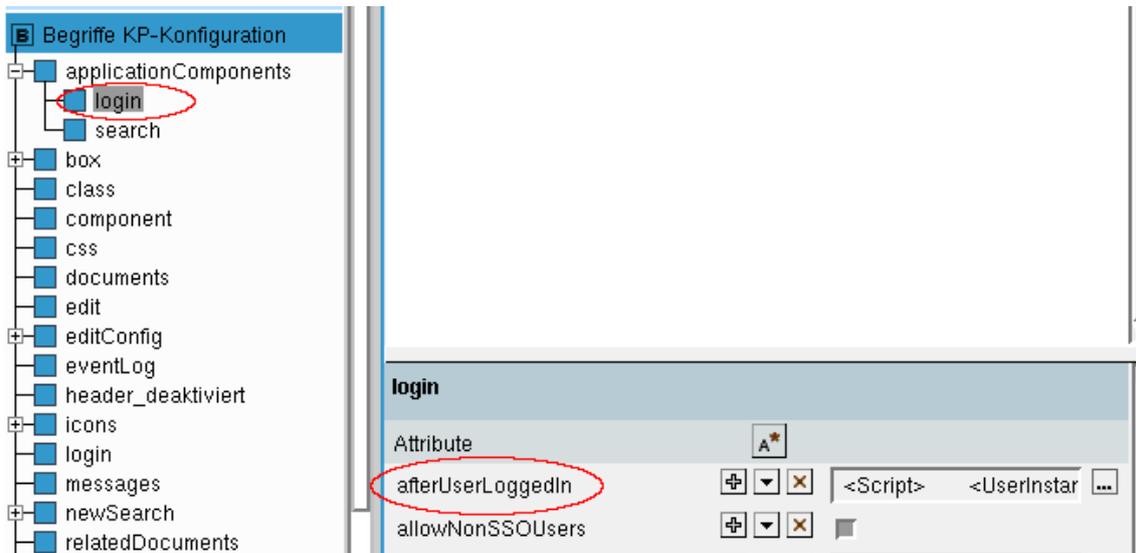
Für Knowledge-Portal 3.5:

In der Version 3.5 der Knowledge-Portal funktioniert diese Einstellung anders: Hier muss man das Datum in einem Attribut merken, damit es dann in einer Expertensuche verwendet werden kann.

Attribut mit Login-Trigger befüllen

Als erstes muss in der Schema-Definition für Individuen von Web-Nutzer ein Attribut "Letztes Login" mit dem internen Namen "login" angelegt werden.

Dann muss der Login-Trigger in der Login-Komponente konfiguriert werden. Falls nicht vorhanden, muss dazu in der Schemadefinition für Begriffe von "applicationComponents/config" unter dem KP-Config-Begriff, ein Zeichenketten-Attribut mit dem Namen "afterUserLoggedIn" angelegt werden.



Dieses Attribut legen Sie am Begriff "login" an und befüllen es mit dem Skript

```
<Script>
  <UserInstance>
    <Output><path path="label()"></path><cr/></Output>
    <CreateAttribute modifyExisting="true"
      name="$lastLogin$"
      value="currentTimestamp()" />
    <Output>Last login date: <path path="@$lastLogin$"/></Output>
  </UserInstance>
</Script>
```

15 Knowledge-Portal 4.0 - Versionshinweise

15.1 Was ist neu?

15.1.1 Box-Konfiguration

In früheren Knowledge-Portal-Versionen wurde die Box-Konfiguration dadurch verkompliziert, dass man je nach Box-Typ für die Konfiguration einen Begriff (Expertensuch- und KPath-Boxen) oder ein Individuum (Micro-Edit- und Topic-Detail-Boxen) anlegen musste. Die Konfiguration wurde zusätzlich noch dadurch erschwert, dass die konfigurierbaren Eigenschaften der Boxen (z.B. der Standard-Anzeigebereich) an Begriffen und an Individuen auf unterschiedliche Attribute abgebildet waren (bspw. "defaultDisplayArea" bei Begriffen und "Anzeigebereich" bei Micro-Edit-Boxen).

Im Knowledge-Portal 4.0 wurde die Box-Konfiguration nun vereinheitlicht und komplett auf eine individuenbasierte Konfiguration umgestellt. Für jede benutzerdefinierte Box gibt es in der neuen Knowledge-Portal-Konfiguration nun ein Individuum. Dabei hat sich die Begriffshierarchie des Konfigurationsmodells für die Boxen etwas verändert. Hier ein Überblick:



Inhalt der Kontextbox	Konfigurationsbegriff KP 3.9	Konfigurationsbegriff KP 4.0 (individuenfähig)
Expertensuchen	box > context	box > collectionBox > contextFolder
Einfache Suchen und Suchen über Nutch bzw. die XSLTSearchComponent)	box > searchManager	box > collectionBox > searchManagerBox
KPath, der eine Topic-Menge als Ergebnis zurückliefert	box > kpathContext	box > collectionBox > kpathContextBox > kpathTopicsBox
KPath, der eine Menge von einfachen Werten (Zahlen, Zeichenketten) zurückliefert	box > kpathContext	box > collectionBox > kpathContextBox > kpathValuesBox
Relationsziele (Micro-Edit)	box > microedit	box > collectionBox > microEdit
ViewConfig-XML	box > topicDetail	box > topicDetail

Außerdem ist es nicht mehr notwendig (und auch nicht zu empfehlen), Attribute oder Relationen zur Box-Konfiguration manuell anzulegen. Das komplette Schema für die Box-Konfiguration wird nun beim Einrichten der Knowledge-Portal-Komponente im Admin-Tool im Wissensnetz angelegt.

Die bestehende begriffsbasierte Box-Konfiguration aus dem KP 3.9 wird beim Upgrade auf die Version 4.0 automatisch in Individuen umgewandelt. Details hierzu sind im nachfolgenden Kapitel "Migration" beschrieben.

Konfiguration von Namen und Übersetzungen

Die Namen von Boxen werden im Knowledge-Portal 4.0 nur noch über das spezielle Namensattribut "Boxname" vergeben. Es ist also keine Übersetzung der Namen über die Message-Konfiguration notwendig. Die konfigurierten Übersetzungen aus den Messages werden - soweit möglich - bei der Migration auf die Version 4.0 automatisch übernommen.

Zuordnung von Boxen zu Detailseiten

Die Zuordnung von Boxen zu den Detailseiten von Begriffen und Individuen wird im KP 4.0 nun generell über die Relation "Box wird angezeigt für Individuen von" und "Box wird angezeigt für Unterbegriffe von" abgewickelt, die an Box-Individuen angelegt werden kann. Damit werden die folgenden (alten) Konfigurationsattribute an Begriffen überflüssig:

- contextSearchFolderName
- contextSearchFolderNameEdit
- kpathContextBoxDefinitions
- kpathContextBoxDefinitionsEdit
- contextSearchName



- contextSearchNameEdit

Die Möglichkeit zur Konfiguration von Kontextboxen für die Pflegeseite fällt damit weg. Es wurde jedoch auch nur sehr selten Gebrauch davon gemacht.

