



FirstSpirit™

Your Content Integration Platform

FirstSpirit™ Release Notes

FirstSpirit™ Version 4.2 Release 4

Version	1.56
Status	RELEASED
Datum	2011-05-03
Abteilung	FS-Core
Autor/ Autoren	B. Gutknecht
Copyright	2011 e-Spirit AG
Dateiname	RELN424DE_FirstSpirit_Releasenotes

e-Spirit AG

Barcelonaweg 14
44269 Dortmund | Germany

T +49 231 . 286 61-30
F +49 231 . 286 61-59

info@e-Spirit.com
www.e-Spirit.com

e-Spirit^{AG}

Inhaltsverzeichnis

1	Einführung	6
2	Systemvoraussetzungen und Installation	7
2.1	Systemvoraussetzungen.....	7
2.1.1	Web-Browser (für die Verwendung von FirstSpirit WebEdit).....	7
2.1.2	Web- und Servlet-Engine / Application-Server.....	8
2.1.3	Datenbanken.....	8
2.1.4	Betriebssysteme.....	10
2.1.5	Java.....	13
2.2	Installation und Update.....	14
2.2.1	Update.....	14
2.2.2	Downgrades.....	16
3	Das FirstSpirit AppCenter	18
3.1	Native Applikationsintegrationen in 4.2R4.....	19
3.1.1	Voraussetzungen hinsichtlich 32-/64-Bit-Architekturen.....	20
3.1.2	Verwendung von AppCenter-Anwendungen.....	22
3.1.3	Web-Browser-Integration.....	22
3.1.4	Microsoft Office-Integration.....	23
3.1.5	Integration des Windows Media Players.....	24
3.2	Java-Applikationsintegrationen in 4.2R4 (Java Image Editor).....	24
3.3	Applikationsintegrationen im BETA-Stadium.....	25
3.4	Kundenindividuelle Applikationsintegrationen.....	26
3.5	Weiterführende Dokumentation.....	27



4	Neue/geänderte Funktionen für Redakteure	28
4.1	Neue/geänderte Funktionen in JavaClient und WebClient.....	28
4.1.1	Eingabekomponenten: Weiterentwicklung FS_LIST	28
4.1.2	Neue Eingabekomponente: FS_BUTTON.....	30
4.1.3	Referenznamen: Automatische Umwandlung von Sonderzeichen.....	31
4.2	Neue/geänderte Funktionen im JavaClient	34
4.2.1	Neue Icons für Dateien der Medien-Verwaltung	34
4.2.2	Funktionserweiterungen der Integrierten Vorschau	35
4.2.3	Bildbearbeitung im AppCenter.....	44
4.2.4	Datenquellen-Verwaltung: Verhalten von gefilterten Datensätzen	54
4.2.5	Überarbeitung von Versionsvergleich und Übersetzungshilfe	55
4.2.6	Versionshistorie: Neue Option "Teilarchivierte Revisionen einblenden"	58
4.2.7	Änderungen bei Arbeitsabläufen und Aufgaben	59
4.3	Neue/geänderte Funktionen im WebClient	61
4.3.1	Überarbeitete WebEdit Symbolleiste.....	61
4.3.2	Neue Quick-Edit-Leiste auf Seitenebene	62
4.3.3	Globale Suche	63
4.3.4	Statusleiste für Arbeitsabläufe.....	66
4.3.5	Überarbeitete Auswahldialoge.....	67
4.3.6	Überarbeitete "Bearbeiten"- und "Neu"-Dialoge	70
4.3.7	Überarbeiteter DOM-Editor.....	74
4.3.8	Datums-/Zeitauswahl mit Sekundenangabe über Kalenderansicht	76
4.3.9	Umstellung der WebEdit-Hilfe.....	76
5	Neue/geänderte Funktionen für Vorlagenentwickler	78



5.1	Eingabekomponenten: Migration zu FS_REFERENCE	80
5.1.1	Funktionsunterschiede FS_REFERENCE und entsprechende CMS_INPUT_-Eingabekomponenten	81
5.1.2	Umstellung von bestehenden Formularen.....	83
5.1.3	Ausnahmen	84
5.1.4	Umstellung der Ausgabekanäle	87
5.2	Eingabekomponenten: Migration zu FS_DATASET	90
5.2.1	Funktionsunterschiede FS_DATASET und CMS_INPUT_OBJECTCHOOSEER.....	91
5.2.2	Umstellung von bestehenden Formularen.....	91
5.2.3	Umstellung der Ausgabekanäle	92
5.3	Eingabekomponenten: Weiterentwicklung FS_LIST	93
5.3.1	Konfiguration.....	94
5.3.2	Umstellung von bestehenden Formularen.....	94
5.3.3	Umstellung der Ausgabekanäle	102
5.4	Neue Eingabekomponente FS_BUTTON.....	106
5.5	CMS_INCLUDE_OPTIONS: Auswahlmenge einschränken	107
5.6	Funktionserweiterungen der Integrierten Vorschau.....	108
5.7	Inline-Tabellen: CMS_INPUT_NUMBER in Stilvorlagen.....	109
5.8	Sichtbarkeit von Eingabekomponenten einschränken	109
5.9	Navigation: <code>parentpath</code> blendet Menüebenen bei Bedarf aus	111
5.10	Tagging.....	112
5.11	Aufruf von Klassen aus Modulen.....	113
5.12	Neue bzw. geänderte Tastaturkürzel	114
5.13	API-Erweiterungen	115
5.13.1	FirstSpirit Access-API	115
5.13.2	FirstSpirit Developer-API.....	116
5.13.3	FirstSpirit Application-API	117



6	Neue/geänderte Funktionen für Administratoren	119
6.1	Erstellen und Verwenden von JDBC-Treiber-Modulen	119
6.1.1	Erstellen eines JDBC-Treiber-Moduls	120
6.1.2	Installation und Konfiguration des JDBC-Treiber-Moduls	126
6.2	Funktionserweiterungen der Integrierten Vorschau	133
6.2.1	Integrierte Vorschau: Medien	134
6.2.2	Browser-Engine	136
6.2.3	Office-Engine	137
6.2.4	Graphik-Engine	138
6.3	Globale Web-Applikationen	139
6.3.1	Installieren von globalen Web-Applikationen	139
6.3.2	Konfiguration	140
6.3.3	Aktualisierung	141
6.3.4	Zugriff über die API	141
6.4	Performance und Sicherheit	142
6.5	Neuerungen im Lizenz-Management	144
6.5.1	license.OFFICE_INTEGRATION	144
6.5.2	license.APPTAB_SLOTS	145
6.6	NTLM-Authentifizierung und Windows 7	147
6.7	WebEdit-Themes	148
6.8	Überwachung des Festplattenplatzes	149
6.9	Referenznamen: Automatische Umwandlung von Sonderzeichen	153
6.9.1	Konfiguration	155
6.9.2	Symbole und äquivalente ASCII-Codierung (hexadezimal)	156
6.10	Referenznamen: Änderung projektweit unterbinden	157
6.11	Vereinfachte Encoding-Definition für Sprachen	159
6.12	Neue Funktionalitäten im Bereich Arbeitsabläufe und Aufgaben	160
6.12.1	Archivierung von geschlossenen Aufgaben	160



6.12.2	Cache-Speicher für Arbeitsabläufe und Aufgaben konfigurieren	161
6.12.3	Arbeitsablauf-E-Mails an Gruppen versenden.....	162
7	Neue/geänderte Funktionen in Modulen.....	164
7.1	FirstSpirit Portal SAP EP BP.....	164
7.1.1	Auswertung von Gruppen in CMS_INPUT_PERMISSION.....	164
7.1.2	Änderung des Skripts "Portal_createLink".....	165
7.2	FirstSpirit Personalisation.....	165
7.2.1	Benutzer-Anmeldung über Kerberos.....	165
7.3	FirstSpirit Search.....	166
7.4	FirstSpirit Paket-Verwaltung.....	166
8	Anhang.....	169
8.1	Umbenennung von FirstSpirit-Modulen.....	169
8.2	Änderungen im Softwareverhalten.....	171
8.3	Auslaufende Funktionen in FirstSpirit Version 4.2R4.....	182
8.4	Ankündigungen für FirstSpirit Version 5.0.....	183



1 Einführung

FirstSpirit™ Version 4.2 Release 4 (kurz: FirstSpirit V4.2R4) ist nach der Version 4.2 Release 2 die zweite Release-Version, die nach der initialen Freigabe von FirstSpirit Version 4.2 veröffentlicht wird (nähere Hinweise zum Release-Management siehe auch *FirstSpirit Release Notes* zur Version 4.2R2).

In diesem Dokument werden die in FirstSpirit V4.2R4 neu realisierten Funktionen vorgestellt. Dabei wird vorausgesetzt, dass der Leser FirstSpirit™ bereits kennt. Die Kapitel sind soweit es geht den verschiedenen Benutzergruppen zugeordnet und die neuen Funktionalitäten aus der Sicht der jeweiligen Gruppe beschrieben. Insbesondere zum Verständnis der Kapitel 5 bis 7 sind tiefergehende Kenntnisse im jeweiligen Bereich (Vorlagenentwicklung, Administration) erforderlich. Für Änderungen an hier beschriebenen Features und allgemeine Fehlerbehebungen siehe auch die Release-Notes, die zu neuen freigegebenen FirstSpirit-Build-Versionen veröffentlicht werden.

Eine wesentliche Rolle für die aktuelle Version spielt das neue "FirstSpirit AppCenter", eine Infrastruktur, mit der Fremdsoftware in die Redaktionsoberfläche des FirstSpirit-JavaClients eingebunden werden kann. Möglichkeiten und Beispiele werden recht ausführlich in Kapitel 3 ab Seite 18 vorgestellt.



2 Systemvoraussetzungen und Installation

Während sich die Installationsprozedur für FirstSpirit Version 4.2 Release 4 gegenüber V4.2R2 prinzipiell nicht geändert hat, haben sich hinsichtlich der Systemvoraussetzungen in V4.2R2 einige Änderungen ergeben.

2.1 Systemvoraussetzungen

Ziel in Version 4.2R4 war es, die von FirstSpirit unterstützten Web-Browser, Betriebssysteme und Datenbanken auf den aktuellen Stand zu bringen. Welche entsprechenden Produkte und Versionen gewartet bzw. nicht mehr gewartet werden, wird in den folgenden Kapiteln dargestellt.

Einige Versionen wurden "aus der Wartung" genommen wurden bzw. läuft für einige Versionen die Wartung aus. Dies bedeutet, dass ab FirstSpirit 4.2R4 nicht mehr getestet wird, ob diese Versionen noch korrekt mit neu implementierten Features in FirstSpirit funktionieren, und es besteht kein Anspruch mehr auf Fehlerbeseitigung für diese Versionen. Alle FirstSpirit Versionen 4.x bis zur Version 4.2R2 einschließlich sind jedoch mit diesen, aus der Wartung genommenen Versionen prinzipiell uneingeschränkt lauffähig. Auch für neue Funktionalitäten in Version 4.2R4 wird versucht, eine Lauffähigkeit mit aus der Wartung genommenen Konfigurationen zu ermöglichen, hierfür gibt es aber keine Gewährleistung. Mit Version 5.0 entfällt die Wartung für diese Versionen komplett.

Weitere Informationen zu den Systemvoraussetzungen für FirstSpirit Version 4.2R4 siehe aktuelles Technisches Datenblatt.



Zu den Systemvoraussetzungen hinsichtlich des FirstSpirit AppCenters siehe Kapitel 3 ab Seite 18.

2.1.1 Web-Browser (für die Verwendung von FirstSpirit WebEdit)

Folgende Web-Browser für die Arbeit mit FirstSpirit WebEdit sind ab FirstSpirit Version 4.2R4 in die Wartung aufgenommen worden bzw. werden nicht mehr gewartet:

- **Mozilla Firefox:** Die Versionen 3.5 und 3.6 wurden in die Wartung aufgenommen. Der Support für die Version 2.0 läuft hingegen aus.



- **Microsoft Internet Explorer:** Die Version 8 wurde neu in die Wartung aufgenommen, Version 7 wird weiterhin unterstützt. Die Wartung für die Version 6 läuft hingegen aus.
- **Andere Browser,** z. B. Safari oder Google Chrome, werden nicht offiziell unterstützt.



Zur Verwendung von Mozilla Firefox und Microsoft Internet Explorer für die Integrierte Vorschau im FirstSpirit-JavaClient siehe Kapitel 3.1.3 Seite 22.

2.1.2 Web- und Servlet-Engine / Application-Server

Für den Einsatz der FirstSpirit-Web-Applikationen wird eine Servlet-Engine benötigt, die die Servlet-API in der Version 2.4 implementiert.



Die Web-Applikationen der FirstSpirit-Module orientieren sich grundsätzlich an der Version 2.4, es sind aber modulspezifische Abweichungen möglich.

FirstSpirit unterstützt in der Version 4.2R4 die Zusammenarbeit mit folgenden Web-Servern:

- Apache 2
- Microsoft Internet Information Server (IIS) 6 / 7

2.1.3 Datenbanken

Folgende Datenbanken sind mit FirstSpirit Version 4.2R4 in die Wartung aufgenommen worden bzw. werden nicht mehr gewartet:

- **Apache Derby:** Die FirstSpirit-interne Datenbank wurde auf die Version 10.5.3.0 aktualisiert. Diese Apache Derby-Datenbank ist allerdings nicht für den Produktivbetrieb geeignet und sollte daher lediglich für Tests verwendet werden.





Achtung: Durch die Aktualisierung der Apache Derby-Datenbank ist es nicht möglich, Projekte, die diese Datenbank verwenden und mit FirstSpirit V4.2 Release 4 angelegt wurden, mit älteren FirstSpirit-Versionen zu verwenden (siehe auch Kapitel 2.2.2 Seite 16).

- **MySQL:** Die Wartung für MySQL in der Version 4.1 ist ausgelaufen, dafür wurde die Version 5.1 (jeweils Linux und Windows) neu in die Wartung aufgenommen. Die Version 5.0 wird weiterhin gewartet.
- **Microsoft SQL Server:** Die Wartung für den Microsoft SQL Server 2000 (Windows) ist ausgelaufen, dafür wurde der Microsoft SQL Server 2008 (Windows) neu in die Wartung aufgenommen. Die Version 2005 (Windows) wird weiterhin gewartet.



Für den Microsoft SQL Server 2008 gibt es keinen eigenen Datenbank-Layer, stattdessen muss der Layer für die Version 2005 ausgewählt werden.

- **Oracle:** Die Wartung für Oracle-Datenbanken in der Version 9 (Solaris und Linux) ist ausgelaufen, dafür wurde die Version 11 (Solaris, Linux, Windows) neu in die Wartung aufgenommen. Die Version 10 (Solaris, Linux, Windows) wird weiterhin gewartet.



Für Oracle sollte der JDBC-Treiber der Serie 10.1 (ojdbc14_10.1.0.x.jar) verwendet werden, da bei der Version 10.2 Probleme bei Verwendung des Datentyps LONG ab einer Länge von 4.000 Zeichen bei UTF-8-Kodierung vorkommen können. Alternativ muss bei Verwendung des 10.2-Treibers der Kompatibilitätsmodus für Oracle 9 LONG aktiviert werden, da LONG seit Oracle 9 deprecated ist. Dazu muss der Parameter

```
jdbc.property.oracle.jdbc.RetainV9LongBindBehavior=true
```

in der Datenbankkonfiguration ergänzt werden.

- **IBM DB2:** Die Wartung für IBM DB2 in der Version 8.2 (Linux) ist ausgelaufen, dagegen wurden die Versionen 9.5 und 9.7 (Linux, Windows) neu in die Wartung aufgenommen. Die Versionen 9.1 (Linux, AIX, Windows) und 8.2 (AIX) werden weiterhin gewartet.



- **PostgreSQL:** PostgreSQL in den Versionen 8.0 und 8.2 wird weiterhin gewartet, zusätzlich wurden die Versionen 8.3 und 8.4 (alle jeweils Linux und Windows) neu in die Wartung aufgenommen.

	⊕ neu in 4.2R4	○ beibehalten	⊖ Wartung ausgelaufen
MySQL	5.1	5.0	4.1
Microsoft SQL	2008	2005	2000
Oracle	11	10	9
IBM DB 2	9.7 9.5	9.1 8.2 (AIX)	8.2 (Linux)
PostgreSQL	8.4 8.3	8.2 8.0	



Ab Version 4.2R4 wird die Einbindung von JDBC-Treiberdateien als FirstSpirit Modul empfohlen. Sie sollten nicht mehr im Verzeichnis `.../shared/lib` abgelegt werden, u.a. um verschiedene Versionen eines Treibers nebeneinander betreiben zu können. Die Umstellung auf FirstSpirit-Module wird in Kapitel 6.1 ab Seite 119 erläutert.

2.1.4 Betriebssysteme

Folgende Betriebssysteme sind mit FirstSpirit Version 4.2R4 in die Wartung aufgenommen worden bzw. werden nicht mehr gewartet:

2.1.4.1 FirstSpirit-JavaClient

- **Microsoft Windows:** Microsoft Windows 7 wurde neu in die Wartung aufgenommen, der Support für Microsoft Windows XP läuft dagegen aus. Microsoft Vista wird weiterhin gewartet.
Der FirstSpirit-JavaClient kann sowohl auf 32- als auch auf 64-Bit-Plattformen betrieben werden. Da aber beim Einsatz der Integrierten Vorschau gegenwärtig noch native 32-Bit-Applikationen benötigt werden, ist beim Betrieb in 64-Bit-Umgebungen ggf. mit erhöhtem administrativen Aufwand zu rechnen.



- **Mac OS X / Linux:** Mac OS X (Version 10.6) und Linux werden ab der Version 4.2R4 offiziell unterstützt, allerdings mit Einschränkungen: Unter diesen beiden Betriebssystemen werden im Rahmen der Qualitätssicherung nur die FirstSpirit-Basisfunktionen regelmäßig getestet, der Schwerpunkt der Qualitätssicherung liegt aber weiterhin auf Microsoft Windows. Gerade unter Linux gibt es eine Vielzahl an Oberflächen (Window-Manager), für die ein vollständiger Funktionstest einen hohen Aufwand bedeutet. Daher kann es bei regelmäßiger Arbeit mit Mac OS X und Linux zu mehr betriebssystem-bedingten Problemen kommen, als unter Microsoft Windows, diese werden aber, so weit wie technisch möglich, im Rahmen der Softwarewartung beseitigt. Aufgrund von gravierenden Unterschieden zu Windows-basierten Systemen (speziell im Bereich der Oberflächenbibliotheken) gibt es darüber hinaus z. B. Einschränkungen bei Drag & Drop-Funktionen und hinsichtlich der Integration von nativen Anwendungen, z. B. in der Integrierten Vorschau, so dass die AppCenter-API auf Mac OS X und Linux nicht unterstützt wird. Neben der Mac OS-Version 10.6 kann prinzipiell auch die Version 10.5 verwendet werden, diese wird allerdings nicht offiziell von FirstSpirit unterstützt.

Sollen unter Linux nicht-lateinische Zeichen im FirstSpirit-JavaClient verwendet werden können (also z. B. chinesische Zeichen), muss dazu das Java-Paket (JRE) mit Sun-JVM der jeweiligen Linux-Distribution verwendet werden, z. B. die Pakete "sun-java6-jre", "sun-java6-plugin" oder "sun-java6-fonts" von Debian/GNU und Ubuntu oder "java-1_6_0-sun-1.6.0" und "java-1_6_0-sun-plugin-1.6.0" von SUSE und Red Hat. Der Betrieb des JavaClients unter anderen Java-Systemen wie OpenJDK, GNU Java oder Apache Harmony wird nicht unterstützt.



*Weitere Hinweise zu speziellen Systemvoraussetzungen für die Verwendung der **Integrierten Vorschau** sowie des FirstSpirit AppCenters siehe Kapitel 3 und Unterkapitel ab Seite 18.*





Wird Microsoft Windows 7 oder Vista eingesetzt, kann das NTLM-Loginmodul, über das die NTLM-Authentifizierung am FirstSpirit-Server vorgenommen wird, nur noch verwendet werden, wenn die Windows-7-Sicherheitseinstellungen herabgesetzt werden. Eine Anmeldung über Kerberos-Ticket (Integrierte Windows-Anmeldung) ist in Kombination mit Mozilla Firefox und Microsoft Internet Explorer ab V4.2R2 möglich und in diesem Kontext die zu bevorzugende Variante. Siehe dazu auch Kapitel 6.6 Seite 147. Hinweise zur Anmeldung über Kerberos-Ticket in Verbindung mit dem FirstSpirit-Modul "Personalisation" siehe Kapitel 7.2.1 Seite 165.

2.1.4.2 FirstSpirit-Server

- **Microsoft Windows:** Microsoft Windows 2008 Server, Microsoft Windows 2008 Server R2 und Microsoft Windows 2003 R2 wurden neu in die Wartung aufgenommen, der Support für Microsoft Windows 2000 und Microsoft Windows 2003 läuft hingegen aus.
- **Debian GNU/Linux:** Debian GNU/Linux in der Version 5.0 wird weiterhin gewartet, die Wartung für die Version 4.0 läuft hingegen aus.
- **Suse Linux:** Suse Linux Enterprise Server in der Version 11 wurde neu in die Wartung aufgenommen, Version 10 wird weiterhin gewartet, die Wartung für die Version 9 läuft hingegen aus.
- **Red Hat:** Red Hat Enterprise Linux in der Version 5 wird weiterhin gewartet, die Wartung für die Version 4 läuft hingegen aus.
- **Solaris:** Solaris wird weiterhin in den Versionen 9 und 10 gewartet.
- **IBM AIX:** IBM AIX in der Version 6.1 wurde neu in die Wartung aufgenommen, in der Version 5.3 wird IBM AIX weiterhin gewartet.





Wird der FirstSpirit-Server unter **Microsoft Windows 7 oder 2008 R2** betrieben, kann es zu Problemen kommen ("DbChecksumException"), wenn eine Berkley Datenbank in einer Version, die kleiner ist als 3.3.92, eingesetzt wird. Bis zur FirstSpirit Version 4.2R2 wurde die Version 3.3.87 eingesetzt, allerdings war dort Windows 2008 R2 nicht offiziell freigegeben. In der FirstSpirit Version 4.2R4 ist die Berkley Datenbank in der Version 3.3.96 enthalten, so dass 4.2R4 mit Windows 7 und Windows 2008 R2 kompatibel ist.

Microsoft Windows 2008 (ohne R2) und alle vorhergehenden Windows-Versionen sowie die Verwendung von Windows 7 oder Windows 2008 R2 für die FirstSpirit-Clients sind NICHT betroffen.

2.1.5 Java

2.1.5.1 Oracle (Sun) JDK

Mit der Version 4.2R4 wird der Support für das Oracle (vormals Sun) JDK 1.5 komplett eingestellt.

Alle Freigabetests für FirstSpirit Version 4.2R4 sind mit der JDK-Version 1.6.0_23 durchgeführt worden, sowohl für den FirstSpirit-Server in der 64-Bit-Version als auch für den FirstSpirit-JavaClient in der 32-Bit-Version. Daher wird für 4.2R4 der Einsatz dieser Version freigegeben und empfohlen.

Für einige Zwischenversionen von 4.2R4 sowie in 4.2R2 kann bei Verwendung des JDK 1.6.0_23 allerdings die Menüleiste des JavaClients fehlerhaft dargestellt sein: Die Menüs bzw. Menüpunkte sind "gedrängt" und haben keinen Zwischenraum. Diese Unschönheit ist mit Version 4.2.428 behoben, eine Rückportierung auf frühere Versionen ist nicht vorgesehen.

2.1.5.2 Apple JDK

Das Apple JDK, das jeweils aktuell im verwendeten Mac OS enthalten ist, ist für den FirstSpirit-JavaClient freigegeben. Für den FirstSpirit Server wird das Apple JDK nicht unterstützt.

2.1.5.3 IBM JDK

Während FirstSpirit Version 4.2R2 unter AIX mit dem IBM JDK der Version 1.6 inkompatibel ist, ist für FirstSpirit V4.2R4 das IBM JDK 1.6 für AIX freigegeben.



Dabei muss mindestens die Version 1.6.0 SR8 verwendet werden, da die IBM JDKs 1.6.0 SR6 und SR7 aufgrund eines Fehlers im JDK auch im Leerlauf 100% CPU-Auslastung verursachen. Für IBM JDK in der Version 1.5 unter AIX ist die Wartung hingegen ausgelaufen.



Das IBM JDK 1.6 ist mit FirstSpirit bis Version 4.2R2 einschließlich inkompatibel (sowohl unter AIX als auch unter Linux)!

2.2 Installation und Update

Zur Installation von V4.2R4 siehe *FirstSpirit Installationsanleitung Version 4.2, 4.2R2 und 4.2R4*.

2.2.1 Update

Ein Update von FirstSpirit Version 4.2 oder 4.2R2 auf FirstSpirit Version 4.2 Release 4 ist problemlos über das Austauschen der JAR-Datei (`fs-server.jar`) möglich (siehe *FirstSpirit Installationsanleitung Version 4.2 und 4.2R2*, Kapitel 4.9). Folgende Veränderungen müssen beim Update von FirstSpirit-Servern vorgenommen werden:

Bei einem Update von 4.2R2 auf 4.2R4 müssen auch alle installierten **FirstSpirit-Module** und alle **Web-Anwendungen** aktualisiert werden (siehe FirstSpirit Handbuch für Administratoren), da in 4.2R4 ein neuer Signierungsschlüssel verwendet wird und es sonst zu Fehlermeldungen kommen kann.



Mit Version 4.2R4 wurde der Button "Aktualisieren" in den Server-Eigenschaften, Bereich "Module" entfernt. Ab 4.2R4 werden Module über den Button "Installieren" aktualisiert.

Für die Verwendung von **JDBC-Treibern** im Datenbank-Kontext ist es für ein Upgrade auf Version 4.2R4 empfehlenswert, die Treiberdateien als FirstSpirit-Modul einzubinden, statt sie im Verzeichnis `.../shared/lib` abzulegen. Siehe dazu Kapitel 6.1 ab Seite 119.

Mit Version 4.2R4 wurde die Eingabekomponente **FS_LIST** erweitert. Bereits bestehende Eingabekomponenten vom Typ `CMS_INPUT_CONTENTAREALIST`, `CMS_INPUT_CONTENTLIST`, `CMS_INPUT_LINKLIST`, `CMS_INPUT_SECTIONLIST` und `CMS_INPUT_TABLIST` können durch eine Anpassung der Formulare nach



FS_LIST migriert werden. Ebenso müssen bereits vor R4 erstellte FS_LIST-Eingabekomponenten FS_LIST-Syntax umgestellt werden (siehe dazu Kapitel 5.3 Seite 93).



*Wurden Daten mit FS_LIST gespeichert, können diese aufgrund eines abweichenden Datenformats **nicht** von CMS_INPUT_CONTENTAREALIST usw. gelesen und interpretiert werden.*

Ab Version 4.2R4 werden Client-Applikationen (z. B. die Mozilla-Engine zur Verwendung der Integrierten Vorschau) abhängig von der Version standardmäßig in ein eigenes Verzeichnis auf dem Arbeitsplatzrechner ausgerollt, für 4.2R4 z. B. `\.firstspirit_4.2R4`. Das bis 4.2R2 einschließlich verwendete Verzeichnis `\.firstspirit` wird dabei nicht vom System gelöscht, sondern muss bei Bedarf (z. B. um Speicherplatz freizugeben) manuell entfernt werden. Siehe dazu auch Kapitel 8.2, Eintrag Rollout-Verzeichnis für Client-Applikationen.

Darüber hinaus wurden **inkompatible Änderungen an Datenformaten** in FirstSpirit vorgenommen. Daher werden bei einem Upgrade auf 4.2R4 automatisch folgende Konvertierungsschritte durchgeführt:

- **Konvertierung von gestarteten Arbeitsabläufen / Aufgaben:** In der Version 4.2R4 wird für Arbeitsabläufe ein performanteres Datenformat verwendet. Bestehende und gestartete Arbeitsabläufe in einem Projekt werden bei einem Upgrade auf 4.2R4 automatisch konvertiert. Dies geschieht beim ersten Start des auf die Version 4.2R4 aktualisierten FirstSpirit-Servers. Dabei werden alle Projekte auf dem Server nacheinander geöffnet und die gestarteten Arbeitsabläufe einzeln konvertiert. Dieser Vorgang kann einige Zeit in Anspruch nehmen. Der Server kann erst **nach** dem Abschluss der Konvertierung benutzt werden.
- **Neuberechnung und Konvertierung von Referenzen:** Auch für Referenzen wird in Version 4.2R4 ein performanteres Datenformat verwendet. Bestehende Referenzen werden bei einem Upgrade auf 4.2R4 automatisch konvertiert und Neuberechnet. Dies wird beim ersten Öffnen von Projekten, die sich auf einem auf die Version 4.2R4 aktualisierten FirstSpirit-Server befinden, vorgenommen. Im Gegensatz zur Arbeitsablauf-Konvertierung erfolgt die Referenz-Neuberechnung asynchron. D. h., Server und Projekte können bereits verwendet werden, während die Referenzberechnung noch läuft. Die Konvertierung des Datenformates der Referenzen erfolgt während der Referenz-Neuberechnung



2.2.2 Downgrades

Durch die Aktualisierung der FirstSpirit-internen **Apache Derby Datenbank** (siehe Kapitel 2.1.3 Seite 8) ist es nicht möglich, Projekte, die diese (nicht für den produktiven Einsatz freigegebene) Datenbank verwenden und mit FirstSpirit Version 4.2R4 angelegt wurden, mit einer älteren FirstSpirit-Version zu verwenden.

Aufgrund der in Kapitel 2.2.1 Seite 14 beschriebenen in Version 4.2R4 neu eingeführten **Datenformate für Arbeitsabläufe und Referenzen** ist ein Downgrade von 4.2R4 nur nach 4.2R2 ab Version 4.2.219 möglich, da erst ab diesem Softwarestand die neuen Datenformate interpretiert werden können. Dies betrifft auch Projekte, die mit einem FirstSpirit-Server der Version 4.2R4 erstellt und anschließend per Export/Import auf einem FirstSpirit-Server der Version 4.2R2 verwendet werden. Projekte, die eine Derby-Datenbank verwenden und in 4.2R4 angelegt wurden, können in keinem Fall mit einem 4.2R2-Server verwendet werden.



Ein Downgrade (oder auch Projektexporte) auf ältere Versionen als FirstSpirit 4.2 Release 2 ist NICHT möglich!

Auch bei einem Downgrade auf 4.2R2 müssen alle installierten **FirstSpirit-Module** und alle **Web-Anwendungen** in der entsprechenden Version verwendet werden (siehe *FirstSpirit Handbuch für Administratoren*), da in 4.2R4 ein neuer Signierungsschlüssel verwendet wird und es sonst zu Fehlermeldungen kommen kann.

Darüber hinaus können **Projekte mit Bildern**, die mit der Version 4.2R4 erstellt wurden, nicht auf Servern mit einer Version kleiner als 4.2.219 verwendet werden. Auf Servern mit einer Version unterhalb von 4.2.219 werden Bilder in der Medien-Verwaltung aufgrund eines fehlenden internen Attributs nicht angezeigt.

In 4.2R4 sind einige **neue Optionen** z. B. im Menü "Ansicht" des JavaClients hinzugekommen (siehe Kapitel 4.2.2.3 Seite 41). Wird ein Projekt, das mit Version 4.2R4 erstellt wurde, z. B. auf einem FirstSpirit-Server der Version 4.2R2 verwendet, kommt es zu Warnungen beim ersten Start des Servers, z. B.:

```
WARN 24.06.2010 12:22:08.063
(de.espirit.firstspirit.common.xml.XmlSerializer): no field
de.espirit.firstspirit.server.projectmanagement.InlinePreviewConstraintsD
TO#_mediaPlayerFileExtensions
```



Werden Änderungen in den Projekteigenschaften vorgenommen, werden beim nächsten Server-Start keine Warnungen mehr ausgegeben.



3 Das FirstSpirit AppCenter

e-Spirit sieht sich als Produkthersteller, der konsequent eine Best-of-Breed-Strategie verfolgt. Dazu gehört die systematische Fokussierung aller Eigenimplementierungen auf die Kernbestandteile des Produktes FirstSpirit und die bewusste Verlagerung von spezifischen Funktionen in Fremdprodukte der jeweiligen Marktführer. Die entscheidende Voraussetzung, um diesen populären Outsourcing-Gedanken in einem Softwareprodukt erfolgreich umsetzen zu können, ist dabei die "nahtlose Integration": es darf für den Endbenutzer keinen Bruch zwischen den eingesetzten Produkten geben! Die Benutzerführung muss für den Anwender voll integriert, nahtlos und optisch wie aus einem Guss erscheinen.

Dieser Gedanke der nahtlosen Integration von Fremdanwendungen in die FirstSpirit Redaktionsumgebung wird als "AppCenter" bezeichnet. Das FirstSpirit AppCenter stellt einen definierten Bereich innerhalb des Redaktionssystems zur Verfügung, in dem eigenständige Anwendungen ablaufen können, die nicht Bestandteil von FirstSpirit sind (sogenannte "AppCenter-Anwendungen").

Beispiele für AppCenter-Anwendungen sind die mit FirstSpirit Version 4.2R4 verfügbare Integration von Microsoft Office (siehe Kapitel 4.2.2.1 Seite 38 bzw. Kapitel 6.2.3 Seite 137) oder die Funktionen zur integrierten Bildbearbeitung (siehe Kapitel 4.2.2 Seite 35 bzw. Kapitel 4.2.3 Seite 44). Auch bei den bereits in den Versionen 4.2 R0 und R2 integrierten Web-Browsern Mozilla Firefox und Microsoft Internet Explorer handelt es sich um AppCenter-Anwendungen, sie werden als "Integrierte Vorschau" betitelt. All diese AppCenter-Anwendungen wurden von e-Spirit als Produktbestandteile realisiert. Es gibt aber auch eine Reihe von AppCenter-Anwendungen, die als FirstSpirit-Module realisiert werden. Diese AppCenter-Module können sowohl von e-Spirit selbst, aber auch von einem Partner entwickelt werden.

Folgende Beispiele, die im Rahmen des AppCenters bereits erfolgreich von e-Spirit integriert wurden, vermitteln einen Eindruck davon, welche Möglichkeiten das AppCenter über die aktuell als Produktbestandteile realisierten Anwendungen hinaus bietet: Mittels der Integration von Google Maps bzw. Google Earth kann im FirstSpirit-Redaktionssystem einfach und intuitiv mit Geo-Koordinaten gearbeitet werden, mittels der Integration des Online-Video-Angebots von MovingImage24 können Videos ausgewählt und per Mausklick in FirstSpirit eingebunden werden und mittels der Integration der Online-Bilddatenbanken Fotolia und Pixelio kann im FirstSpirit-Redaktionssystem direkt nach Bildmaterial recherchiert und die gewünschten Bilder, ebenfalls per Mausklick, nach FirstSpirit übernommen werden.



Technisch gesehen besteht das AppCenter aus einer Menge von Schnittstellen, die von e-Spirit für die Nutzung durch Partner freigegeben wurden, damit diese im Rahmen des AppCenters spezifische Anwendungen realisieren oder integrieren können (siehe auch Kapitel 3.5 Seite 27). Gegenwärtig beschränken sich die verfügbaren AppCenter-Schnittstellen auf die Infrastruktur, die für die Integration von Web-Anwendungen benötigt werden. Bereits realisiert, aber noch nicht öffentlich verfügbar sind entsprechende Schnittstellen für die Integration von nativen Anwendungen (analog zu Microsoft Office, siehe auch Kapitel 3.1 Seite 19) oder auch Java-Anwendungen (analog zur integrierten Java-Bildbearbeitung, Java Image Editor, siehe auch Kapitel 3.2 Seite 24). Die Realisierung von AppCenter-Anwendungen und deren Integration in FirstSpirit wird allgemein als "Applikationsintegration" bezeichnet.

Im Folgenden werden Voraussetzungen und Einschränkungen bei der Verwendung von AppCenter-Anwendungen dargestellt sowie auf weiterführende Dokumentation verwiesen.