Konfiguration von Actions

Die Konfiguration der Actions, die an den Boxen selbst oder an den Einträgen innerhalb der Box angeboten werden sollen, lassen sich im KP 4.0 über die Relationen "Box verwendet Action-Transformer" und "Boxeinträge verwenden ActionTransformer" konfigurieren. Als Relationsziel müssen Individuen des Begriffs "actionTransformer" angelegt werden. Auch hier werden bestehende Box-Action- bzw. Box-Item-Action-Konfigurationen in Form von Unterbegriffen bei der Migration in Individuen umgewandelt. Der Vorteil der Konfiguration über Individuen liegt darin, dass man eine Action-Transformer-Konfiguration durchaus auch mit mehreren Boxen verknüpfen kann, wenn man die gleiche Action in mehreren Boxen einsetzen.

Standardeinstellungen für Boxen

Die Konfiguration von Standardeinstellungen für Boxen erfolgt im KP 4.0 weiterhin über Eigenschaften, die an Unterbegriffen von "box" angelegt werden. Hierfür gilt nun jedoch das selbe Konfigurationsschema wie bei den Box-Individuen. Boxeinstellungen an Box-(Unter-)Begriffen werden auf alle untergeordneten Individuen vererbt.

Konfiguration der standardmäßig vom Knowledge-Portal bereitgestellten Boxen

Die vom Knowledge-Portal bereitgestellten Boxen werden wie in früheren KP-Versionen über Begriffe konfiguriert, da es ansonsten zu Konflikten mit benutzerdefinierten Box-Individuen kommen könnte. Allerdings werden die Konfigurationsbegriffe nun automatisch über die Knowledge-Portal-Komponente im Admin-Tool angelegt, so dass man die Begriffe nicht mehr per Hand anlegen muss. Das Konfigurationsschema ist wie bereits oben beschrieben das selbe, wie bei den Box-Individuen. Hier eine Übersicht über die Konfigurationsbegriffe für Standard-Boxen:

15.1.2 LESS statt CSS

Das KP 4.0 verwendet zur Definition von Stylesheets anstelle von reinem CSS die dynamische Stylesheet-Sprache "LESS" (siehe <http://www.lesscss.de>). LESS erweitert die normale CSS-Syntax um einige neue Sprachelemente wie z.B. Variablen und Funktionen (z.B. "lighten" oder "darken" zur Berechnung von abgeleiteten Farbwerten). Das CSS für das Web-Frontend wird dann aus den LESS-Definitionen generiert.

Um die Anpassung von Farben zu vereinfachen, wurden die Grundeinstellungen hierfür in eine Reihe von LESS-Variablen ausgelagert, die in sämtlichen KP-Stylesheets verwendet wer-



den:

```
@themeBgColor: white;
@themeBgColorLight: darken(@themeBgColor, 15%);
@themeHighlightColor: #003061;
@themeBgHeaderColor: @themeHighlightColor;
@themeHeaderTextColor: contrast(@themeBgHeaderColor);
@themeTextColor: black; @themeLinkColor: @themeTextColor;
@themeErrorColor: red;
@themeWarningColor: orange;
@themeSuccessColor: green;
@themeSeparatorColor: darken(@themeBgColor, @borderDarken);
```

Durch die Anpassung der o.g. Variablen lässt sich somit das gesamte Farbklima des KPs beeinflussen. Die Grundeinstellungen für Schriftart und Schriftgröße sowie für die Darstellung von Rahmen wurden in die folgenden LESS-Variablen ausgelagert:

```
@borderDarken: 25%;
@ui_fsDefault: 15px;
@ui_ffDefault: Arial, Verdana, sans-serif;
```

Es ist im KP 4.0 auch nicht mehr notwendig, ein spezielles JQuery-UI-Theme als Bestandteil von Themes in den Volume-Ressourcen zu hinterlegen, da auch dieses automatisch von den Grundeinstellungen im KP abgeleitet wird.

Möchte man in den LESS-Definitionen auf das Verzeichnis der Volume-Ressourcen verweisen (etwa zum Einbinden von Grafiken), so kann hierfür die LESS-Variablen "@volumeResourcesPath" verwendet werden. Beispiel:

```
div#headerRight {
    background-image: url('@{volumeResourcesPath}/gfx/iviews_logo_web500.jpg');
    width: 205px;
}
```

Eine komplette Liste der im KP 4.0 verwendeten LESS-Variablen ist am Ende dieses Dokumentes im Kapitel "Referenz" angehängt.

Umstellungen bei den Stylesheet-Definitionen im Wissensnetz

Die CSS-Definitionen, die im Wissensnetz am Begriff "css" in den Attributen "all", "ie" und "other" abgelegt sind, werden beim Upgrade auf das Knowledge-Portal 4.0 zum Begriff "less" kopiert. Die Attribute am Begriff "css" werden nicht mehr berücksichtigt und sollten nach dem Upgrade manuell entfernt werden. Da das Knowledge-Portal den Internet-Explorer 6 nicht mehr unterstützt, können keine IE6-spezifischen Stylesheets mehr im Attribut "ie6" angegeben werden.

Umstellungen innerhalb von Knowledge-Portal-Themes

Durch die Umstellung von CSS auf LESS haben sich innerhalb der Themes in den Volume-Ressourcen (Unterverzeichnis "themes" in der "volumeResources.zip") folgende Änderungen ergeben:

- Das Unterverzeichnis "css" wird nicht mehr verwendet. Statt dessen müssen die Stylesheet-Definitionen im Unterverzeichnis "less" abgelegt werden. Die Dateien können beliebig benannt werden, der Dateiname muss aber auf ".less" enden. Bestehende



CSS-Definitionen (Dateien mit der Endung ".cssj" im Unterverzeichnis "css") werden vom Knowledge-Portal automatisch in einer Datei namens "upgraded.less" umgewandelt, die im Unterverzeichnis "less" abgelegt wird. Innerhalb der Stylesheet-Definitionen kann man nun in URLs die LESS-Variable "@{themePath}" verwenden, um auf das Verzeichnis des aktiven Themes zu verweisen (in früheren KP-Versionen musste der Platzhalter "\${themeBasePath}" verwendet werden). Beispiel:

```
.threeColumnBody .column1 {  
    background-image: url("@{themePath}/gfx/hintergrund.png");  
}
```

- Bei bestehenden Knowledge-Portal-Themes wird automatisch eine Datei namens "themeSettings.less" im LESS-Unterverzeichnis angelegt, in der alle vom Knowledge-Portal unterstützten LESS-Variablen mit den jeweiligen Standardwerten und einer kurzen Erläuterung eingetragen werden. Wenn innerhalb des KP-Themes auch ein JQuery-UI-Theme hinterlegt ist (Unterverzeichnis "jqueryui"), so werden die im JQuery-UI-Theme definierten Farbwerte und Schriftarten auf die dazu passenden Variablen in der Datei "themeSettings.less" übertragen.
- Das JQuery-UI-Theme, welches innerhalb eines Knowledge-Portal-Themes im Unterverzeichnis "jqueryui" abgelegt werden musste, wird im KP 4.0 nicht mehr berücksichtigt (siehe oben).