3.1 Native Applikationsintegrationen in 4.2R4

Die mit Version 4.2R4 bereits als Produktbestandteil von FirstSpirit realisierten AppCenter-Integrationen bieten Redakteuren einen hohen Mehrwert für die redaktionelle Arbeit: Die Integration der Web-Browser Mozilla Firefox und Microsoft Internet Explorer erlaubt nicht nur eine direkte Vorschau im Client, sondern visualisiert auch den Zusammenhang zwischen den im Client eingegebenen Inhalten und ihrer Auswirkung bzw. Darstellung auf der Webseite und erleichtert damit die Navigation durch das Projekt. Mit den für diese Browser offiziell verfügbaren Plugins können auch PDF-, Flash-Dateien usw. in der integrierten Vorschau dargestellt werden. Mit den integrierten Microsoft Office- bzw. OpenOffice-Anwendungen können Microsoft Office Word-, Excel- und Powerpoint- bzw. OpenOffice Writer-, Calc und Impress-Dateien direkt im JavaClient bearbeitet und abgespeichert werden. Die OpenOffice-Integration befindet sich dabei aktuell in der BETA-Test-Phase und ist nicht offiziell freigegeben.

Um diese nativen Integrationen nutzen zu können, muss die jeweilige Anwendung auf dem Arbeitsplatzrechner installiert sein. Die Integrationen im JavaClient greifen dabei auf die lokale Konfiguration der Anwendungen zurück. Eine Ausnahme dabei bildet dabei der Mozilla Firefox: dieser wird mit dem FirstSpirit Kernprodukt ausgeliefert und von FirstSpirit selbst verwaltet.

Bei der Verwendung der oben genannten Integrationen im Client kann es dabei prinzipbedingt zu Einschränkungen kommen, z. B. weil einige der integrierten Anwendungen nicht vollständig mit allen Plattformen oder Bittigkeiten (32 oder 64



Bit, siehe dazu Kapitel 3.1.1 Seite 20) zusammenarbeiten. So kann z. B. unter Linux als Web-Browser nicht der Microsoft Internet Explorer, als Office-Anwendung nicht Microsoft Office verwendet werden.

3.1.1 Voraussetzungen hinsichtlich 32-/64-Bit-Architekturen

Bei der Verwendung von **32-Bit-Systemen** gibt es gegenwärtig die wenigsten Interoperabilitätsprobleme, da die verwendete Software-Infrastruktur (Betriebssystem, JDK und native Applikation) durchgängig in 32 Bit vorliegt. Durch die bekannten Limitierungen der 32-Bit-Architektur, speziell im Bereich des Hauptspeichers, schreitet die Einführung von 64-Bit-Systemen aber zügig voran – mit Microsoft Windows 7 auch auf dem Desktop. Im Rahmen der nativen Applikationsintegration ergibt sich daraus eine neue, prinzipbedingte Komplexitätsstufe, die im Folgenden kurz umrissen wird:

Grundsätzlich sind aktuell **64-Bit-Betriebssysteme** wie Windows 7 oder Linux aus Kompatibilitätsgründen auch in der Lage, 32-Bit-Anwendungen auszuführen. Damit ergibt sich das Problem, dass auf einem 64-Bit-Betriebssystem sowohl 64- als auch 32-Bit-Anwendungen vorhanden sein können. Dies gilt speziell auch für das JDK/JRE: dies kann auf einem 64-Bit-Betriebssystem in einer 32- und/oder einer 64-Bit-Variante vorliegen.

Der FirstSpirit-JavaClient selbst ist sowohl mit einem 32- als auch einem 64-Bit-Java kompatibel. Der Unterschied liegt in der Applikationsintegration des JavaClients: Auch auf einem 64-Bit-Betriebssystem können in Verbindung mit einem 32-Bit-Java (prinzipbedingt) nur 32-Bit-Applikationen integriert werden!

Auf einem 64-Bit-Betriebssystem ist aber prinzipiell der Einsatz eines 64-Bit-Java-Systems wünschenswert. Im FirstSpirit-JavaClient ist die Applikationsintegration so realisiert, dass **alle** integrierten nativen Applikationen ebenfalls als 64-Bit-Variante eingebunden werden, d. h. aus einem 64-Bit-Java heraus werden keine 32-Bit-Anwendungen gestartet! (Fehlermeldung: "Can't load IA 32-bit .dll on a AMD 64-bit platform")

Es müssen also alle zu integrierenden Anwendungen in einer 64-Bit-Version vorliegen, z. B. "Microsoft Internet Explorer (64 Bit)". Leider sind einige Anwendungen (speziell unter Microsoft Windows) gegenwärtig noch nicht in einer 64-Bit-Variante verfügbar (z. B. Mozilla Firefox und OpenOffice für Windows 64 Bit). Daher können diese Anwendungen in Verbindung mit einem 64-Bit-Java nicht genutzt werden. Ein Einsatz dieser Anwendungen ist nur möglich, wenn auf dem 64-Bit-Betriebssystem ein 32-Bit-Java verwendet wird!



Bei der Verwendung von 64-Bit-Web-Browsern wie z. B. dem Microsoft Internet Explorer, müssen auch alle im Browser verwendeten Plugins in einer 64-Bit-Variante vorliegen. Das Adobe Acrobat Reader-Plugin, zur Anzeige von PDFs, ist aktuell z. B. nicht als 64-Bit-Variante verfügbar. Als Alternative kann "PDFexchange" verwendet werden.

Mit FirstSpirit Version 4.2R4 wird sowohl für Windows als auch für Linux ein durchgängiger 64-Bit-Betrieb auch in Verbindung mit der Applikationsintegration möglich! Aufgrund der prinzipbedingten Komplexität ist der 64-Bit-Support für die Applikationsintegration aktuell aber noch experimentell, da noch nicht alle möglichen Kombinationen (Betriebssystem, Java, Fremdanwendung) vollständig auf Kompatibilität getestet wurden. Es ist geplant, den 64-Bit-Support im Verlauf von FirstSpirit Version 4.2R4 freizugeben. Sobald er freigegeben ist, wird dies auch entsprechend ins *Technische Datenblatt* zu FirstSpirit aufgenommen. Daher kann es in diesem Bereich aktuell noch Diskrepanzen zwischen den vorliegenden *Release Notes* und dem aktuellen *Technischen Datenblatt* geben.

Zu beachten ist weiterhin, dass die verfügbaren Kombinationsmöglichkeiten aufgrund der geschilderten Abhängigkeiten eingeschränkt sind (siehe Tabelle):

	Mozilla Firefox	Internet Explorer	Vorschau von PDF, SWF usw.	Microsoft Office-Anwendungen	OpenOffice-Anwendungen (BETA)
Windows 32 Bit	+	+	+	+	+
Windows 64 Bit	/ nur 32 Bit	+ 32 Bit und 64 Bit (BETA)	/ teilweise nur 32 Bit	+	/ nur 32 Bit
Mac OS 10.6	+ nur 32 Bit	-	+ nur 32 Bit	-	-
Linux 32 Bit	+	-	+	-	+
Linux 64 Bit	+ 32 Bit und 64 Bit (BETA)	-	/ teilweise nur 32 Bit	-	+ 32 Bit und 64 Bit (BETA)

Legende: (+) kompatibel (/) mit Einschränkungen (-) nicht kompatibel

Die Bit-Angaben beziehen sich jeweils auf die Anwendung **und** das JRE!



3.1.2 Verwendung von AppCenter-Anwendungen

Für die Verwendung von AppCenter-Anwendungen (Kapitel 3.1 bis 3.4 einschließlich) ist grundsätzlich zu beachten, dass FirstSpirit für die Applikationsintegration die erforderlichen Schnittstellen bereitstellt, aber in der Regel keinen Einfluss auf die integrierten Anwendungen selbst hat. Integrierte Fremdanwendungen sind kein Produktbestandteil von FirstSpirit. Das bedeutet u.a., dass die Verantwortung für die Funktionalität der integrierten Anwendungen beim Hersteller bzw. beim Kunden oder Partner liegt, der die Anwendung realisiert.

Probleme können im Rahmen des FirstSpirit-Produktsupports gemeldet werden und werden (nach Möglichkeit) beseitigt, wenn sie auf der Ebene der Integrations-Schnittstelle liegen. Ein Anspruch auf Fehlerbeseitigungen **innerhalb** der in den Kapiteln 3.1 bis 3.4 beschriebenen Fremdanwendungen gegenüber e-Spirit besteht aber **nicht**.

3.1.3 Web-Browser-Integration

Microsoft Internet Explorer und Mozilla Firefox werden je nach Einstellung im JavaClient oder in den Projekteinstellungen (Projekt-Administrator) zur Anzeige der HTML-Vorschau sowie von Medien aus der FirstSpirit Medien-Verwaltung vom Typ PDF und Adobe Flash verwendet.

Beide Browser werden gegenwärtig als 32-Bit-Applikationen gestartet. Für die Verwendung des Mozilla Firefox' wird eine spezielle in den JavaClient integrierte Version des Firefox' verwendet, so dass keine lokale Installation von Mozilla Firefox auf dem Arbeitsplatzrechner erforderlich ist. Soll der Microsoft Internet Explorer verwendet werden, muss eine Installation dieses Browsers in der 32-Bit-Variante auf dem Arbeitsplatzrechner verfügbar sein. Darüber hinaus muss zum gegenwärtigen Zeitpunkt das JRE immer eine 32-Bit-Anwendung sein, damit die integrierte Vorschau funktionsfähig ist. Eine Unterstützung für 64-Bit-JREs in Verbindung mit der Integrierten Vorschau ist geplant, erfordert aber einen stabilen 64-Bit Mozilla Core, der aktuell unter Windows noch nicht verfügbar ist.

Die Browser dienen der Anzeige von speziellen Dateiformaten, z. B. PDF, SWF usw. Die Unterstützung dieser Dateiformate ist plattformunabhängig, und damit prinzipiell auch unter Linux und Mac OS verfügbar, allerdings immer nur in der 32-Bit-Variante. Es kann jedoch nicht garantiert werden, dass sämtliche Browser-Plugins betriebssystem-weit korrekt mit der Integrierten Vorschau funktionieren. Plugins von bekannten Herstellern, wie z. B. Adobe Flash und Adobe Acrobat Reader, führen aber erfahrungsgemäß nicht zu Problemen.



In der initialen Release-Version von 4.2R4 wird mit hoher Wahrscheinlichkeit **unter Mac OS** die Integrierte Vorschau nicht oder nur sehr eingeschränkt verfügbar sein. An einer Beseitigung des Fehlers wird gearbeitet, da es sich jedoch um technische Probleme mit einer Fremdkomponente handelt, ist eine rechtzeitige Fehlerbeseitigung zum Freigabetermin nicht sichergestellt. Darüber hinaus ist die Funktion zur Anzeige von PDF-Dokumenten in der Integrierten Vorschau gegenwärtig aus technischen Gründen auf Mac OS X nicht verfügbar.



Es wird empfohlen, einen Internet Explorer ab Version 8 zu verwenden. Internet Explorer bis Version 8 unterstützen keine Base64-Dekodierung. Dies kann zu Problemen bei der Injektion von Bild-Elementen im Rahmen der Applikationsintegration führen (z. B. bei der Anzeige der Komponente FS_BUTTON (siehe Kapitel 5.4 Seite 106) in der Integrierten Vorschau oder der Integration einer Bilddatenbank, siehe auch Kapitel 3.4 Seite 26).



Die Anzeige von Java-Applets wird vom JxBrowser, der für die Integrierte Vorschau (Menü "Ansicht" / "Browser Engine") verwendet wird, nicht unterstützt.

3.1.4 Microsoft Office-Integration

Die Anzeige von Microsoft Office-Dateiformaten ist nur unter Windows möglich und erfordert eine lokale Office-Installation. Dabei wird jeweils die zum Format des gewählten Mediums passende Anwendung verwendet, also z. B. Microsoft Word für Medien vom Typ DOC, Microsoft Excel für Medien vom Typ XLS usw. Es kann prinzipiell auch ein 64-Bit-JRE verwendet werden, allerdings steht dann die Integrierte Vorschau nicht zur Verfügung.





Die gleichzeitige Verwendung der Integrierten Vorschau für Medien mit Microsoft Office und extern als Microsoft Office-Desktop-Anwendung (z. B. gestartet über Windows oder aus dem JavaClient heraus) kann unter bestimmten Umständen zu Problemen führen und ist keine zugesicherte Produkteigenschaft der FirstSpirit-Integration. In diesem Fall sollte nur entweder die Integrierte Vorschau für Office-Dokumente in FirstSpirit **oder** die Microsoft Office-Desktop-Anwendung verwendet werden und **kein** Mischbetrieb. Informationen zu weiteren Einschränkungen siehe auch Kapitel 4.2.2.2 Seite 39.

3.1.5 Integration des Windows Media Players

Sollen Dateien der FirstSpirit Medien-Verwaltung vom Dateiformat AVI, MPG, MPEG, WMV, ASF, MP3, MP4 über die Windows Media Player-Integration wiedergegeben werden können, muss dieser auf dem Arbeitsplatzrechner installiert sein. Die Verwendung ist nur unter Microsoft Windows möglich.

3.2 Java-Applikationsintegrationen in 4.2R4 (Java Image Editor)

Im Gegensatz zu Picnik und Pixlr (siehe Kapitel 3.3 Seite 25) handelt es sich beim Java Image Editor um eine integrierte Java-Anwendung. Wie der Mozilla Firefox wird er mit dem FirstSpirit Kernprodukt ausgeliefert und von FirstSpirit selbst verwaltet. Er wurde im Rahmen der AppCenter-Entwicklungsarbeiten als Beispiel für die Integration einer Fremdanwendung auf Java-Basis realisiert und war ursprünglich nicht als Produktbestandteil vorgesehen. Da sich aber in der Praxis herausgestellt hat, dass sich eine Reihe von (einfachen) Bildmanipulationen mithilfe dieser Bildbearbeitungssoftware durchführen lassen, wurde diese Fremdimplementierung lizenziert und für die Nutzung in FirstSpirit freigegeben.

Analog zu den anderen Integrationen in Kapitel 3.1 erfolgt die Freigabe für die Nutzung des Java Image Editors explizit "ohne Funktionsgarantie" für die Anwendung selbst. D. h. es wird von Seiten e-Spirit weder explizit noch implizit eine Garantie für die Funktionen der Bildbearbeitung übernommen, sondern die Nutzungsfreigabe erfolgt "as-is" (siehe auch Kapitel 3.1.2 Seite 22). Wenn die Nutzung von Bildbearbeitungsfunktionen eine produktionskritische Funktion ist, sollte eine externe Bildbearbeitungssoftware mit dem entsprechenden Hersteller-Support eingesetzt werden.

Aktuell steht der Java Image Editor unter Mac OS nur sehr eingeschränkt zur Verfügung. Informationen zu weiteren Einschränkungen siehe auch Kapitel 4.2.3.2



Seite 46.

3.3 Applikationsintegrationen im BETA-Stadium

Einige mit dem Kernprodukt ausgelieferte Integrationen befinden sich aktuell noch im BETA-Stadium und sind nicht für den produktiven Einsatz freigegeben:

OpenOffice: OpenOffice-Dateiformate können auf Microsoft- und Linux-Betriebssystemen genutzt werden. Wie bei der Integration der Microsoft Office-Anwendungen muss auch hier die entsprechende OpenOffice-Anwendung auf dem Arbeitsplatzrechner installiert sein. Beim Einsatz von OpenOffice unter Linux kann es gegenwärtig noch zu Darstellungsproblemen kommen, in der Praxis unterscheidet sich die Darstellung aber nur marginal von der Windows-Version.

Google Docs: Mit Google Docs können Office-Dokumente im FirstSpirit-JavaClient auch unter Mac OS X bearbeitet werden. Dazu ist eine aktive Verbindung zum Internet erforderlich sowie ein Google-Konto (<http://docs.google.com>). Es ist also für die Erstellung und Bearbeitung von Office-Dokumenten keinerlei lokal installierte Software mehr erforderlich – im Gegenzug müssen aber alle Dokumente vor der Bearbeitung auf einen Google-Server hochgeladen werden. Dieses datenschutztechnisch sensible Thema, die aktuell noch recht starken Einschränkungen bei der Dokumentenbearbeitung (z. B. Limitierung der Dateigröße und Inkompatibilitäten) und die Verwendung einer noch nicht freigegebenen API führen dazu, dass die Google Docs-Integration in FirstSpirit Version 4.2R4 nur prototypischen Charakter hat und für den produktiven Einsatz **nicht** freigegeben ist. In einigen Fällen können Dokumente durch die Bearbeitung in Google Docs sogar beschädigt werden. Dies ist kein FirstSpirit-Problem.

Die kostenfreie Version von Google Docs unterstützt nur Dateigrößen bis max. 1 MB. Aktuell kann Google Docs im JavaClient nur mit der Browser Engine "Internet Explorer" verwendet werden. Informationen zu weiteren Einschränkungen siehe auch Kapitel 4.2.2.2 Seite 39 und betreffende Internetseite des Anbieters.

Picnik und Pixlr: Die Grafik-Engines Picnik und Pixlr greifen auf die Online-Bildbearbeitungsdienste www.picnik.de und www.pixlr.com zu und benötigen daher jeweils eine aktive Verbindung zum Internet. Auch hier werden die Bilder faktisch auf Server der jeweiligen Anbieter hochgeladen und dort bearbeitet. Dies sollte bei der Verwendung im Hinblick auf den Datenschutz berücksichtigt werden. Für weitere Informationen siehe auch die betreffenden Internetseiten der Anbieter.



3.4 Kundenindividuelle Applikationsintegrationen

Das FirstSpirit AppCenter ermöglicht es Kunden und Partnern, individuelle Anwendungen in den JavaClient zu integrieren und so den Client an die speziellen Bedürfnisse anzupassen. Insbesondere die bereits integrierten Browser stellen dazu eine Infrastruktur bereit, mit der sehr einfach die Integration von Online-Angeboten realisiert werden kann, z. B. für eine schnelle Internet-Recherche nach Text- oder Bildmaterial oder die Nutzung von Google-Diensten. Zur Nutzung von Online-Angeboten ist grundsätzlich eine aktive Internetverbindung, evtl. ein Konto usw. erforderlich.

Auch bei der Integration und Verwendung von kundenindividuellen AppCenter-Anwendungen ist grundsätzlich zu beachten, dass FirstSpirit für die Applikationsintegration die erforderlichen Schnittstellen bereitstellt, aber in der Regel keinen Einfluss auf die integrierten Anwendungen selbst hat. Integrierte Fremdanwendungen sind kein Produktbestandteil von FirstSpirit. Das bedeutet u.a., dass die Verantwortung für die Funktionalität der integrierten Anwendungen beim Hersteller bzw. beim Kunden oder Partner liegt, der die Anwendung realisiert.

Probleme können im Rahmen des FirstSpirit-Produktsupports gemeldet werden und werden (nach Möglichkeit) beseitigt, wenn sie auf der Ebene der Integrations-Schnittstelle liegen. Ein Anspruch auf Fehlerbeseitigungen **innerhalb** der in den Kapiteln 3.1 bis 3.4 beschriebenen Fremdanwendungen gegenüber e-Spirit besteht aber **nicht**.



*Bei der Verwendung von **Mac OS und Linux** ist zu beachten, dass die Funktionsfähigkeit von AppCenter-Anwendungen unter diesen Plattformen nicht garantiert werden kann, daher werden diese Plattformen für AppCenter-Anwendungen nicht unterstützt.*

Für die Integration eigener, einfacher AppCenter-Anwendungen wurde in 4.2R4 u.a. auch die Eingabekomponente FS_BUTTON eingeführt. Mit dieser wird ein individuell definierbares Icon oder eine Schaltfläche zur Verfügung gestellt, das / die nach per Klick ein in FirstSpirit hinterlegtes Skript oder eine Klasse ausführen kann (siehe Kapitel 5.4 Seite 106), die ggf. eine AppCenter-Anwendung realisiert. Eigene Web-Anwendungen zur Verwendung im AppCenter können auf dem Server in den Server-Eigenschaften, Bereich "Web-Applikationen", installiert werden (siehe Kapitel 6.3 Seite 139).

Die Verwendung eigener Anwendungen im AppCenter erfordert eine Lizenz. Nähere



Informationen dazu siehe Kapitel 6.5.2 Seite 145.

3.5 Weiterführende Dokumentation

Detaillierte Informationen zum FirstSpirit AppCenter-Konzept, zur technischen Architektur und illustrierte Anwendungsbeispiele befinden sich auch im *FirstSpirit Whitepaper 2011*.

Informationen, wie individuelle Anwendungen für FirstSpirit realisiert werden können, finden sich im *FirstSpirit Entwicklerhandbuch für Komponenten*. Das dazu gehörige Zip-Archiv (MODDEV4x_modexamples.zip) enthält Beispiele zur allgemeinen Modul- und Komponenten-Entwicklung in FirstSpirit, weitere Beispiele für AppCenter-Anwendungen sind über das e-Spirit Helpdesk erhältlich. Ebenfalls wird eine Dokumentation für Entwickler zum FirstSpirit AppCenter bereitgestellt werden.

Darüber hinaus steht eine Applikationsintegrations-API ("Application-API") zur Verfügung. Diese bleibt in 4.2R4 stabil, mit FirstSpirit Version 5 kann sie sich ggf. ändern. Siehe dazu auch *FirstSpirit Online Dokumentation*, Bereich "Vorlagenentwicklung" / "FirstSpirit API" / "Applikationsintegration".



4 Neue/geänderte Funktionen für Redakteure

4.1 Neue/geänderte Funktionen in JavaClient und WebClient

4.1.1 Eingabekomponenten: Weiterentwicklung FS_LIST

Die Eingabekomponente FS_LIST bildet seit ihrer Einführung in der initialen Release-Version zu 4.2 die Funktionen der Eingabekomponente CMS_INPUT_CONTENTAREALIST ab. Ziel der in 4.2 begonnenen Konsolidierung der FirstSpirit-Eingabekomponenten soll sein, alle Eingabekomponenten, die Listencharakter haben, in FS_LIST zusammenzufassen. Daher wurden in Version 4.2R4 die Funktionen der folgenden Eingabekomponenten hinzugefügt:

- CMS_INPUT_CONTENTLIST
- CMS_INPUT_LINKLIST
- CMS_INPUT_SECTIONLIST
- CMS_INPUT_TABLIST


FS_LIST kann ebenso bedient werden, wie es von diesen Eingabekomponenten bekannt ist. Teilweise werden andere Icons verwendet sowie der bereits aus anderen Eingabekomponenten bekannte Auswahldialog inklusive Suchfunktion.





Abbildung 4-1: FS_LIST zur Auswahl von Datensätzen


Folgende Icons stehen je nach Vorgaben des Vorlagenentwicklers und dem Anwendungszweck der Eingabekomponente zur Verfügung:






 Hinzufügen (nur für Datensätze), durch einen Klick auf dieses Icon öffnet sich ein Auswahldialog zur Auswahl eines Datensatzes



 Neu, im Falle von Datensätzen öffnet sich durch einen Klick auf dieses Icon ein Fenster zur Eingabe eines neuen Datensatzes, im Falle von Absätzen und Verweisen öffnet sich ein Auswahldialog zur Auswahl einer entsprechenden Vorlage


 Kopieren, im Falle von Datensätzen wird durch einen Klick auf dieses Icon durch Kopie des markierten Datensatzes ein neuer Datensatz in der entsprechenden Datenquelle angelegt und zur Liste hinzugefügt.


 Entfernen, durch einen Klick auf dieses Icon wird das markierte Listenelement aus der Liste gelöscht


 Löschen (nur für Datensätze), durch einen Klick auf dieses Icon wird der markierte Datensatz aus der Liste und der betreffenden Datenquelle gelöscht, er kann später nicht wieder ausgewählt werden.

  Nach oben / unten verschieben, durch einen Klick auf dieses Icon wird das markierte Listenelement um eine Position nach oben bzw. unten verschoben

 Bearbeiten, durch einen Klick auf dieses Icon öffnet sich ein Fenster zur Bearbeitung des ausgewählten Listenelements. Im Ansichts-Modus wird das Icon  angezeigt.

 Anzeigen, werden in der Eingabekomponente Übersicht und Detailansichten dargestellt, wird durch einen Klick auf dieses Icon die Detailansicht des in der Übersicht markierten Eintrags aufgeklappt und mit einem weiteren Klick wieder eingeklappt. Wird nur die Übersicht angezeigt, wird mit einem Klick auf einen Eintrag in der Übersicht die entsprechende Detailansicht in einem separaten Fenster angezeigt.

 Springe zu, im Falle von Datensätzen kann mit einem Klick auf dieses Icon zur Datenquellen-Verwaltung gewechselt werden, um den ausgewählten Datensatz dort zu bearbeiten.

 In eigenem Fenster öffnen, durch einen Klick auf dieses Icon öffnet sich der markierte Listeneintrags in einem separaten Fenster



Darüber hinaus ist auch eine Bearbeitung der Listen über ein Kontextmenü oder Tastaturkürzel möglich. Hier stehen dieselben Funktionen zur Verfügung, wie sie die Icons bieten:



Abbildung 4-2: FS_LIST – Kontextmenü

Alle Daten, die im Projekt über die oben genannten Eingabekomponenten eingepflegt worden sind, stehen nach einer Umstellung auf FS_LIST unverändert zur Verfügung. Die Umstellung nimmt in der Regel der Vorlagenentwickler vor.



In **WebEdit** wird FS_LIST in 4.2R4 nur rudimentär unterstützt: aktuell stehen nur die Funktionen der Eingabekomponente CMS_INPUT_CONTENTAREALIST (Liste zur Verwaltung von Absatzinhalten) zur Verfügung. Mehr Funktionen können über den sogenannten Applet-Modus bereitgestellt werden. Dies ist aber nur eine Übergangslösung, eine native FS_LIST-Unterstützung wird es ab FirstSpirit Version 5.0 geben.

Informationen zur Umstellung auf FS_LIST und den Konfigurationsmöglichkeiten siehe auch Kapitel 5.3 Seite 93.

4.1.2 Neue Eingabekomponente: FS_BUTTON

Die neue Eingabekomponente FS_BUTTON stellt je nach Vorgabe durch den Vorlagenentwickler eine Schaltfläche



Abbildung 4-3: FS_BUTTON – Schaltfläche



ein Icon

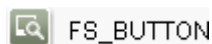


Abbildung 4-4: FS_BUTTON – Icon

oder einen einfachen Verweis

FS_BUTTON

Abbildung 4-5: FS_BUTTON – Verweis

dar. Beim Anklicken wird eine Funktion ausgeführt, die vom Vorlagenentwickler definiert werden kann. Je nach Konfiguration der Eingabekomponente kann die Funktion auch ausgeführt werden, wenn ein Objekt (aus FirstSpirit, Dateien vom lokalen Arbeitsplatzrechner usw.) mit der Maus auf dieses Bedienelement gezogen wird.



Die Eingabekomponente FS_BUTTON wird aktuell im FirstSpirit-WebClient nicht unterstützt. Eine entsprechende Unterstützung ist für FirstSpirit Version 5 geplant.

4.1.3 Referenznamen: Automatische Umwandlung von Sonderzeichen

Referenznamen dürfen in keiner Verwaltung Sonderzeichen enthalten, im FirstSpirit-JavaClient können darüber hinaus keine Großbuchstaben für Referenznamen verwendet werden. Im JavaClient werden diese ungültigen Zeichen noch während der Eingabe entfernt, Großbuchstaben werden in Kleinbuchstaben umgewandelt. Im FirstSpirit WebClient werden Sonderzeichen automatisch in Unterstriche umgewandelt (_) und auch Großbuchstaben sind erlaubt.

Je nach Vorgaben durch den Administrator können ab FirstSpirit Version 4.2R4 Sonderzeichen in Referenznamen automatisch in gültige Zeichen umgeformt werden, im JavaClient direkt bei der Eingabe beim Anlegen eines FirstSpirit-Objekts oder beim Ändern eines Referenznamens (Kontextmenü "Extras" / "Referenznamen ändern"), im WebClient nach dem Speichern. Dazu gibt der Administrator Regeln an, welche Sonderzeichen in welche gültigen Zeichen umgewandelt werden sollen. Ein "Ü" könnte dann z. B. automatisch in ein "ue" transformiert werden, Leerzeichen in Unterstriche usw. Dies ist z. B. sinnvoll, wenn Referenznamen bei der Generierung eines Projekts für die Erzeugung von URLs genutzt werden.



Wird der Referenzname im **JavaClient** aus dem Anzeigenamen erzeugt (z. B. beim Anlegen von Seiten in der Inhalte-Verwaltung), wird der Referenzname ebenfalls entsprechend umgeformt. So könnte z. B. aus dem Anzeigenamen "Über uns" bei der Eingabe der Referenzname "ueber_uns" werden:

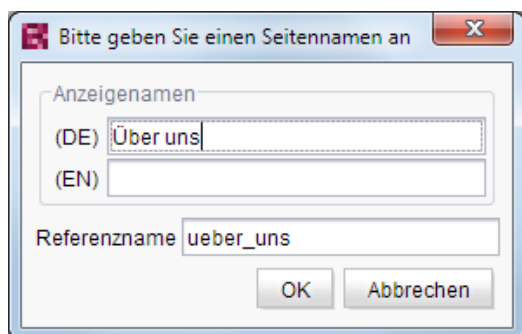


Abbildung 4-6: Referenzname beim Anlegen einer neuen Seite



Referenznamen werden nur dann angezeigt und sind nur dann änderbar, wenn der Administrator dies für den Redakteur bzw. das Projekt so eingestellt hat.



Generell (unabhängig von den Regeln, die der Administrator für die Umwandlung angibt) werden im JavaClient bei Referenznamen Großbuchstaben bei der Eingabe direkt in Kleinbuchstaben umgewandelt. Sonderzeichen, für die der Administrator keine Ersetzungsregel angegeben hat, können nicht in das Feld "Referenzname" eingegeben werden.



Je nach Konfiguration durch den Administrator kann das Ersetzungsergebnis im Feld "Anzeigename" von dem im Feld "Referenzname" abweichen.



Werden in **WebEdit** Anzeigenamen, aus denen Referenznamen gebildet werden, mit Sonderzeichen angegeben, werden diese nach dem Speichern ebenfalls entsprechend umgeformt, z. B.

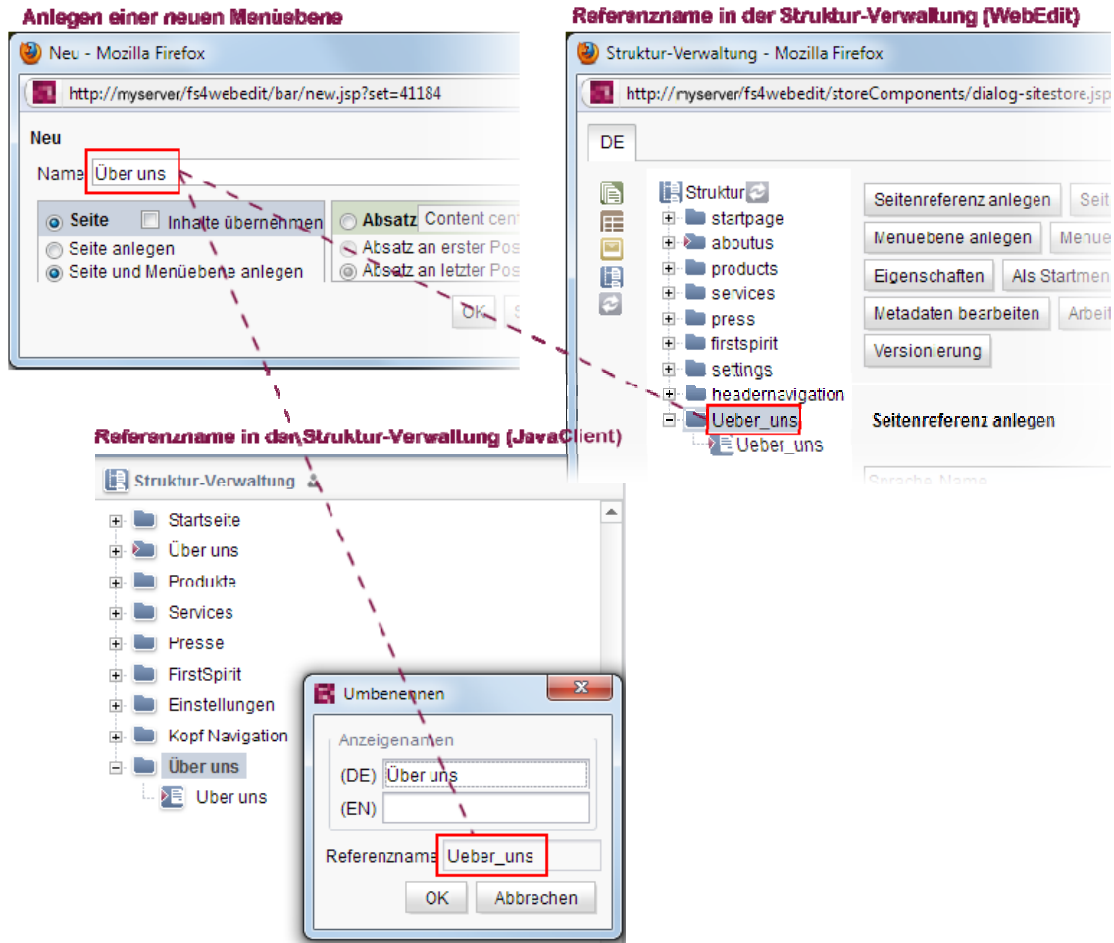


Abbildung 4-7: Umformung von Referenznamen in WebEdit



4.2 Neue/geänderte Funktionen im JavaClient

4.2.1 Neue Icons für Dateien der Medien-Verwaltung

Wurden in der Baumstruktur der Medien-Verwaltung bislang nur unterschiedliche Icons für Bilder und Dateien, sprachabhängig und sprachunabhängig, verwendet, werden nun für Dateien auch spezifische Icons dargestellt, die sich am Dateityp orientieren. So kann direkt in der Baumstruktur abgelesen werden, um welchen Dateityp es sich handelt, ohne erst in die Detailansicht wechseln zu müssen. Diese Icons werden nun auch in der Ordner-Übersicht der Medien-Verwaltung (siehe Abbildung 4-8), auf den Arbeitsbereich-Registern, in Auswahldialogen usw. verwendet, z. B.:

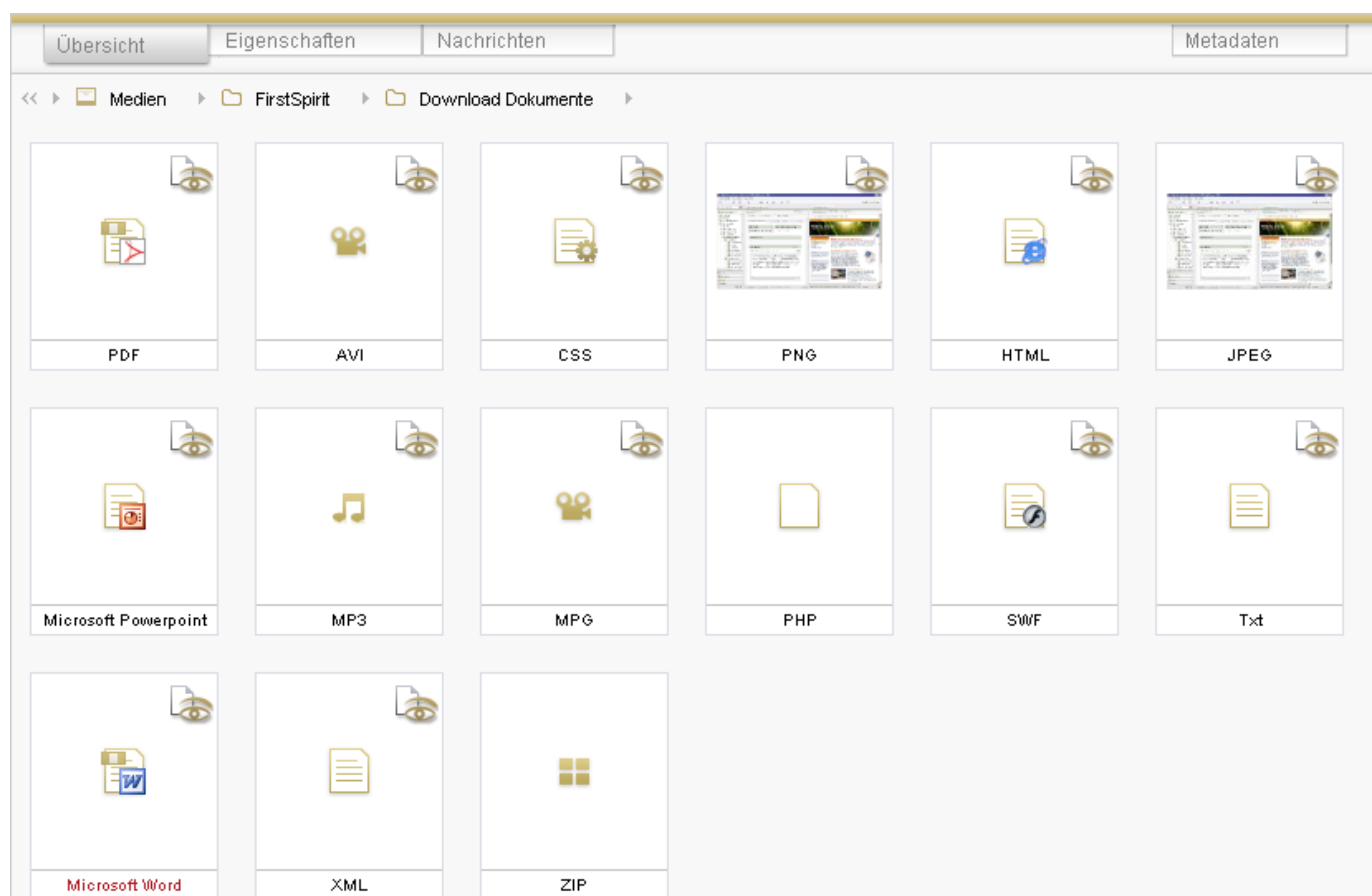





Abbildung 4-8: Medien-Verwaltung – Ordner-Übersicht

Dateitypen, für die kein eigenes Icon verwendet wird, werden dabei mit dem Icon  angezeigt, sprachabhängige Dateien sind besonders gekennzeichnet: 



Ist die Integrierte Vorschau für Medien aktiviert (siehe Kapitel 4.2.2.3 Seite 41), zeigt das Icon  an, dass die betreffende Datei in der Integrierten Vorschau (siehe Kapitel 4.2.2 Seite 35) dargestellt werden kann. Mit einem Klick darauf öffnet sich das Medium in der Integrierten Vorschau.