15.1.3 Performance-Verbesserungen

Das Laden der KP-Konfiguration aus dem Wissensnetz wurde erheblich beschleunigt, so dass sich die Zeit, die zum Testen von Konfigurationsänderungen aufgewendet werden muss, erheblich reduziert.

Die Ermittlung der auf Detailseiten anzuzeigenden Boxen geht nun deutlich schneller vonstatten als bei früheren KP-Versionen. Dadurch sinkt die Seitenladezeit von Detailseiten.

15.2 Referenz

15.2.1 Bereitgestellte LESS-Variablen

```
@page_margin_left: 5px;  
@page_margin_right: 5px;  
@page_column_spacing: 10px;  
@page_column1_width: 230px;  
@page_column3_width: 208px;  
@page_minwidth: 964px;  
@page_header_height: 80px;  
@page_header_spacing: 5px;  
  
@themeBgColor: white;  
@themeBgColorLight: darken(@themeBgColor, 15%);  
@themeHighlightColor: #003061;
```



```
@themeBgHeaderColor: @themeHighlightColor;

@themeHeaderTextColor: contrast(@themeBgHeaderColor);

@themeTextColor: black;
@themeLinkColor: @themeTextColor;

@themeErrorColor: red;
@themeWarningColor: orange;
@themeSuccessColor: green;
@themeSeparatorColor: darken(@themeBgColor, @borderDarken);

@borderDarken: 25%;

@ui_fsDefault: 15px;
@ui_ffDefault: Arial, Verdana, sans-serif;

@ui_iconColorHeader: @ui_fcHeader;
@ui_iconColorContent: @ui_fcContent;
@ui_iconColorDefault: @ui_fcDefault;
@ui_iconColorHover: @ui_fcHover;
@ui_iconColorActive: @ui_fcActive;
@ui_iconColorHighlight: @ui_fcHighlight;
@ui_iconColorError: @ui_fcError;

@ui_bgColorShadow: darken(@themeBgColor, 30%);
@ui_bgColorOverlay: fade(darken(@themeBgColor, 90%), 30%);
@ui_cornerRadius: 4px;

@ui_bgColorError: @themeErrorColor;
@ui_fcError: contrast(@ui_bgColorError, lighten(@themeErrorColor, 80%));
@ui_borderColorError: darken(@ui_bgColorError, @borderDarken);

@ui_bgColorHighlight: @themeBgHeaderColor;
@ui_fcHighlight: contrast(@ui_bgColorHighlight, lighten(@ui_bgColorHighlight, 80%));
@ui_borderColorHighlight: darken(@ui_bgColorHighlight, @borderDarken);

@ui_bgColorActive: @themeBgHeaderColor;
@ui_fcActive: @themeHeaderTextColor;
@ui_borderColorActive: darker(@ui_bgColorActive, @borderDarken);

@ui_bgColorHover: @ui_bgColorHighlight;
@ui_fcHover: contrast(@ui_bgColorHover);
@ui_borderColorHover: @ui_fcHover;

@ui_bgColorDefault: @themeBgColor;
@ui_fcDefault: contrast(@ui_bgColorDefault);
```



```
@ui_borderColorDefault: darken(@ui_bgColorDefault, @borderDarken);
```

```
@ui_bgColorHeader: @themeBgHeaderColor;  
@ui_fcHeader: @themeHeaderTextColor;  
@ui_borderColorHeader: @ui_bgColorHeader;  
@ui_fwHeader: bold;  
@ui_fsHeader: @ui_fsDefault * 0.9;
```

```
@ui_fcContent: @themeTextColor; // documented  
@ui_bgColorContent: @themeBgColor; // documented  
@ui_borderColorContent: darken(@themeBgColor, @borderDarken);
```

```
@ui_bgTextureActive: flat;  
@ui_bgTextureContent: flat;  
@ui_bgTextureDefault: flat;  
@ui_bgTextureError: flat;  
@ui_bgTextureHeader: flat;  
@ui_bgTextureHover: flat;  
@ui_bgTextureOverlay: flat;  
@ui_bgTextureHighlight: flat;  
@ui_bgTextureShadow: flat;
```

```
@ui_bgImgOpacityActive: 100;  
@ui_bgImgOpacityContent: 100;  
@ui_bgImgOpacityDefault: 100;  
@ui_bgImgOpacityError: 100;  
@ui_bgImgOpacityHeader: 100;  
@ui_bgImgOpacityHover: 100;  
@ui_bgImgOpacityOverlay: 100;  
@ui_bgImgOpacityHighlight: 100;  
@ui_bgImgOpacityShadow: 100;
```

16 Knowledge-Portal 5.0 - Versionshinweise

16.1 Was ist neu?

16.1.1 Änderungen beim BLOB-Streaming

Für den Up- und Download von Dateien verwendet das Knowledge-Portal 5.0 im Gegensatz zur Vorgängerversion keine KEM-Streaming-Bridges mehr, sondern REST-Bridges. Da man für viele Funktionen wie bspw. den Excel-Export auch früher schon REST-Bridges konfigurieren musste, war dies ein logischer Schritt zur Vereinfachung der Konfiguration.

Die REST-Bridges können genau wie Streaming-Bridges in der Datei "kem.properties" unter dem Konfigurationsschlüssel "bridges" eingetragen werden (bspw. "///localhost:8815/rest" anstelle von "///localhost:4714/streaming"). Der separate Konfigurationsschlüssel "restBridges", unter dem die REST-Bridges noch im KP 4 eingetragen werden mussten, wird im KP 5 nicht mehr verwendet.

Das KP 5 kann REST-Bridge nun auch über eine KLoadBalancer-Bridge verwenden ("bridges=///localhost/kloadba



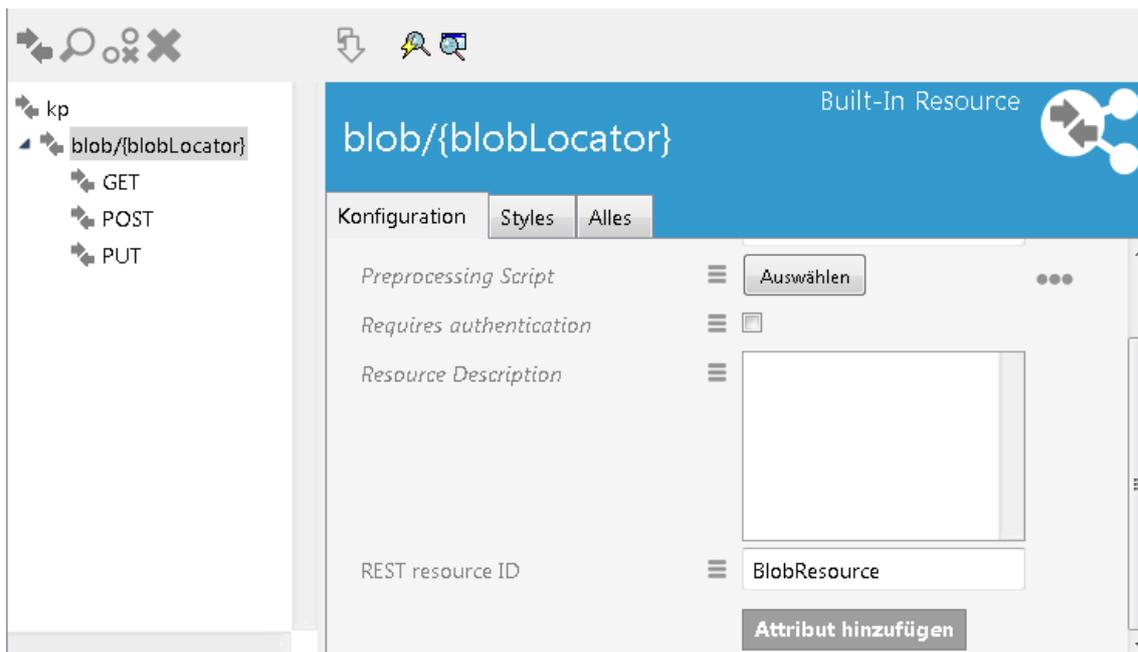
Dazu muss in der KLoadbalancerKonfiguration eine Gruppe namens "REST" für die REST-Bridges konfiguriert sein. Beispiel:

```
[KLoadBalancer]
configNames=KEM,REST
...

[REST]
bridgeClientClassName=KWeb.KHTTPRestBridge
inifile=bridge-rest.ini
bridgeLogfile=rest-<id>.log
maxBridges=2
ports=8815-8816
```

In der REST-Bridge-Konfigurationsdatei (im Beispiel "bridge-rest.ini") darf wie üblich kein Port konfiguriert werden, da dieser von der Load-Balancer-Bridge beim Start gesetzt wird. Die STREAMING-Gruppe in der KLoadBalancer-Konfiguration wird nicht mehr verwendet und kann deshalb ausgetragen werden.

Damit der Upload und Download über REST im KP funktioniert, muss mit dem Admin-Tool die REST-Komponente im Wissensnetz eingerichtet und mit dem Knowledge-Builder ein REST-Servicedefinition namens "kp" angelegt werden:



Es muss ein Request mit dem Pfad "blob/{blobLocator}" und den Methoden "GET", "PUT" und "POST" angelegt werden. Wichtig ist, dass bei dem Request das Attribut "REST resource ID" auf den Wert "BlobResource" gesetzt wird. Parameter müssen an den Methoden nicht angegeben werden.

17 Tomcat-Konfiguration

Da der IIS den Request an den Tomcat weiterleitet, müssen die Pfade zu den einzelnen Dateien auf beiden Servern übereinstimmen. Bei der Benutzung des Root-Kontextes führt dies zu Problemen, da die Webinhalte im Verzeichnis /ROOT/ (Aus IIS-Sicht) auf dem Tomcat-Server unter dem Pfad "/" zu erreichen sind.



Um dieses Problem zu lösen muss das Basisverzeichnis der IIS-Website nicht auf das webapps-Verzeichnis, sondern direkt auf die Webapplikation im Root-Kontext zeigen. Das Basisverzeichnis wäre dann:

```
C:\path\to\tomcat\webapps\ROOT\
```

worker.properties (Beispielkonfiguration)

```
# workers.properties
#
# This file provides minimal jk configuration properties needed to
# connect to Tomcat.
#
# The workers that jk should create and work with
#
worker.list = worker-kp-root, worker-kp-wb

#
# Define a worker for webapp kp-root
# Note that the name and the type do not have to match.
#
worker.worker-kp-root.type=ajp13
worker.worker-kp-root.host=localhost
worker.worker-kp-root.port=8009

# Define a worker for webapp kp-wb
#
worker.worker-kp-wb.type=ajp13
worker.worker-kp-wb.host=localhost
worker.worker-kp-wb.port=8009
```

17.1 Anforderungen

17.1.1 Version

Das Knowledgeportal ab der Version 2.5 erfordert mindestens eine Tomcat der Version 5.5.25 oder höher.

Tipp: Wird eine ältere Version verwendet, kann es beim Anzeigen von schwergewichtigeren Detailseiten zu JSP-Fehlern kommen. Ist dies der Fall sollte die Version des Tomcat überprüft werden.

17.2 Encoding

Um möglichst viele Sprachen bzw. Zeichensätze zu unterstützen, verwendet das Knowledge-Portal grundsätzlich ein UTF-8 Encoding. Entsprechend müssen Strings, die als Parameter in URLs übergeben werden, korrekt codiert werden.

Der Tomcat wird entsprechend beim Connector konfiguriert. Setzen Sie dazu den Parameter *URIEncoding* beim *http-Connector* auf *utf-8*.

Die Einstellung wird in der Datei *tomcat_verzeichnis/conf/server.xml* vorgenommen. Dort finden Sie einen Eintrag "Connector" der für die HTTP-Schnittstelle zuständig ist. Dieser hat meist Port 8080 oder 80 konfiguriert. Bei diesem Connector muss der Parameter **URIEncoding** auf den Wert "**utf-8**" gesetzt werden.



Der Eintrag sieht dann ähnlich wie folgender aus (andere Parameter können abweichen):

```
<Connector port="8080" maxHttpHeaderSize="8192" maxThreads="150" minSpareThreads="25"
maxSpareThreads="75" enableLookups="false" redirectPort="8443" acceptCount="100" connec-
tionTimeout="20000" disableUploadTimeout="true" URIEncoding="utf-8" />
```

17.3 Tomcat Connector für IIS

17.3.1 Tomcat Konfiguration

Tomcat bietet die Möglichkeit über verschiedenen Connectoren auf die Webapplikationen zuzugreifen. Im Normalfall wird der HTTP/1.1-Connector verwendet, der den Zugriff über HTTP im Browser ermöglicht.

Die Kommunikation zwischen IIS und Tomcat findet aus Performancegründen über AJP/1.3 (ApacheJServ Protocol) statt. Daher muss der AJP/1.3-Connector im Tomcat konfiguriert sein.

Dieser Connector ist nach der Installation von Tomcat bereits aktiviert. Daher sollten keine Anpassungen nötig sein, werden jedoch hier zur Kontrolle nochmals aufgeführt.

Zum Aktivieren des Connectors muss folgende Zeile in die Datei *tomcat_base_dir/conf/server.xml* eingetragen werden:

```
<Connector port="8009" enableLookups="false" redirectPort="8443" protocol="AJP/1.3" URIEnconding="UTF-
8" />
```

Wird der Connector beim Starten des Tomcat-Servers erfolgreich geladen sollte folgender Logeintrag erscheinen (Log-Level INFO):

```
08.01.2008 15:39:53 org.apache.coyote.ajp.AjpAprProtocol init
INFO: Initializing Coyote AJP/1.3 on ajp-8009
```

17.3.2 IIS-Website erstellen

Eine Webapplikation kann auf zwei unterschiedliche Arten in das Dokument-Root des Tomcats bzw. IIS integriert werden. Unterschied dabei ist, über welchen Pfad auf die Webapplikation zugegriffen werden kann.