4.2.2 Funktionserweiterungen der Integrierten Vorschau

Die in FirstSpirit Version 4.2 im JavaClient eingeführte "Integrierte Vorschau" kann ab 4.2R4 nicht nur wie bisher für eine Vorschau der aktuell bearbeiteten Inhalte verwendet werden, sondern es können (bei korrekter Konfiguration (siehe Kapitel 4.2.2.3 Seite 41)) auch andere Dateiformate (z. B. Microsoft Word- oder Excel-Dateien) siehe Kapitel 4.2.2.1 Seite 38) und die FirstSpirit Online Hilfe (siehe Kapitel 4.2.2.4 Seite 43) nun dort dargestellt werden, und zwar mithilfe einer zum Format passenden Anwendung.



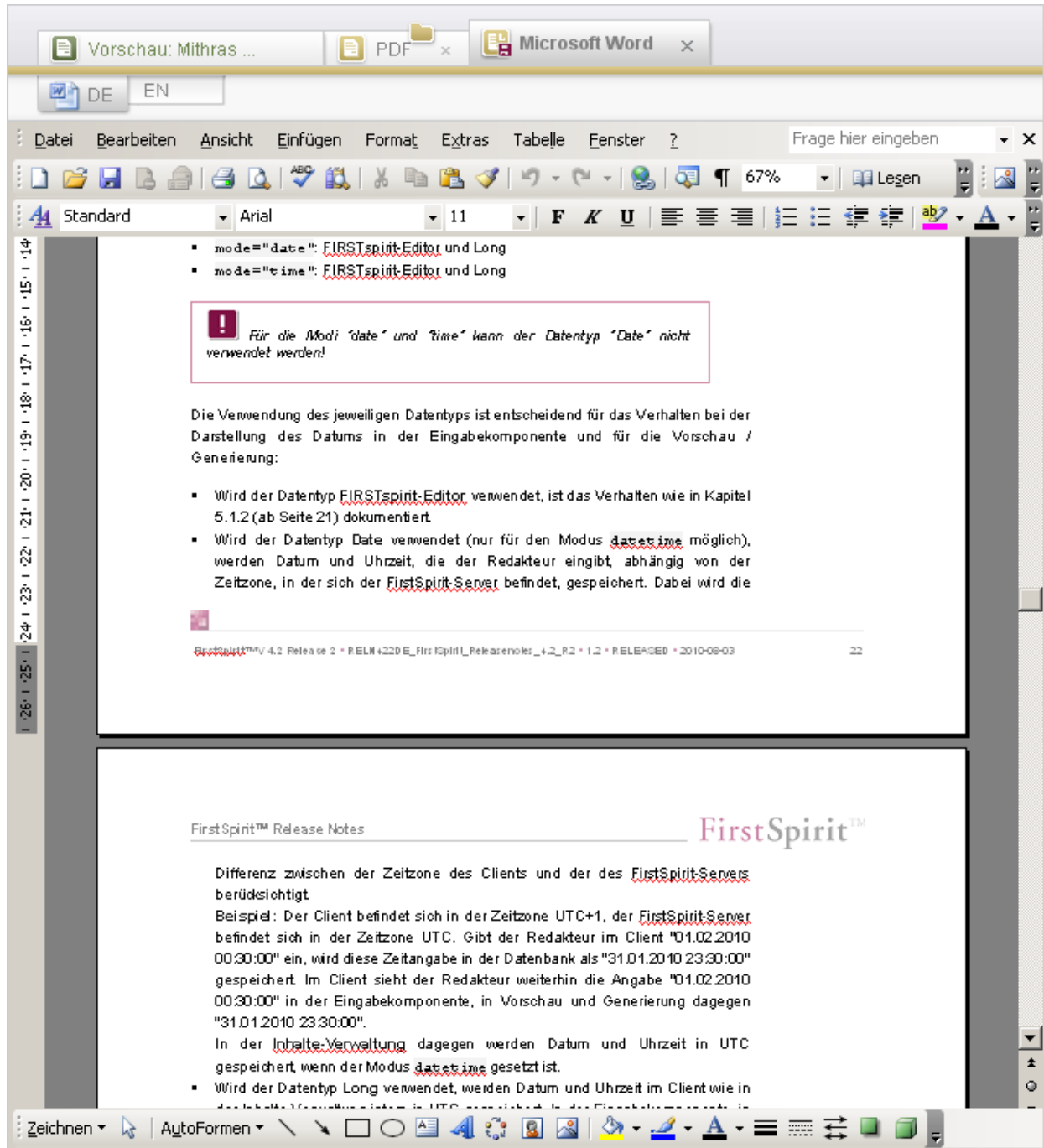


Abbildung 4-9: Anzeige eines Microsoft-Word-Dokuments in der Integrierten Vorschau


Dazu können mehrere **Register** in der Integrierten Vorschau gleichzeitig geöffnet werden, z. B. über

- das Vorschau-Icon der JavaClient-Symboleiste,
- Strg + P,
- den Kontextmenü-Eintrag "Vorschau"



- das neue Vorschau-Icon auf Ordner-Ebene der Medien-Verwaltung (siehe auch Abbildung 4-8)
- den Wechsel in den Bearbeitungs-Modus

Die Register können allerdings nicht wie im Bearbeitungsbereich (mittlere Spalte des JavaClients) verschoben werden: Sie behalten immer die Reihenfolge bei, in der sie geöffnet wurden, bis sie geschlossen werden. Ganz links befindet sich dabei das Register mit der Vorschau der Ausgabekanäle (sofern die "Integrierte Vorschau für Inhalte" aktiviert ist, siehe dazu auch Kapitel 4.2.2.3 Seite 41). Die Register zeigen den Namen des jeweiligen Mediums (bzw. "Hilfe", wenn die FirstSpirit Online Hilfe geöffnet ist) und das Icon für den Dateityp an. Zusätzlich wird anhand des Icons dargestellt, ob es sich um ein sprachabhängiges Medium handelt und ob das Medium nicht gespeicherte Änderungen aufweist.

Die aus dem mittleren Bearbeitungsbereich des JavaClients bekannte Funktion "Baumsynchronisation (de)aktivieren" steht in der Medien-Vorschau nicht zur Verfügung: die Synchronisation ist hier ständig aktiv, solange in der Baumstruktur die Medien-Verwaltung angezeigt wird, sich der Benutzer also in der Medien-Verwaltung befindet. Wird dann in der Integrierten Vorschau zu einem anderen Register gewechselt, wird in der Baumstruktur zu diesem Medium gewechselt und damit wird es auch im Bearbeitungsbereich angezeigt. Wurde ein Medium aus einer Ordner-Übersicht heraus in der Vorschau geöffnet (siehe Kapitel 4.2.1 Seite 34), wird dies durch das Ordner-Icon  auf dem jeweiligen Register angezeigt. Mit einem Klick auf dieses Register wird im Bearbeitungsbereich nicht das Medium sondern der beinhaltende Ordner angezeigt. Ein Tooltip gibt den Namen des Ordners an. Über das Kontextmenü auf dem Ordner-Icon kann zur Ordner-Übersicht gewechselt werden.


Auf den **hängenden Registern** werden weitere Informationen zu der aktuell dargestellten Datei angezeigt:

- "Medium": Es handelt sich um ein sprachunabhängiges Medium, bei einigen Dateiformaten wird auch ein Dateityp-Icon angezeigt.
- "DE" / "EN" usw.: Es handelt sich um ein sprachabhängiges Medium, bei einigen Dateiformaten wird auch ein Dateityp- bzw. Anwendungs-Icon angezeigt, bei PDFs und Flash-Dateien wird z. B. das Icon des jeweils gewählten Browsers angezeigt. Über das jeweilige Sprachregister kann zur Medien-Vorschau in der gewünschten Sprache gewechselt werden.

Vorschaubereich schließen / Andere Vorschaubereiche schließen: Diese Kontextmenüeinträge stehen, wie im Bearbeitungsbereich (mittlere Spalte des



JavaClients), auf jedem Register zur Verfügung.

: Über dieses Icon können die einzelnen Register geschlossen werden, bis auf das, welches die Vorschau der Ausgabekanäle enthält. Wird der Browser gewechselt (Menü "Ansicht" / "Browser Engine"), schließen sich die Register ebenfalls.

4.2.2.1 Vorschau von Medien

Um die Integrierte Vorschau in der Medien-Verwaltung von der bisher bekannten Integrierten Vorschau für HTML-Inhalte abgrenzen zu können, wird die bisherige HTML-Vorschau in der rechten Spalte des JavaClients als "Integrierte Vorschau für Inhalte" bezeichnet, die neue Vorschau als "Integrierte Vorschau für Medien". In der Integrierten Vorschau für Medien können Medien – abhängig von der Vorgabe durch den Projekt-Administrator (siehe Kapitel 6.2 Seite 133 ff.) – folgendermaßen dargestellt werden:

- **Bilder** werden in der jeweiligen Sprache und in den verschiedenen Auflösungen über eine java-basierte Anwendung angezeigt. Zur Anzeige einer bestimmten Auflösung muss in der Liste der Auflösungen die gewünschte Auflösung markiert und das Vorschau-Icon angeklickt oder alternativ per Doppelklick aktiviert werden. Je nach Vorgabe durch den Projekt-Administrator kann statt einer Vorschau auch eine Bildbearbeitungsanwendung zur Verfügung stehen (siehe Kapitel 4.2.3 Seite 44).
- **PDF- und Flash-Dokumente** werden mithilfe eines Plugins des verwendeten Browsers, mit dem diese Dokumente dargestellt werden können, angezeigt. Das entsprechende Plugin muss dazu auf dem Arbeitsplatzrechner des Redakteurs installiert und korrekt konfiguriert sein. Es stehen alle Funktionen des Plugins zur Verfügung.
- **Office-Dokumente** (Microsoft Word, Excel, Powerpoint; OpenOffice Writer, Calc, Impress [nur BETA-Test-Stadium!]) können in der jeweiligen Microsoft Office-Anwendung, die auf dem Arbeitsplatzrechner installiert ist bzw. über OpenOffice oder Google Docs [jeweils BETA-Test-Stadium!] geöffnet werden. So kann ein Word-Dokument z. B. direkt in der FirstSpirit Medien-Verwaltung bearbeitet werden. Dazu muss in den Bearbeitungs-Modus gewechselt werden. Änderungen müssen über die Speicher-Funktionen des JavaClient gespeichert werden, damit das geänderte Dokument im JavaClient verfügbar ist.





Werden Änderungen an Dokumenten, z. B. Microsoft Excel-Tabellen, vorgenommen, die sich nicht im FirstSpirit-Bearbeitungs-Modus befinden, können bei der Bearbeitung der Dokumente Hinweis-Meldungen der jeweiligen Microsoft Office-Anwendung angezeigt werden, die ggf. mit "OK" geschlossen werden können.



Die OpenOffice- sowie Google Docs-Integration befindet sich aktuell im BETA-Test-Stadium und ist nicht offiziell freigegeben!

- **Audio- und Video-Dateien** (z. B. MP3, WMV, AVI, MPG) werden in der Integrierten Vorschau über den Windows Media Player abgespielt.
- **Text-Dateien** (z. B. TXT, HTML, XML, CSS) werden über einen in FirstSpirit integrierten Text-Editor dargestellt. FirstSpirit-Syntax wird dabei farbig hervorgehoben (Syntax-Highlighting).



Damit das gewünschte Dokument aus der Medien-Verwaltung (z. B. Word-Datei) richtig dargestellt und bearbeitet werden kann, ist die korrekte Abspeicherung in der FirstSpirit Medien-Verwaltung als Bild oder Datei wichtig.

Die Medien-Dateien können auch in einem separaten Fenster dargestellt werden (Menü "Ansicht" / "Anzeigebereich der Vorschau" / "in separatem Fenster").

4.2.2.2 Hinweise und Einschränkungen

Microsoft Office-Anwendungen

Die gleichzeitige Verwendung der Integrierten Vorschau für Medien mit Microsoft Office und extern als Microsoft Office-Desktop-Anwendung (z. B. gestartet über Windows oder aus dem JavaClient heraus) kann unter bestimmten Umständen zu Problemen führen und ist keine zugesicherte Produkteigenschaft der FirstSpirit-Integration. In diesem Fall sollte nur entweder die Integrierte Vorschau für Office-Dokumente in FirstSpirit **oder** die Microsoft Office-Desktop-Anwendung verwendet werden und **kein** Mischbetrieb.

Im Falle von Microsoft Excel ist beispielsweise keine Bearbeitung von Excel-Dateien über die Desktop-Anwendung möglich, wenn in der Integrierten Vorschau eine



Excel-Datei angezeigt wird: weder können bereits geöffnete Excel-Dateien bearbeitet werden, noch können Excel-Dateien z. B. per Doppelklick in der Desktop-Anwendung geöffnet werden. Werden alle Reiter der Integrierten Vorschau, die eine Vorschau zu einer Excel-Datei enthalten, geschlossen, ist auch wieder eine Bearbeitung und das Öffnen von Excel-Dateien möglich.

Einige Menü-Funktionen der Office-Anwendungen können darüber hinaus aus technischen Gründen nicht ausgeführt werden (z. B. "Datei" / "Neu"). Oder sind in Dokumenten beispielsweise Bilder als Verknüpfung eingefügt (und nicht eingebettet), kann auf diese nach dem Hochladen in den JavaClient nur dann zugegriffen werden, wenn der Pfad zu diesen Bildern korrekt ist. Probleme kann es geben, wenn es sich um relative Pfade handelt oder um absolute Pfade, die z. B. aufgrund der individuellen Systemeinstellung verschiedener Redakteure, die im selben Projekt arbeiten, abweichen (z. B. abweichender Laufwerksbuchstabe der Netzwerkadresse).

Anzeige von PDF-Dateien

Beim Start des JavaClients kann sich möglicherweise ein Fenster des Adobe Readers öffnen. Dies geschieht aus technischen Gründen und kann durch geeignete Einstellungen in der lokalen Installation des Adobe Acrobat Readers auf dem Arbeitsplatzrechner unterbunden werden. Je nach verwendeter Version muss dazu im Menü "Datei" oder "Bearbeiten" der Menüpunkt "Voreinstellungen" / "Grundeinstellungen" aufgerufen werden. Dort muss die Option "PDF in Browser anzeigen" bzw. "Web-Browser Integration aktivieren" aktiviert werden.

Google Docs

Die Verwendung von Google Docs für die Bearbeitung von Office-Dokumenten ist aktuell keine zugesicherte Produkteigenschaft. In der Praxis ist noch eine Reihe von Problemen zu erwarten. Dabei ist nicht auszuschließen, dass Dokumente durch die Bearbeitung in Google Docs beschädigt werden. Dies ist KEIN FirstSpirit-Problem!

Zur Verwendung von Google Docs ist eine Anmeldung bei Google erforderlich, dazu erfolgt im JavaClient eine entsprechende Abfrage:





Abbildung 4-10: Anmeldung zur Google Docs-Nutzung

Die kostenfreie Version von Google Docs unterstützt nur Dateigrößen bis max. 1 MB. Aktuell kann Google Docs im JavaClient nur mit der Browser Engine "Internet Explorer" verwendet werden.

4.2.2.3 Konfiguration über das Menü "Ansicht"

Über das Menü "**Ansicht**" / "**Integrierte Vorschau**" kann eingestellt werden, ob die Integrierte Vorschau wie bisher nur für Inhalte (Inhalte- und Struktur-Verwaltung) oder auch für Medien (Medien-Verwaltung) verwendet werden soll:

- **für Inhalte verwenden:** Mit dieser Einstellung wird die Integrierte Vorschau wie bisher für Inhalte in der Inhalte- und Struktur-Verwaltung verwendet:

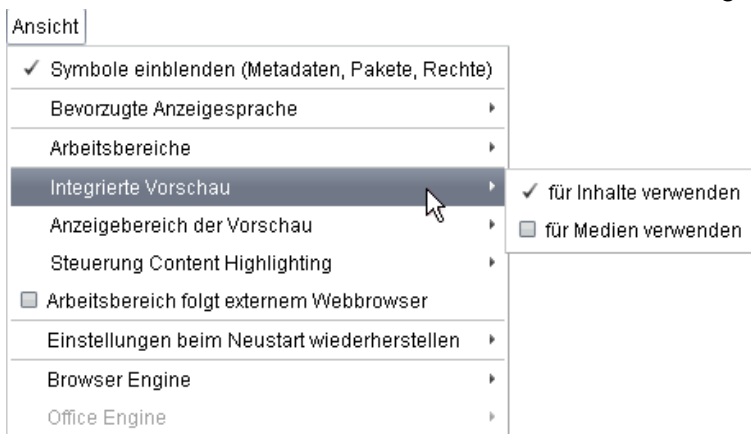


Abbildung 4-11: Menü "Ansicht" / "Integrierte Vorschau"

In diesem Fall wird für die Vorschau von Inhalten die Integrierte Vorschau verwendet, für die Vorschau und Bearbeitung von Medien die unter "Benutzereinstellungen", Register "Editoren" (unter "Globale Einstellungen") ausgewählte oder die im Betriebssystem eingetragene Standard-Anwendung.

- **für Inhalte verwenden** und **für Medien verwenden:** Sollen in der Integrierten Vorschau auch Medien der Medien-Verwaltung angezeigt werden und ggf.



bearbeitet werden können, müssen beide Optionen aktiviert werden. Über den Menüeintrag "Office Engine" kann ausgewählt werden, ob zur Anzeige und Bearbeitung von Medien aus der Medien-Verwaltung Microsoft Office, OpenOffice (zurzeit nur BETA-Test-Phase!) oder Google Docs (zurzeit nur BETA-Test-Phase!) verwendet werden soll. Siehe auch weiter unten.

- **für Medien verwenden:** Soll die Integrierte Vorschau nur für Medien der Medien-Verwaltung verwendet werden, kann die Option "für Medien verwenden" aktiviert werden. In diesem Fall wird für die Vorschau von Inhalten der in den "Benutzereinstellungen", Register "Browser" (unter "Globale Einstellungen") ausgewählte oder der im Betriebssystem eingetragene Standard-Browser verwendet. Der Menüeintrag "Steuerung Content Highlighting" ist dann ausgegraut.
- Sind **beide Optionen deaktiviert**, wird für die Vorschau von Inhalten der in den "Benutzereinstellungen", Register "Browser" (unter "Globale Einstellungen") ausgewählte oder der im Betriebssystem eingetragene Standard-Browser verwendet, für die Vorschau von Medien die unter "Benutzereinstellungen", Register "Editoren" (unter "Globale Einstellungen") ausgewählte oder die im Betriebssystem eingetragene Standard-Anwendung.

Ist die Option "für Medien verwenden" aktiviert, kann über das Menü "**Ansicht**" / "**Office Engine**" eingestellt werden, welche Anwendung für Microsoft Office- und OpenOffice-Dateiformate verwendet werden soll. Um Microsoft Office und OpenOffice verwenden zu können, müssen diese Anwendungen auf dem Arbeitsplatzrechner installiert sein.

- **Microsoft Office (nur Windows):** Ist diese Option aktiviert, wird zur Darstellung und Bearbeitung von Office-Dokumenten in der Integrierten Vorschau die betreffende Microsoft Office-Anwendung verwendet. Wird der JavaClient auf einem anderen Betriebssystem als Windows betrieben, wird dieser Menüpunkt nicht angezeigt.
- **OpenOffice (BETA, nicht MacOS):** Ist diese Option aktiviert, wird zur Darstellung und Bearbeitung von Office-Dokumenten in der Integrierten Vorschau die betreffende OpenOffice-Anwendung verwendet. Wird der JavaClient auf einem MacOS betrieben, wird dieser Menüpunkt nicht angezeigt.
- **Google Docs (BETA):** Ist diese Option aktiviert, wird zur Darstellung und Bearbeitung von Office-Dokumenten in der Integrierten Vorschau Google Docs verwendet.





Die OpenOffice- sowie Google Docs-Integration befindet sich aktuell im BETA-Test-Stadium und ist nicht offiziell freigegeben!

- **deaktiviert:** Ist diese Option aktiviert, wird weder Microsoft Office noch OpenOffice zur Darstellung von Office-Dokumenten verwendet, sondern diese werden wie bisher in einer externen Anwendung geöffnet.

Ist die Option "für Medien verwenden" aktiviert, kann über das Menü "**Ansicht**" / "**Grafik Engine**" eingestellt werden, welche Anwendung für Bild-Dateiformate verwendet werden soll.

- **Java Image Editor:** Ist diese Option aktiviert, wird zur Darstellung und Bearbeitung von Bildern der Java Image Editor verwendet (siehe Kapitel 4.2.3.2 Seite 46).
- **Einfache Bildbearbeitung (Picnik):** Ist diese Option aktiviert, wird zur Darstellung und Bearbeitung von Bildern der Online-Bildbearbeitungsdienst www.picnik.de verwendet (siehe Kapitel 4.2.3 Seite 44).
- **Erweiterte Bildbearbeitung (Pixlr):** Ist diese Option aktiviert, wird zur Darstellung und Bearbeitung von Bildern der Online-Bildbearbeitungsdienst www.pixlr.com verwendet (siehe Kapitel 4.2.3 Seite 44).
- **deaktiviert:** Ist diese Option aktiviert, stehen die bislang bekannten Bildbearbeitungsfunktionen zur Verfügung (siehe Kapitel 4.2.3.1 Seite 46).




Alle hier erläuterten Menüpunkte können je nach Vorgabe des Projekt-Administrators und je nach Betriebssystem, auf dem der JavaClient aktuell betrieben wird, ausgegraut bzw. nicht vorhanden sein.


4.2.2.4 Anzeige der FirstSpirit Online Hilfe

Während die FirstSpirit Online Hilfe (ODFS) bisher nur in einem externen Browser angezeigt wurde, kann sie jetzt in der Integrierten Vorschau geöffnet werden. Die Suche, die im ODFS vorhanden ist, kann ebenfalls in der Integrierten Vorschau genutzt werden. Je nach dem, in welchem Kontext sich der Benutzer aktuell befindet, wird beim Aufruf der Hilfe die entsprechende Dokumentation angezeigt. Bei der Arbeit in der Inhalte-Verwaltung wird in der internen Vorschau beim Aufruf der Hilfe z. B. das Kapitel "Inhalte-Verwaltung des JavaClient" im *FirstSpirit Handbuch für Redakteure (JavaClient)* geöffnet.



Die FirstSpirit Online Hilfe kann über F1, das Icon  der JavaClient-Symbolleiste oder das "Hilfe"-Menü der JavaClient-Menüleiste angezeigt werden. Dazu muss die Integrierte Vorschau für Inhalte aktiviert sein (Menü "Ansicht" / "Integrierte Vorschau" / "für Inhalte verwenden").

4.2.3 Bildbearbeitung im AppCenter

Zu den bisherigen Zuschnittsfunktionen für Bilder der Medien-Verwaltung innerhalb des JavaClients (über die Funktion "Bildausschnitt bearbeiten"  im Arbeitsbereich eines Bildes) kommen in Version 4.2R4 weitere hinzu. Diese werden über spezielle Bildbearbeitungsanwendungen ("Editoren") im AppCenter zur Verfügung gestellt, bieten erweiterte Bearbeitungsfunktionen und können – je nach Einstellungen des Projekt-Administrators (vgl. Kapitel 6.2.4 Seite 138) über die Menüeinträge

- Java Image Editor
- Einfache Bildbearbeitung (Picnik) und
- Erweiterte Bildbearbeitung (Pixlr)

im Menü "Ansicht" / "Grafik Engine" aktiviert werden. Diese Editoren unterscheiden sich hinsichtlich ihrer Bedienung und ihres Funktionsumfangs. Dabei handelt es sich bei Picnik und Pixlr um Online-Bildbearbeitungsdienste, die auf www.picnik.com bzw. www.pixlr.com zugreifen. Picnik ermöglicht eine einfache, intuitive Veränderung von Bildern, Pixlr orientiert sich eher an Profi-Bildbearbeitungssoftware wie Adobe Photoshop. Zur Verwendung dieser Editoren ist eine aktive Internetverbindung erforderlich. Weitere Informationen sind auf den entsprechenden Internetseiten zu finden. Der Java Image Editor hingegen ist eine integrierte Java-Anwendung, die ohne Internetzugang und speziell auf dem Arbeitsplatzrechner installierte Software auskommt. Auch sein Funktionsspektrum orientiert sich an dem von Profi-Bildbearbeitungssoftware.



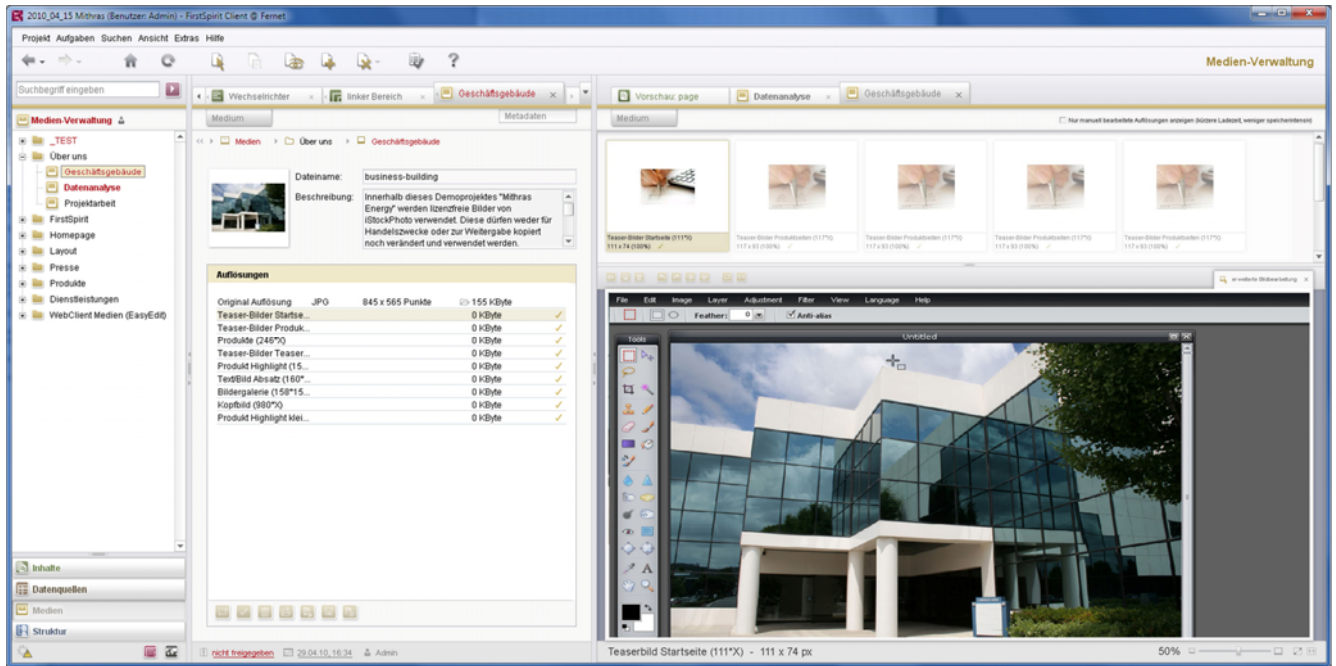





Abbildung 4-12: JavaClient – Erweiterte Bildbearbeitung

Zu jedem Bild der Medien-Verwaltung wird im AppCenter (dazu muss die Option "Integrierte Vorschau für Medien" im Menü "Ansicht" / "Integrierte Vorschau" aktiviert sein, siehe auch Kapitel 4.2.2.3 Seite 41) oben eine Leiste ("Ribbon") mit den verschiedenen Auflösungen des Bildes angezeigt. Sind mehr Auflösungen vorhanden, als in der aktuellen Höhe darstellbar, kann die Leiste vertikal gescrollt werden. Bei Bedarf kann die Leiste eingeklappt werden. Die Original-Auflösung sowie manuell bearbeitete Auflösungen werden mit weißem Hintergrund, die vom System automatisch berechneten, nicht bearbeiteten Auflösungen abgeschwächt dargestellt. Dabei werden auch hier, wie von der bisherigen Auflösungsliste im Arbeitsbereich (mittlerer Bereich des JavaClients) bekannt, Bildvarianten mit dem Icon  markiert, deren Auflösung nicht mit der vorgegebenen Auflösung übereinstimmen. Die Bearbeitungsfunktionen des gewählten Editors beziehen sich immer auf die in der Leiste gewählte Auflösung (gelbe Umrandung). Unterhalb befindet sich der Bereich, in dem das Bild mit den jeweils zur Verfügung stehenden Funktionen bearbeitet werden kann.



Für jedes Bild, das im AppCenter dargestellt oder bearbeitet werden soll, muss ein eigener Reiter geöffnet werden, und zwar über das Wechseln in den Bearbeitungsmodus (Kontextmenü "Bearbeiten an/aus", Icon "In den Bearbeitungsmodus wechseln" der Symbolleiste oder Strg + E) oder das Anfordern einer Vorschau (Kontextmenü "Vorschau", das Icon "Vorschau" der Symbolleiste oder Strg + P).



Wird der "Bearbeiten"-Modus für ein Bild aktiviert, stehen zunächst die bisher bekannten Bearbeitungsfunktionen (Icon  im Arbeitsbereich) zur Verfügung (siehe Kapitel 4.2.3.1 Seite 46).

Über die Schaltfläche "Erweiterte Bildbearbeitung" wird die erweiterte Bildbearbeitung gestartet, und zwar mit dem Editor, der über das Menü "Ansicht" / "Grafik Engine" eingestellt ist. Die Funktionen, die über das Icon  zur Verfügung standen, sind dann nicht mehr vorhanden. Wird die Schaltfläche "Erweiterte Bildbearbeitung" erneut angeklickt, wird wieder zum einfachen Bildbearbeitungs-Modus zurückgekehrt.

4.2.3.1 Einfache Bildbearbeitung

Um die einfache Bildbearbeitung mit den Funktionen zu starten, die bisher im Arbeitsbereich zur Verfügung standen, muss das Icon  im AppCenter angeklickt werden. Es kann dann nur das aktuell ausgewählte Bild / die ausgewählte Auflösung bearbeitet werden. Soll eine andere Auflösung bearbeitet werden, muss das Icon  erneut angeklickt werden, erst dann kann eine andere Auflösung ausgewählt werden.

Das Register "Vorschau" im Arbeitsbereich kann bei Bedarf über eine Aktualisierung mittels F5 oder das entsprechende Icon in der Symbolleiste ein- oder ausgeblendet werden.

4.2.3.2 Der Java Image Editor

Um ein Bild / eine Auflösung mit dem Java Image Editor bearbeiten zu können, muss im Menü "Ansicht" / "Grafik Engine" der Menüpunkt "Java Image Editor" ausgewählt sein. Das zu bearbeitende Bild in der Medien-Verwaltung muss in den Bearbeitungs-Modus gesetzt werden. Im Ribbon muss die gewünschte Auflösung per Mausklick ausgewählt werden, so dass die Schaltfläche "Erweiterte Bildbearbeitung" aktiv wird.

Nach einem Klick auf diese Schaltfläche wird im Bearbeitungsfenster unterhalb des Ribbons der Editor mit seinen Bearbeitungsfunktionen und -werkzeugen sowie der Bilddatei zu der im Ribbon gewählten Auflösung angezeigt:



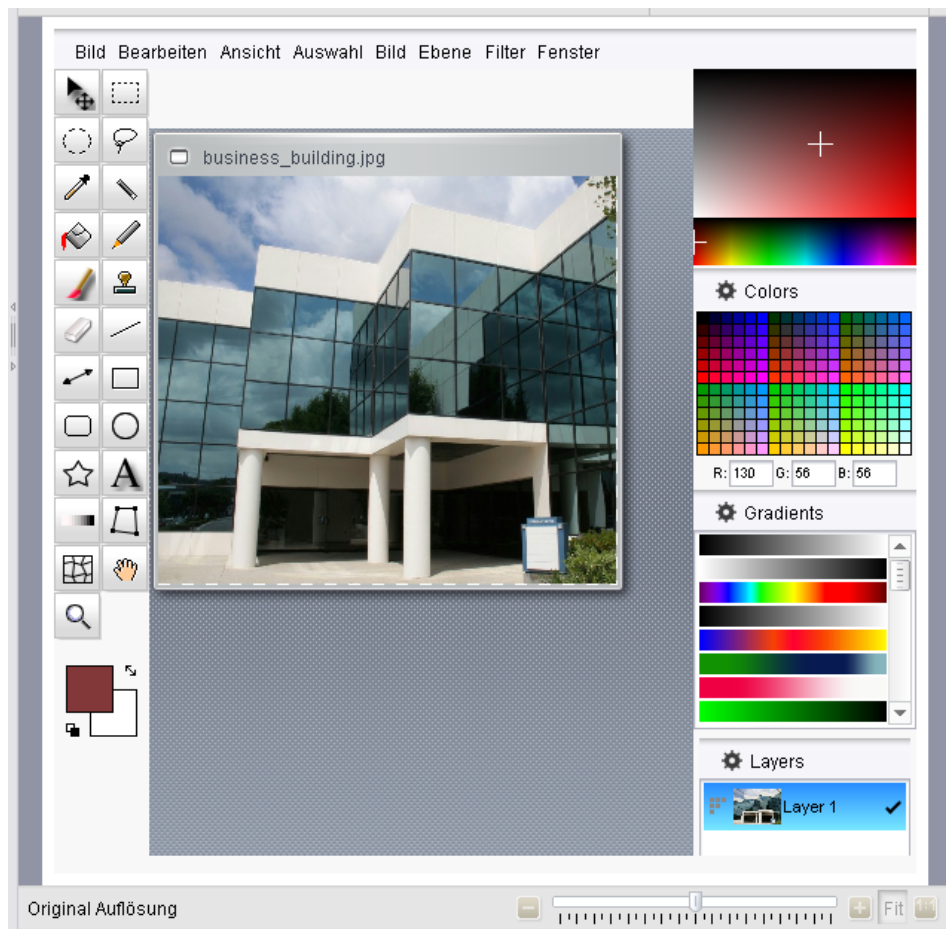


Abbildung 4-13: Erweiterte Bildbearbeitung – Java Image Editor

Das Bild kann mit den zur Verfügung stehenden Funktionen bearbeitet werden. Diese sind vergleichbar mit denen anderer einschlägiger Bildbearbeitungsprogramme.



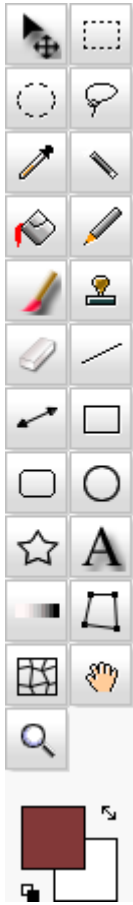
Einige Menüs, Tooltips und Konfigurationsdialoge (z. B. bei Filtern) können aus technischen Gründen nicht lokalisiert werden und stehen daher nur in Englisch zur Verfügung.