Eine Webapplikation im Root-Kontext ist direkt über den Hostnamen erreichbar (z.B. <http://www.example.org/>). Die Angabe eines Pfades ist nicht nötig. Dieser Root-Kontext zeigt per default (andere Konfiguration im Tomcat möglich) auf die Webapplikation mit dem Namen "ROOT".

Ist eine Webapplikation nicht im Root-Kontext, ist beim Aufruf der Name der Webapplikation als Pfad anzugeben. Die WebApp "kp" wäre dann unter dem Pfad <http://www.example.org/kp/> erreichbar.

Je nachdem, in welchem Kontext die Webapplikation erreichbar sein soll, unterscheidet sich die Konfiguration des IIS geringfügig. Die Unterschiede werden im Abschnitt "IIS-Website für Root-Kontext" beschrieben.

Anlegen der Website

Die Website wird normal im IIS-Manager über "New > Website" angelegt. Das Basisverzeichnis zeigt dabei auf das webapps-Verzeichnis des Tomcat, also

```
c:\path\to\tomcat\webapps\
```

Wenn alle Anfragen, also auch auf statische Ressourcen wie Bilder, vom Tomcat bearbeitet werden sollen, kann hier auch ein leeres Verzeichnis angegeben werden. Dies muss dann



jedoch bei der Konfiguration des *isapi_redirect*-Filters beachtet werden.

Im Regelfall werden, zur Verbesserung der Performance, Anfragen auf statische Inhalte vom IIS bearbeitet und ausgeliefert.

Die Konfiguration der Rechte spielt erst beim einbinden des *isapi_redirect*-Filters eine Rolle und können hier frei vergeben werden, es wird jedoch mindestens das Leserecht benötigt.

17.3.2.1 IIS-Website für Root-Kontext

Da der IIS den Request an den Tomcat weiterleitet, müssen die Pfade zu den einzelnen Dateien auf beiden Servern übereinstimmen. Bei der Benutzung des Root-Kontextes führt dies zu Problemen, da die Webinhalte im Verzeichnis */ROOT/* (Aus IIS-Sicht) auf dem Tomcat-Server unter dem Pfad *"/* zu erreichen sind.

Um dieses Problem zu lösen muss das Basisverzeichnis der IIS-Website nicht auf das *webapps*-Verzeichnis, sondern direkt auf die Webapplikation im Root-Kontext zeigen. Das Basisverzeichnis wäre dann:

```
C:\path\to\tomcat\webapps\ROOT\
```

Integration weiterer Webapplikationen

Da das Basisverzeichnis nun direkt auf eine Webapplikation zeigt, können andere Webapplikationen nicht mehr erreicht werden. Um diese zu integrieren, müssen *Virtuelle Verzeichnisse* erstellt werden, die auf das Verzeichnis der jeweiligen Webapplikation zeigt. Dabei ist darauf zu achten, dass der Alias des virtuellen Verzeichnisses mit dem Kontext der Webapplikation, als dem entsprechenden Unterverzeichnis im *webapps*-Verzeichnis, übereinstimmt.

Um die Webapplikation "kp2" zu integrieren muss also das Virtuelle Verzeichnis "kp2" im Basisverzeichnis der IIS-Website erstellt werden. Dieses zeigt dann auf den Pfad

```
c:\path\to\tomcat\webapps\kp2\
```

Die Webapplikation ist somit auf dem IIS und dem Tomcat unter der URL <http://www.example.org/kp2/> erreichbar.

17.3.3 Tomcat-Connector Konfigurieren

Die Konfigurationsanweisungen basieren auf dem IIS-Webserver-HowTo von der Tomcat Website.

http://tomcat.apache.org/connectors-doc/webserver_howto/iis.html

17.3.3.1 Download

Der IIS-Connector kann von der Apache-Download-Site heruntergeladen werden. Für die Integration in den IIS wird die Datei ***isapi_redirect.dll*** benötigt.

Die Konnektoren sind über den Link <http://www.apache.org/dist/tomcat/tomcat-connectors/jk/binaries/> erreichbar. Aktuelle Version zum Zeitpunkt der Erstellung dieser Dokumentation ist 1.2.26.

Speichern Sie die heruntergeladenen Datei *isapi_redirect.dll* im Verzeichnis ***c:\path\to\tomcat\isapi_redirect***. Ort und Name des Verzeichnisses spielt keine Rolle. Wenn Sie einen anderes Verzeichnis wählen, beachten Sie dies bitte in den folgenden Schritten.



17.3.3.2 Konfigurationsdateien

Die folgenden Konfigurationsdateien können bis auf *isapi_redirect.properties* an einem beliebigen Ort abgelegt werden, dies muss jedoch in der Datei *isapi_redirect.properties* berücksichtigt werden.

Die Anweisungen gehen davon aus, dass alle Konfigurationsdateien im Verzeichnis *c:\path\to\tomcat\isapi_redir* abgelegt werden.

isapi_redirect.properties

Erstellen Sie die Datei ***isapi_redirect.properties*** im gleichen Verzeichnis in dem Sie die DLL abgelegt haben, im Beispiel also *c:\path\to\tomcat\isapi_redirect*. Die Konfigurationsdatei muss dabei den gleichen Namen haben wie die DLL-Datei. Lediglich der Suffix muss anstatt *.dll* *.properties* lauten. Nennen Sie die DLL-Datei um, muss auch die Properties-Datei umbenannt werden.

Eine Liste der möglichen Konfigurationsoptionen kann dem Apache Tomcat Connector - Reference Guide oder dem folgenden Beispiel entnommen werden.

Reference Guide: <http://tomcat.apache.org/tomcat-4.1-doc/jk/reference/printer/iis.html>

isapi_redirect.properties (Beispielkonfiguration)

```
# Configuration file for the Jakarta ISAPI Redirector
# The path to the ISAPI Redirector Extension, relative to the website
# This must be in a virtual directory with execute privileges
extension_uri=/jakarta/isapi_redirect.dll

# Full path to the log file for the ISAPI Redirector
log_file=c:\path\to\tomcat\isapi_redirect\isapi_redirect.log

# Log level (debug, info, warn, error or trace)
log_level=info

# Full path to the workers.properties file
worker_file=c:\path\to\tomcat\isapi_redirect\workers.properties

# Full path to the uriworkermap.properties file
worker_mount_file=c:\path\to\tomcat\isapi_redirect\uriworkermap.properties

# Full path to the rewrites.properties file
rewrite_rule_file=c:\path\to\tomcat\isapi_redirect\rewrites.properties
```

workers.properties

In dieser Konfigurationsdatei werden die verschiedenen Worker-Threads erzeugt und konfiguriert. Diese benötigen Tomcat-Serveradresse, Protokoll, ...

Es ist empfehlenswert für jede Webapplikation einen eigenen Workerthread zu konfigurieren. Das Erzeugen von Worker-Threads geschieht über *worker.list* (Value besteht aus kommaseparierter Liste).