Einige der in den Menüs angezeigten Tastaturkürzel funktionieren für den Editor nicht, sondern führen die bislang bekannte Funktion im JavaClient aus.



4.2.3.2.1 Werkzeugleiste



Manche Werkzeuge stellen erweiterte Konfigurationsmöglichkeiten zur Verfügung. Diese werden oberhalb des Bildes, unterhalb der Menüleiste angezeigt.

Auswahlwerkzeuge

Über dieses Icon kann eine Auswahl bewegt oder über die Anfasser der Auswahl verformt werden.



Über diese Icons kann ein quadratischer / rechteckiger, runder / ovaler oder ein Ausschnitt mit benutzerdefinierter Form des Bildes zur weiteren Bearbeitung ausgewählt werden. Eine solche Auswahl kann z. B. über das Menü "Bearbeiten" angewendet, über das Menü "Auswahl" modifiziert und über das Icon



bewegt werden. Alle Malwerkzeuge (s.u.) werden nur innerhalb einer Auswahl angewendet. Zusätzliche Funktionen stehen mit gedrückter Strg-Taste zur Verfügung.





Über dieses Icon können zusammenhängende Bereiche ausgewählt werden, die eine gleiche oder ähnliche Farbe haben.



Über dieses Icon kann die Farbe von Pixeln des Bildes aufgenommen werden.

Mal- und Zeichenwerkzeuge



Über dieses Icon können aneinander grenzende Flächen mit ähnlichem Farbwert gefüllt werden.



Mit diesem Icon kann eine hartkantige Freihandlinie gezeichnet werden.



Mit diesem Icon können weiche Farbstriche erstellt werden.



Mit diesem Icon kann ein definierter Bereich des Bildes dupliziert werden.



Mit diesem Icon können Pixel des Bildes transparent gemacht werden.



Mit diesen Icons können gerade Linien bzw. Pfeile mit definierbarer Breite und Füllmethode gezogen werden.



Mit diesen Icons können ausgefüllte oder nicht ausgefüllte Quadrate / Rechtecke, mit eckigen oder abgerundeten Ecken, Kreise / Ovale bzw. Sterne erstellt werden.

Ansicht



Hat das Bildfenster aufgrund der eingestellten Zoom-Stufe Scrollleisten, kann das Bild mit diesem Icon innerhalb des Fensters bewegt werden.



Mit diesem Icon kann die Zoom-Stufe vergrößert / verkleinert (Strg) werden.

Sonstige



Mit diesem Icon kann eine Ebene mit Text ins Bild eingefügt werden.

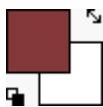




Mit diesem Icon können Farbverläufe eingefügt werden.



Mit diesen Icons kann die Perspektive des Bildes durch das Versetzen der Eckpunkte geändert oder das Bild mithilfe eines Rasters verzerrt werden.



Über diese Icons werden die aktuelle Vorder- und Hintergrundfarbe angezeigt. Sie können über das Pfeil-Icon vertauscht werden. Über das Schwarz-Weiß-Icon kann Schwarz und Weiß als Hintergrund-/Vordergrundfarbe gewählt werden.

4.2.3.2.2 Menü "Bearbeiten"

Über dieses Menü können Auswahlen angewendet werden: Sie können

- ausgeschnitten (**Ausschneiden**)
- kopiert (**Kopie**)
- eingefügt (**Einfügen**)
- gelöscht (**Entfernen**)

werden.

- Existieren mehrere Ebenen übereinander, wird über **Auf eine Ebene reduziert kopieren** im ausgewählten Bereich eine auf eine Ebene reduzierte Kopie aller sichtbaren Ebenen erstellt.
- Darüber hinaus können hier Bearbeitungsschritte rückgängig gemacht werden (**Undo**).

4.2.3.2.3 Menü "Ansicht"

Über dieses Menü kann die Ansicht des Bildes gesteuert werden:

- Über die Einträge **Ansicht vergrößern** und **Ansicht verkleinern** kann das Bild abgestuft gezoomt werden.
- Über die Einträge **1:1**, **2:1**, **4:1**, **8:1** und **16:1** kann eine bestimmte Zoom-Stufe eingestellt werden.
- Über den Eintrag **Raster einblenden** kann ein Raster über dem Bild angezeigt werden.
- Über den Eintrag **Auswahl als Maske anzeigen** kann die Auswahl als Maske dargestellt werden.



4.2.3.2.4 Menü "Auswahl"

Über dieses Menü kann eine bestehende Auswahl modifiziert werden:

- **Alle auswählen:** Wählt das gesamte Bild aus.
- **Auswahl aufheben:** Hebt die Auswahl aus, so dass nichts mehr im Bild selektiert ist.
- **Umkehren:** Der zuvor nicht ausgewählte Bildbereich wird ausgewählt.
- **Unschärf selektieren...:** Erstellt eine Auswahl mit weichen Kanten. Der Radius kann vorgegeben werden.
- **Vergößern:** Vergrößert die Auswahl um den vorgegebenen Radius.
- **Anhand von Transparenz:** Wählt alle transparenten Bereiche des Bildes aus.
- **Neue Ebene aus Auswahl:** Erstellt eine Ebene auf Basis der Auswahl.

4.2.3.2.5 Menü "Bild"

Über dieses Menü kann das Bild insgesamt modifiziert werden:

- **Zuschneiden:** Entfernt Bildteile bis auf die Auswahl.
- **Bildgröße...:** Hier kann die Größe des Bildes geändert werden.
- **Horizontal / Vertikal / Diagonal spiegeln:** Spiegelt das Bild horizontal / vertikal entlang der vertikalen Achse bzw. diagonal über die Querachse.
- **Drehen 90 / -90 / 180 / ...:** Dreht das Bild um 90° im bzw. entgegen dem Uhrzeigersinn, bzw. um 180° im Uhrzeigersinn oder um eine benutzerdefinierte Gradzahl.
- **Auswahl ausfüllen:** Füllt die gesamte aktuelle Auswahl mit der gewählten Vordergrundfarbe.

4.2.3.2.6 Menü "Ebene"

Über dieses Menü können Ebene angelegt, modifiziert oder gelöscht werden. Beim Beenden des Java Image Editors werden mehrere Ebenen zu einer zusammengefasst, beim nächsten Start steht dann nur noch diese Ebene zur Verfügung.

- **Neue Ebene / durch Kopie / durch Ausschneiden:** Erzeugt eine neue leere Ebene, eine Ebene mit dem Inhalt der aktuellen Auswahl bzw. eine Ebene mit dem Inhalt der aktuellen Auswahl und entfernt die Auswahl aus der bestehenden Ebene.
- **Ebene löschen / Duplizieren:** Löscht bzw. dupliziert die aktuell gewählte Ebene.
- **Umbenennen...:** Mit diesem Menüeintrag kann der Name der Ebene geändert werden. Der neue Name wird beim Beenden des Editors nicht mit gespeichert



und steht beim nächsten Start des Java Image Editors nicht mehr zur Verfügung.

- **Nach oben / Nach unten:** Verschiebt die aktuelle Ebene um eine Position nach oben / unten.
- **Mit darunterliegender Ebene zusammenfassen:** Fasst die aktuell ausgewählte Ebene mit der darunterliegenden zusammen.
- **Horizontal / Vertikal / Diagonal spiegeln:** Spiegelt die ausgewählte Ebene horizontal / vertikal entlang der vertikalen Achse bzw. diagonal über die Querachse.
- **Drehen 90 / -90 / 180:** Dreht die ausgewählte Ebene um 90° im bzw. entgegen dem Uhrzeigersinn, bzw. um 180° im Uhrzeigersinn.

4.2.3.2.7 Menü "Filter"

Über dieses Menü können Filter auf das Bild, die aktuelle Ebene bzw. die aktuelle Auswahl angewendet werden.

- **Letzten Filter wiederholen:** Wendet den zuletzt angewendeten Filter noch einmal an.
- **Letzten Filter anzeigen:** Zeigt den zuletzt angewendeten Filter bzw. den entsprechenden Konfigurationsdialog an.

4.2.3.2.8 Menü "Fenster"

Über dieses Menü können die Paletten auf der rechten Seite aus- bzw. eingeblendet werden.

4.2.3.2.9 Paletten

- **Farben:** Hier kann die Vordergrundfarbe per Klick ausgewählt werden.
- **RGB-Farben:** Hier kann die Vordergrundfarbe per Angabe von RGB-Werten ausgewählt werden.
- **Farbfelder ("Colors"):** Hier kann die Vordergrundfarbe aus einer Bibliothek ausgewählt werden. Standardmäßig wird hier eine Palette von Farben verwendet, die plattformunabhängig von allen Browsern verwendet werden können ("Web-Farben"). Bei Bedarf können auch eigene Bibliotheken geladen (und später wieder abgespeichert) werden. Hochgeladene Farbbibliotheken werden beim Beenden des Editors nicht mit gespeichert und stehen beim nächsten Start des Java Image Editors nicht mehr zur Verfügung.
- **Farbverläufe ("Gradients"):** Hier kann ein Farbverlauf ausgewählt werden. Dabei können die bereits vorhandenen verwendet, bearbeitet oder eigene geladen werden. Hochgeladene oder bearbeitete Farbverläufe werden beim






Beenden des Editors nicht mit gespeichert und stehen beim nächsten Start des Java Image Editors nicht mehr zur Verfügung.

- **Ebenen ("Layers"):** Hier können neue Ebenen angelegt, bearbeitet oder gelöscht werden. Siehe dazu auch Kapitel 4.2.3.2.6 Seite 51. Darüber hinaus können Füllmethoden für die aktuelle Ebene eingestellt werden. Beim Beenden des Java Image Editors werden mehrere Ebenen zu einer zusammengefasst, beim nächsten Start steht dann nur noch diese Ebene zur Verfügung.

4.2.3.3 Speichern von bearbeiteten Bildern

Die Editoren stellen eigene Speicher-Funktionen zur Verfügung. Über diese müssen bearbeitete Bilder / Auflösungen zunächst gespeichert werden. Damit die Änderungen final in die Medien-Verwaltung übernommen werden, muss allerdings das gewählte Bild auch in der Medien-Verwaltung noch einmal gespeichert werden.

- Über  gestarteter Editor:
Änderungen an Bildern / Auflösungen müssen zunächst über das Icon  gespeichert werden. Das Bild / die Auflösung kann danach weiter bearbeitet werden. Final müssen die Änderungen über die Icons der Symbolleiste "Speichern" oder "In den Ansichts-Modus wechseln", "Bearbeiten an/aus" im Kontextmenü oder Strg + S bzw. Strg + E übernommen werden.
- Java Image Editor:
Änderungen an Bildern / Auflösungen müssen zunächst über das Icon  gespeichert werden. Das Bild / die Auflösung kann danach weiter bearbeitet werden. Final müssen die Änderungen über die Icons der Symbolleiste "Speichern" oder "In den Ansichts-Modus wechseln", "Bearbeiten an/aus" im Kontextmenü oder Strg + S bzw. Strg + E übernommen werden.
- Picnik:
Änderungen an Bildern / Auflösungen müssen zunächst über die Schaltfläche "In FirstSpirit speichern" gespeichert werden. Das Bild / die Auflösung kann danach weiter bearbeitet werden. Final müssen die Änderungen über die Icons der Symbolleiste "Speichern" oder "In den Ansichts-Modus wechseln", "Bearbeiten an/aus" im Kontextmenü oder Strg + S bzw. Strg + E übernommen werden.
- Pixlr:
Änderungen an Bildern / Auflösungen müssen zunächst über den Menüeintrag "Speichern..." im Menü "Datei" gespeichert werden. Das Bild / die Auflösung kann danach weiter bearbeitet werden. Final müssen die Änderungen über die



Icons der Symbolleiste "Speichern" oder "In den Ansichts-Modus wechseln", "Bearbeiten an/aus" im Kontextmenü oder Strg + S bzw. Strg + E übernommen werden.

4.2.4 Datenquellen-Verwaltung: Verhalten von gefilterten Datensätzen

Seit Version 4.2R2 können Datenquellen mithilfe verschiedener Optionen gefiltert und so die Anzeige von Datensätzen eingeschränkt werden. In diesem Kontext wurden mit Version 4.2R4 einige Optimierungen im Software-Verhalten vorgenommen.



War **bis 4.2R2** für das Anlegen von Filtern (Kontextmenü "Extras" / "Filter setzen" auf Datenquellen-Ebene) das Recht "Objekt anlegen" erforderlich, wird **ab 4.2R4** das Recht "Ordner anlegen" benötigt.

- **Filtereinstellungen wirken einschränkend:** Ab 4.2R4 werden die Suchergebnisse folgender Filtermöglichkeiten UND-verknüpft:
 - persistenter Filter über Kontextmenü "Extras" / "Filter setzen" auf der Datenquelle
 - temporärer Filter über den Link "Filter setzen" oberhalb der Tabelle
 - temporärer Filter über den Link "Suche" oberhalb oder das Icon "Datensuche" unterhalb einer Tabelle.

Durch die Wahl mehrerer Filter wird also das Suchergebnis immer weiter eingeschränkt. Zuvor wurde ein bestehender Filter durch das Auswählen eines anderen aufgehoben.

- **Angezeigte Zeilen freigeben:** Bis Version 4.2R2 einschließlich stand auf Datenquellen im Kontextmenü unter "Extras" die Funktion "Alle Zeilen freigeben" zur Verfügung (in Projekten, die mit Freigabe arbeiten). Mit dieser wurden alle Datensätze der betreffenden Datenquelle gleichzeitig freigegeben, aktuelle Filtereinstellungen wurden nicht berücksichtigt.
Mit Version 4.2R4 wurde diese Funktion durch die Funktion "Angezeigte Zeilen freigeben" ersetzt. Mit dieser werden jetzt nur noch die Datensätze der Datenquelle freigegeben, die in Abhängigkeit von den aktuellen Filtereinstellungen der Datenquelle angezeigt werden. Dabei werden alle zur Verfügung stehenden Filtereinstellungen berücksichtigt:
 - persistenter Filter über Kontextmenü "Extras" / "Filter setzen" auf der Datenquelle
 - temporärer Filter über den Link "Filter setzen" oberhalb der Tabelle
 - temporärer Filter über den Link "Suche" oberhalb oder das Icon



- "Datensuche" unterhalb einer Tabelle,
 - temporäre Anzahleinschränkung über die Combobox oberhalb der Tabelle.
- **Sprung zu gefilterten Datensätzen:** Soll z. B. aus einer Eingabekomponente, in der ein Datensatz referenziert ist, zu einem Datensatz in der Datenquellen-Verwaltung gesprungen werden, der aufgrund der aktuellen Filtereinstellungen nicht in der Übersicht angezeigt wird, wird folgender Meldung ausgegeben: "Der gewünschte Datensatz wurde in der aktuellen Sicht nicht gefunden. Möchten Sie eine Suche nach diesem Datensatz ausführen?".

4.2.5 Überarbeitung von Versionsvergleich und Übersetzungshilfe

Der Dialog für Versionsvergleiche (Kontextmenü-Eintrag "Versionshistorie" / Schaltfläche "Vergleichen" auf jedem Objekt) und die Übersetzungshilfe (Menü "Extras", Eintrag "Übersetzungshilfe" auf Seiten und Absätzen der Inhalte-Verwaltung, Datensätzen der Datenquellen-Verwaltung und auf Globalen Inhalten) ist in der Version 4.2R4 optisch überarbeitet worden.

Die Übersetzungshilfe wird über das Menü "Extras", Menüpunkt "Übersetzungshilfe" aufgerufen. In Version 4.2R4 ist dieser Menüpunkt nur noch auf den Knoten der Baumstruktur aktiv, auf denen die Übersetzungshilfe auch ausgeführt werden kann. Die Versionshistorie kann auf jedem Knoten der Baumstruktur aufgerufen werden, z. B. über Strg + H, die Statusleiste oder das Kontextmenü. Die Combobox "Bearbeitungsmodus" im Dialog zur Umschaltung zwischen Vergleichs- und Übersetzungsmodus ist in Version 4.2R4 entfallen, ebenso der Modus "Nachübersetzen".