Die Konfiguration der einzelnen Workerthreads wird dann über den entsprechenden Schlüssel *worker.worker_name.eigenschaft = Wert* vorgenommen

Es ist auch möglich ein Load-Balancing mit mehreren Worker-Threads zu konfigurieren. Lesen



Sie dazu bitte die offizielle Dokumentation unter:
<http://tomcat.apache.org/connectors-doc/reference/workers.html>

worker.properties (Beispielkonfiguration)

```
# workers.properties
#
# This file provides minimal jk configuration properties needed to
# connect to Tomcat.
#
# The workers that jk should create and work with
#
worker.list = worker-kp-root, worker-kp-wb

#
# Define a worker for webapp kp-root
# Note that the name and the type do not have to match.
#
worker.worker-kp-root.type=ajp13
worker.worker-kp-root.host=localhost
worker.worker-kp-root.port=8009

# Define a worker for webapp kp-wb
#
worker.worker-kp-wb.type=ajp13
worker.worker-kp-wb.host=localhost
worker.worker-kp-wb.port=8009
```

uriworkermap.properties

Diese Konfigurationsdatei definiert welche URLs von welchem Workerthread bearbeitet werden. Weiterführende Informationen zur Syntax finden Sie unter <http://tomcat.apache.org/connectors-doc/reference/uriworkermap.html>

Hinweis: Der Pfad zu einer Webapplikation ohne die Angabe der Indexdatei muss auch im Mapping vorhanden sein. Ist dies nicht der Fall, wird die URL vom Filter nicht erkannt und vom IIS bearbeitet. Dieser sucht dann nach der entsprechenden *Standardinhaltsseite*. Entspricht diese einer Datei die an den Tomcat weitergeleitet werden soll, kommt es zu einem Fehler, da der Filter nicht erneut angewendet wird!

uriworkermap.properties (Beispielkonfiguration)

```
# Webapplikation kp-root die sich im Context "kp-root" befindet
#
#/kp-root/*.skat = worker-kp-root
#/kp-root/*.do = worker-kp-root
#/kp-root/*.cssj = worker-kp-root
#/kp-root/ = worker-kp-root
#/kp-root = worker-kp-root

# Webapplikation kp-root die sich im Root-Context befindet
#
/*.*skat = worker-kp-root
/*.*do = worker-kp-root
/*.*cssj = worker-kp-root
/*.* = worker-kp-root
```



```
# Webapplikation kp-wb im Context "kp-wb"  
/kp-wb/*.skat = worker-kp-wb  
/kp-wb/*.do = worker-kp-wb  
/kp-wb/*.cssj = worker-kp-wb  
/kp-wb/ = worker-kp-wb  
/kp-wb = worker-kp-wb
```

rewrites.properties

Die Datei `rewrites.properties` beinhaltet einfache rewrites. Der Name dieser Konfigurationsdatei wird in der `isapi_redirect.properties` durch den Schlüssel `rewrite_rule_file` definiert. Hier können optional einfache rewrites definiert werden. Diese werden im Format `rewritten=real-url` definiert.

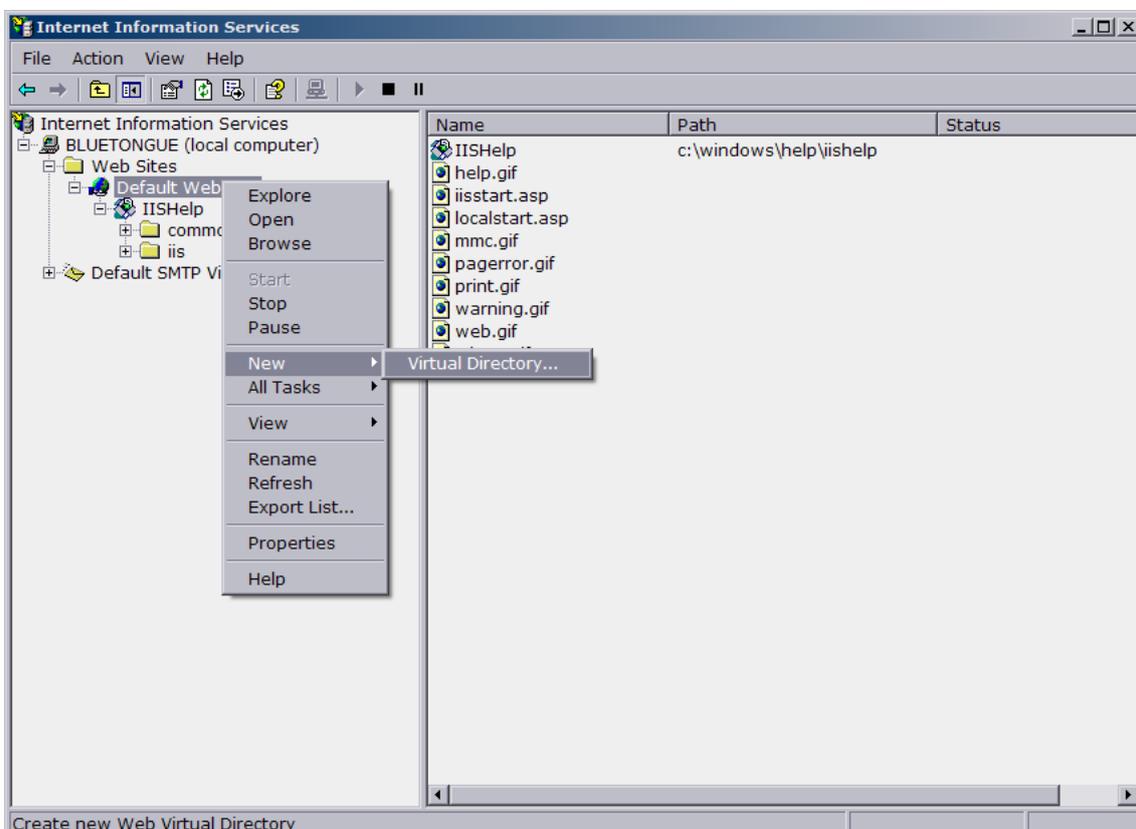
Bitte beachten Sie, dass in `uriworkermap.properties` den neuen Wert verwendet werden muss!

17.3.3.3 Plugin in IIS aktivieren

In diesem Abschnitt werden Schritte zum aktivieren des IIS-Plugins beschrieben. Diese Operationen werden all im IIS-Manager vorgenommen.

Virtuelles Directory erstellen

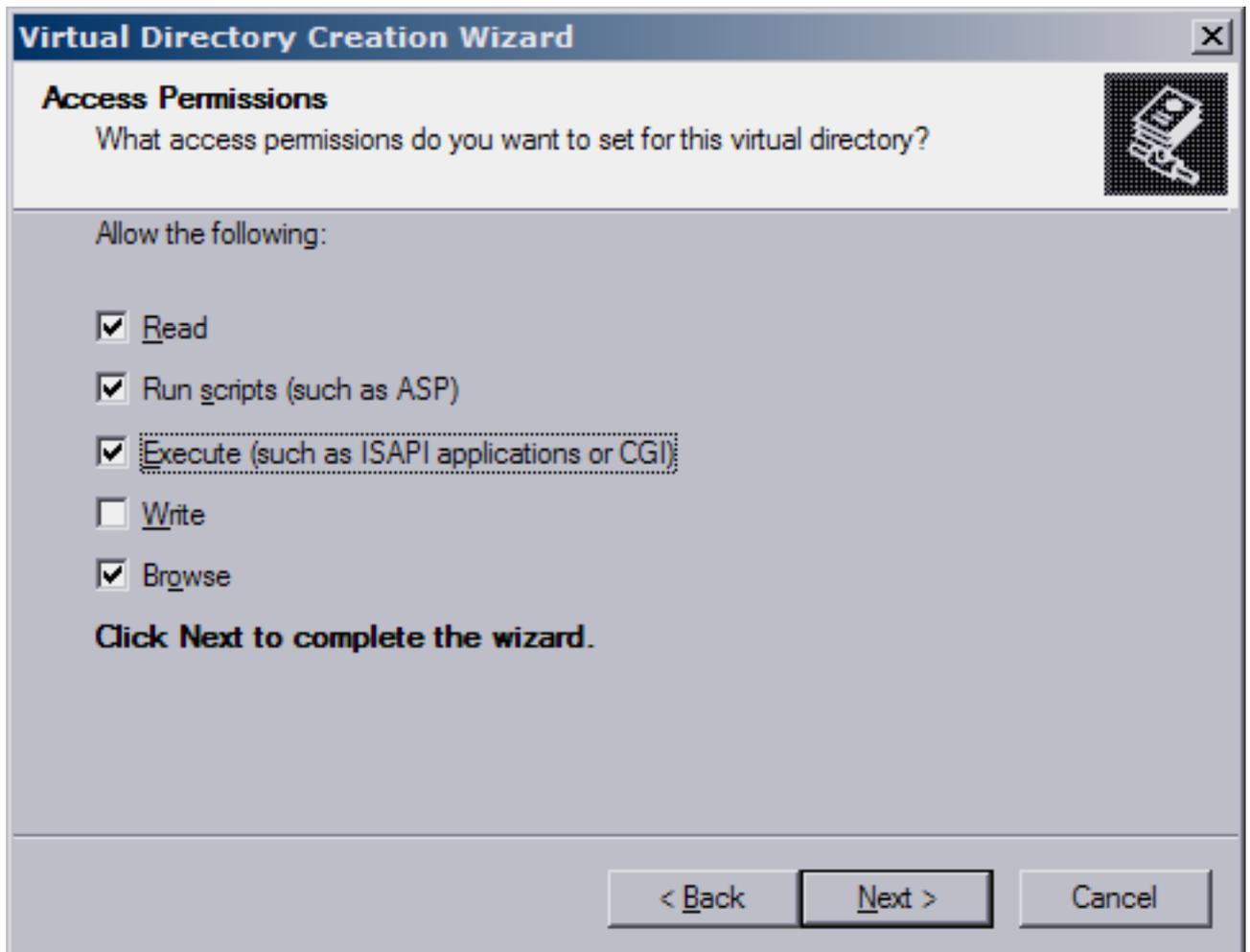
Zum Einbinden des Tomcat-connectors ind die IIS-Website muss im RootVerzeichnis der Website ein Virtuelles Directory erstellt werden. Dieses muss zwingend den Namen **jakarta** haben. Die Einstellung `extension_uri` in der Konfigurationsdatei `isapi_redirect.properties` muss ebenfalls mit `/jakarta/` beginnen. Über diese URL findet der Zugriff auf die DLL statt. Dieser Name ist Vorgeschieden und kann nicht geändert werden.





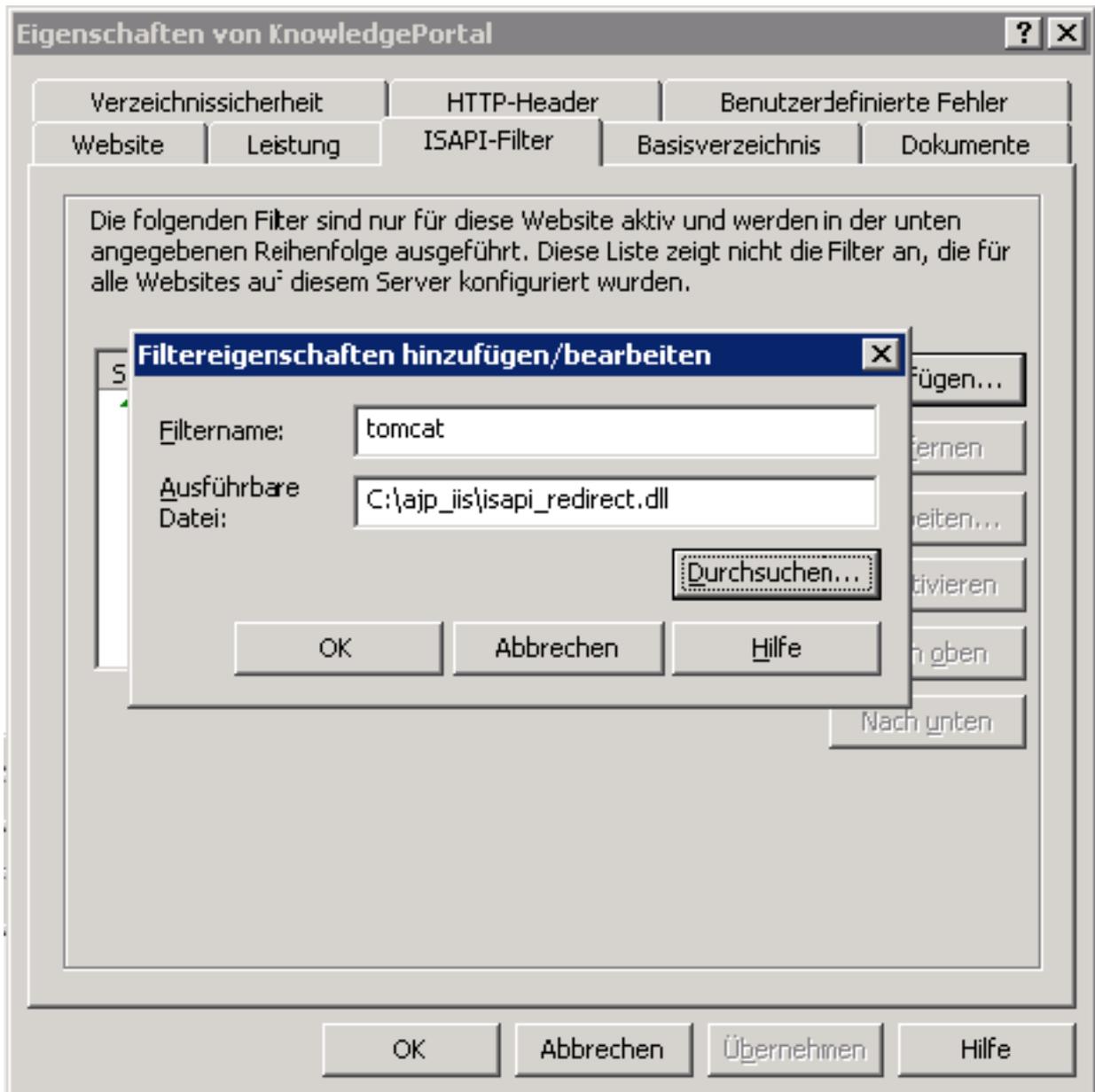
Der physische Pfad muss auf das Verzeichnis gesetzt werden, in dem sich die `isapi_redirect.dll` befindet. Im Konfigurationsbeispiel würde dieser Pfad `c:\path\to\tomcat\isapi_redirect\` entsprechen.

Dem Virtuellen Directory müssen beim erstellen Ausführrechte gegeben werden.



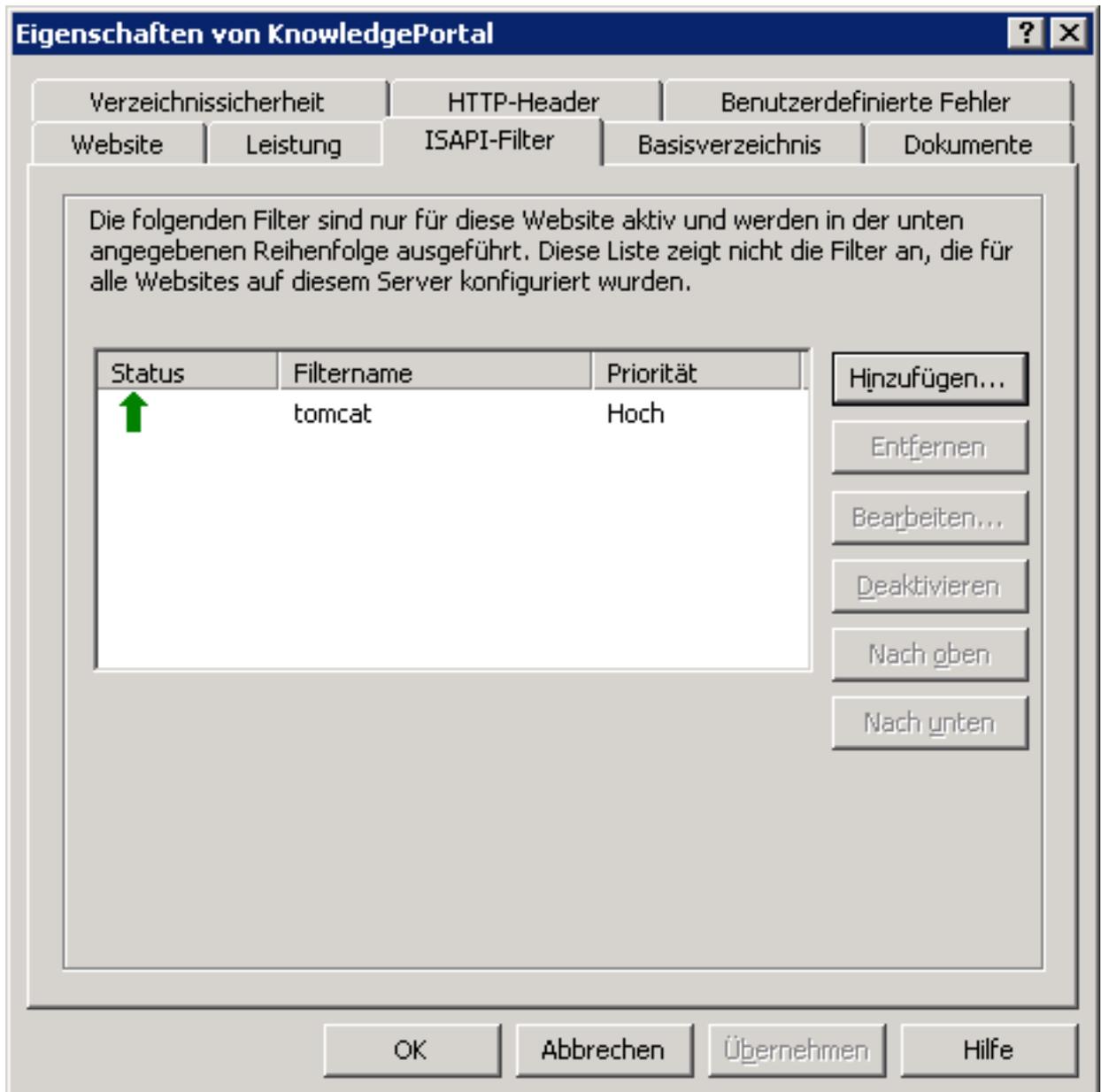
Filter hinzufügen

In den Eigenschaften der IIS-Website lassen sich unter dem Reiter *ISAPI-Filter* neue Filter anlegen. Fügen Sie hier einen neuen Filter hinzu. Der Name ist dabei frei wählbar, sollte jedoch die Aufgabe widerspiegeln. Als Executable muss die Datei `c:\path\to\tomcat\isapi_redirect\isapi_redirect.dll` gewählt werden.



Zum Aktivieren des Filters muss der IIS-Dienst neu gestartet werden. Es reicht nicht nur die IIS-Website neu zu starten. Wählen Sie dazu aus dem Kontextmenü des Servers die Option "Alle Aufgaben" und dann "IIS neu starten".

Nach dem Neustart des IIS sollte der Filter aktiviert sein. Dies wird bei den ISAPI-Filtern in der Spalte Status visualisiert. Hier sollte ein grüner Pfeil nach oben abgebildet sein.



IIS 6.0

Im IIS Version 6.0 muss der Filter zusätzlich als Webdienstenerweiterung hinzugefügt werden. Wählen Sie aus dem Kontextmenü des Ordner *Webdienstenerweiterungen* die Option *Neue Webdienstenerweiterung hinzufügen*. Der Erweiterungsname ist frei wählbar, stimmt jedoch idealerweise mit dem Namen des Filters überein. Als erforderliche Datei muss `c:\path\to\tomcat\isapi_redirect\isapi_re` gewählt werden. *Erweiterungsstatus auf Zugelassen setzen* muss aktiviert werden.

Die Webdienstenerweiterung wird zusätzlich benötigt und ersetzt nicht das hinzufügen des ISAPI-Filters!

17.3.4 Troubleshooting

Die folgenden Kapitel beinhalten Informationen falls die Integration des Tomcat-Connectors nicht auf anhieb klappt.



17.3.4.1 Unzulässige Funktion

Unter Umständen liefert der IIS den String "Unzulässige Funktion" an den Browser zurück. Dies ist der Fall, wenn der ISAPI-Filter kein Mapping auf die jeweiligen Worker-Threads findet.

Problem hierbei ist, dass der Filter beim Initialisieren die Existenz der Datei *uriworkermap.properties* nicht prüft. Die Debugausgabe zeigt zwar den konfigurierten Pfad zur Datei an, stellt jedoch nicht sicher, dass das Mapping auch gelesen wurde.

Tritt der Fehler auf, prüfen Sie bitte folgende Voraussetzungen

1. Der Parameter *worker_mount_file* in der Datei *isapi_redirect.properties* ist richtig gesetzt.
2. Der IIS-User besitzt Leserechte auf die Datei
3. Die Konfiguration in *uriworkermap.properties* ist korrekt