Auf einem Absatz aufgerufen, kann der Dialog nun folgendermaßen aussehen:



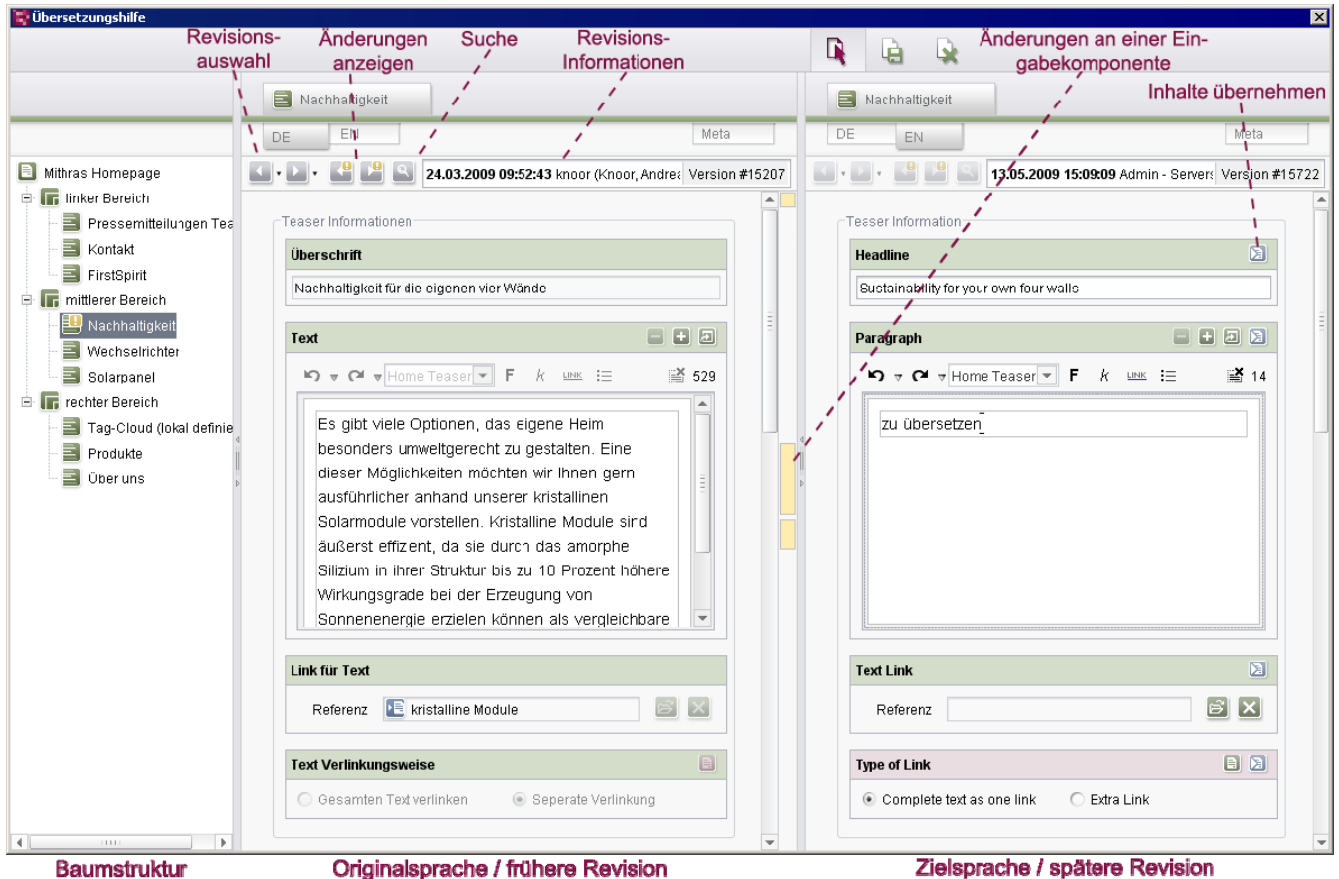


Abbildung 4-14: Versionshistorie / Übersetzungshilfe – Inhalte-Verwaltung

Die Baumstruktur wird nun, wie von anderen Dialogen gewohnt, auf der linken Seite in einem eigenen Fensterbereich dargestellt. In zwei weiteren Fensterbereichen werden die Eingabekomponenten des ausgewählten Objektes angezeigt, jeweils in den zur Verfügung stehenden Sprachen. Für Übersetzungen wird links die Original- und rechts die Zielsprache in der letzten Revision des Objektes, für Vergleiche werden hier die zu vergleichenden Revisionen angezeigt.

Zwischen den Fensterbereichen zeigt eine Leiste an, in welcher Eingabekomponente Änderungen vorhanden sind. Dabei zeigt gelbe Markierung an, dass Daten in der Eingabekomponente geändert wurden, rote Markierung, dass Daten entfernt wurden und grün, dass neue Werte hinzugekommen sind. Ein Tooltip zeigt den Namen der betreffenden Eingabekomponente an und mit einem Klick kann direkt zu der betreffenden Eingabekomponente gesprungen werden.


In der Baumstruktur wird mithilfe von Icons visualisiert, ob und welche Art von Unterschieden es zwischen den verglichenen Versionen gibt. Dabei zeigt z. B. eine gelbe Ausrufungszeichen an, dass es Änderungen hinsichtlich der Inhalte des jeweiligen Objekts (z. B. Absatz oder Seite) gibt, ein grünes Plus zeigt an, dass ein



Knoten hinzugekommen ist, ein rotes Kreuz, dass ein Knoten gelöscht wurde. Innerhalb der Eingabekomponenten werden Änderungen zwischen beiden Versionen durch eine farbige Titelzeile visualisiert. Hinzugekommener Text in DOM-Editor und DOM-Tabelle wird dabei grün hinterlegt, Text, der entfernt wurde, rot hinterlegt dargestellt.

Oberhalb der Eingabekomponenten wird die jeweilige Revision mit der Versionsnummer, Datum, Uhrzeit, letztem Bearbeiter und dem Kommentar, der zu der Revision vergeben wurde, angezeigt ("Revisionsinformationen"). Mithilfe der Vor- und Zurück-Icons kann zur vorherigen oder nächsten Revision ("Revisionsauswahl") bzw. zur vorherigen oder nächsten Änderung ("Änderungen anzeigen") an dem jeweiligen Objekt gewechselt werden.

In den Fensterbereichen werden für **Vergleiche** zwei verschiedene Versionen eines Objekts in derselben Sprache dargestellt. Im Vergleichsmodus können keine Inhalte bearbeitet werden.

Für **Übersetzungen** können bereits vorhandene Inhalte der Originalsprache mithilfe des Icons  in die jeweiligen Eingabekomponenten der Zielsprache übernommen werden, um auf diese Weise eine direkte Übersetzungsvorlage zu haben. Die Übersetzung kann dann in diesem Dialog oder wie gewohnt im betreffenden Absatz bzw. der betreffenden Seite im Arbeitsbereich des JavaClients vorgenommen werden. Für jedes Betätigen des "Speichern"- (bzw. Strg + S) oder "In den Ansichts-Modus wechseln"-Icon (Strg + E) wird eine neue Revision mit dem Kommentar "Übersetzt" erzeugt. Inhalte der Originalsprache können nicht bearbeitet werden.



4.2.6 Versionshistorie: Neue Option "Teilarchivierte Revisionen einblenden"

Im Dialog der Versionshistorie (Menüpunkt "Versionshistorie" im Kontextmenü auf jedem Knoten) gibt es nun eine neue Option "Teilarchivierte Revisionen einblenden":

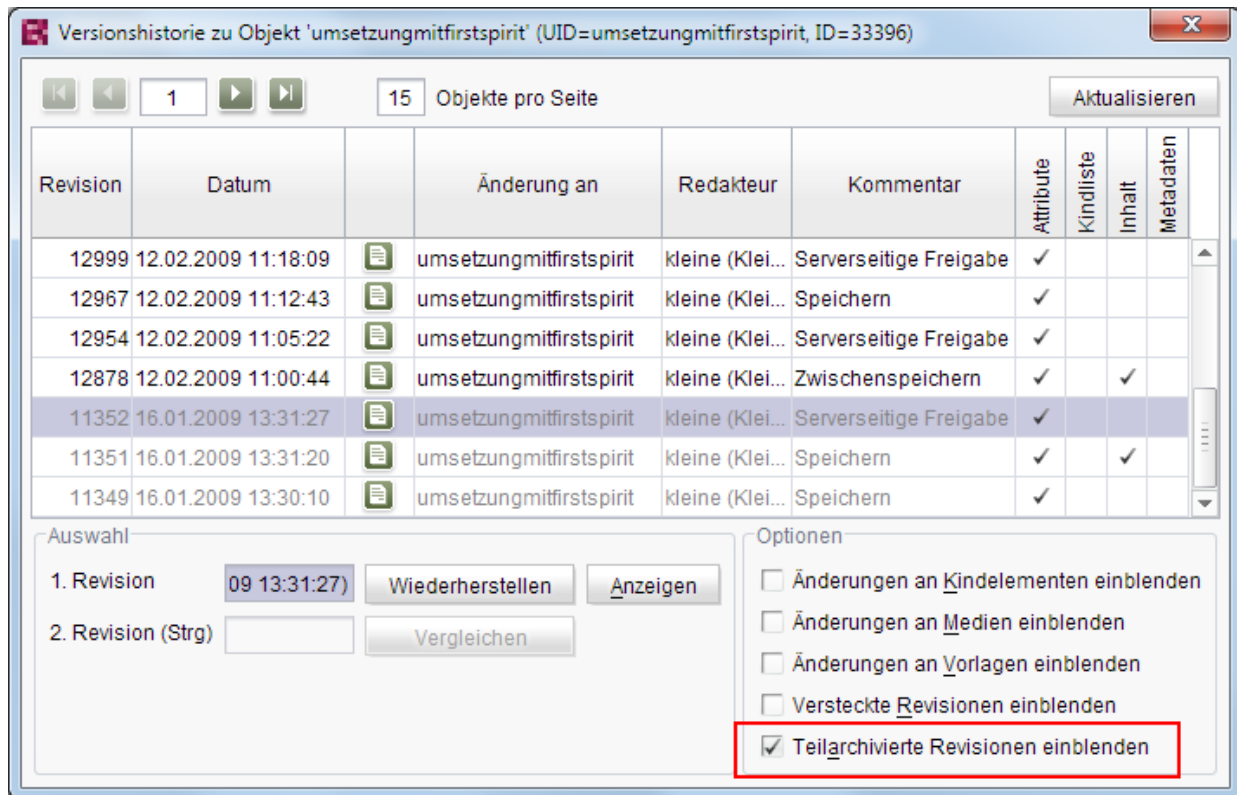


Abbildung 4-15: Versionshistorie mit neuer Option

Nach einer Archivierung (siehe *FirstSpirit Handbuch für Administratoren*, Kapitel "Alte Projektstände archivieren") werden so genannte "teilarchivierte Revisionen" in der Versionshistorie standardmäßig nicht angezeigt. Teilarchivierte Revisionen und versteckte Revisionen konnten bislang mit der Option "Versteckte Revisionen einblenden" eingeblendet werden. Um eine intuitivere Benutzung zu ermöglichen, wurde die neue Option "Teilarchivierte Revisionen einblenden" eingeführt. Mit dieser Option werden alle Revisionen von Objekten eingeblendet, die noch vollständig erhalten sind, aber deren Revisionsnummer kleiner ist als die der kleinsten/letzten noch nicht archivierten Revision. Dies kann z. B. der Fall sein, wenn bei einer Archivierung Seiten aus der Inhalte-Verwaltung archiviert werden, nicht aber die zugrundeliegenden Seitenvorlagen (dies ist über einen entsprechenden Auftrag zu erreichen, in dem die Optionen "Inhalte, Medien und Datenquellen" aktiviert, die Optionen "Vorlagen" und "Systemdaten" deaktiviert sind).



Teilarchivierte Revisionen werden bei Aktivierung der Option in der Tabelle grau dargestellt. Auf ihnen können mithilfe der betreffenden Buttons die Funktionen "Wiederherstellen", "Anzeigen" und "Vergleichen" durchgeführt werden.

4.2.7 Änderungen bei Arbeitsabläufen und Aufgaben

Aus Performance-Gründen werden in der Aufgabenliste (Strg + T oder Menü "Aufgaben" / "Aufgabenliste") auf den Registern "Offene Aufgaben" und "Initiierte Aufgaben" ab 4.2R4 initial nur 25 Aufgaben angezeigt. Liegen mehr Aufgaben vor, können diese über den Button **Ältere Aufgaben anzeigen** eingeblendet werden (siehe Abbildung 4-16). Auf der personalisierten Projekt-Einstiegsseite werden initial nur 10 Aufgaben angezeigt. Mit jedem Klick auf den Button "Ältere Aufgaben anzeigen" werden 10 weitere Aufgaben eingeblendet. Mit einem Klick auf "Alle Aufgaben anzeigen" öffnet sich die Aufgabenliste.

Ungültige Aufgaben, die z. B. dadurch entstehen, dass ein Objekt, auf dem ein Arbeitsablauf aktiv ist, gelöscht wird, werden ab 4.2R4 mit rotem Hintergrund in der Aufgabenliste visualisiert (siehe Abbildung 4-16). Diese können nicht weitergeschaltet, sondern nur über den Button "Aufgabe schließen" geschlossen werden. Kann die Aufgabe repariert werden, z. B. wenn das gelöschte Objekt, zu dem der Arbeitsablauf noch besteht, wiederhergestellt wird, wird der Button "Aufgabe reparieren" eingeblendet. Über diesen werden die Aufgabe, die Status-Farbe und der Schreibschutz zurückgesetzt.



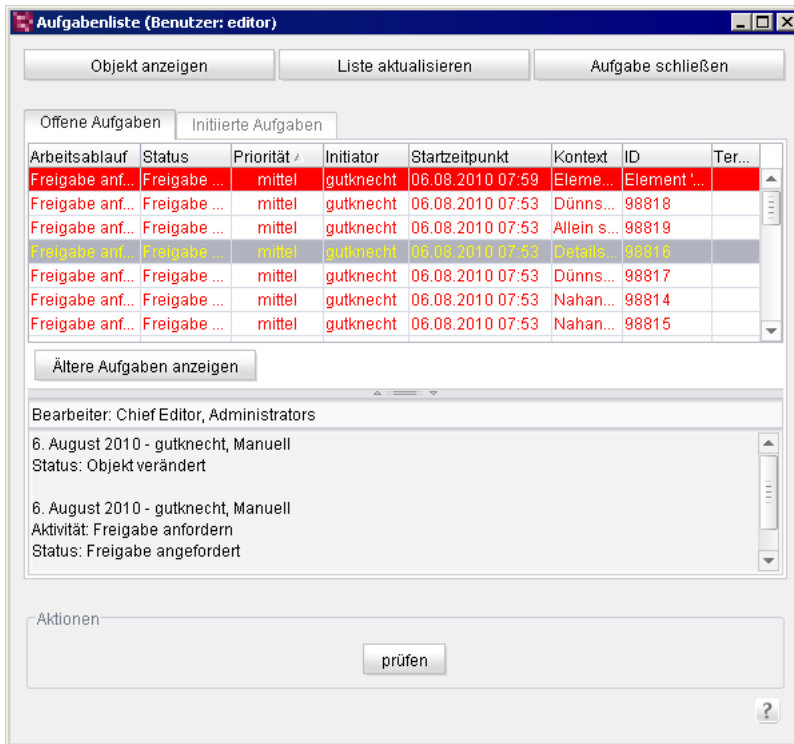


Abbildung 4-16: Aufgabenliste mit ungültigen Aufgaben



4.3 Neue/geänderte Funktionen im WebClient

WebEdit wurde in der Version 4.2R4 noch einmal optisch überarbeitet, bevor mit FirstSpirit Version 5.0 die Ablösung durch eine völlig neue Implementierung erfolgen wird. Darüber hinaus wurden einige neue Funktionen eingeführt. Die Benutzerführung ist im Großen und Ganzen unverändert geblieben.



Ab 4.2R4 wird nur noch das "xp"-Theme unterstützt. Bei diesem handelt es sich jetzt um das Default-Theme. Darüber hinaus werden keine Themes mit kundenspezifischen Einstellungen mehr unterstützt.



Die im JavaClient neu eingeführte Eingabekomponente **FS_BUTTON** (siehe Kapitel 4.1.2 Seite 30) wird aktuell im FirstSpirit-WebClient nicht unterstützt. Eine entsprechende Unterstützung ist für FirstSpirit Version 5 geplant. Die Eingabekomponente **FS_LIST** (siehe Kapitel 4.1.1 Seite 28) wird im WebClient in 4.2R4 nur rudimentär unterstützt: aktuell stehen nur die Funktionen der Eingabekomponente **CMS_INPUT_CONTENTAREALIST** (Liste zur Verwaltung von Absatzinhalten) zur Verfügung. Mehr Funktionen können über den sogenannten Applet-Modus bereitgestellt werden. Dies ist aber nur eine Übergangslösung, eine native **FS_LIST**-Unterstützung wird es ab FirstSpirit Version 5.0 geben.

4.3.1 Überarbeitete WebEdit Symbolleiste

Die in der WebEdit Symbolleiste verwendeten Icons wurden in Version 4.2R4 weiter an die vom FirstSpirit-JavaClient bekannten Icons angepasst. Die Beschriftung der Icons wurde beibehalten, befindet sich nun aber unterhalb der Icons:

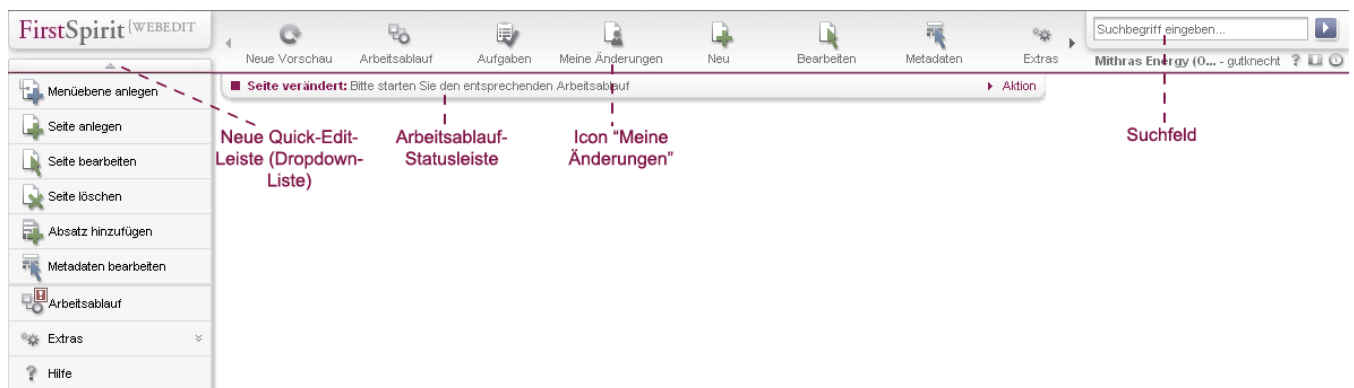


Abbildung 4-17: Überarbeitete WebEdit Symbolleiste



Darüber hinaus wurde die Symbolleiste um einige neue Funktionen erweitert, und zwar um

- **Scroll-Symbole:** Ist das WebEdit-Browserfenster kleiner als die WebEdit-Symbolleiste, kann die Symbolleiste über die Icons ◀ und ▶ nach rechts bzw. links verschoben (gescrollt) werden. Die Icons sind nur dann aktiv, wenn durch Scrollen nach links bzw. rechts weitere Funktionen der Symbolleiste angezeigt werden können.
- **eine Dropdown-Liste:** sie enthält die Funktionen der Quick-Edit-Leiste auf Seitenebene und soll diese in einer späteren Version ablösen (siehe Kapitel 4.3.2 Seite 62),
- **ein Suchfeld:** mit diesem kann das Projekt per Volltextsuche durchsucht werden (siehe dazu auch Kapitel 4.3.3 Seite 63),
- **das Icon "Meine Änderungen":** mit dieser Funktion kann sich der Benutzer alle Seiten des Projekts anzeigen lassen, die er erstellt oder an denen er Änderungen vorgenommen hat. Dabei wird auf die Funktion der **globalen Suche** zurückgegriffen (siehe dazu auch Kapitel 4.3.3 Seite 63),
- **eine Statusleiste für Arbeitsabläufe:** unterhalb der WebEdit Symbolleiste zeigt eine Statusleiste den aktuellen Arbeitsablauf-Status der jeweiligen Seite (inklusive Änderungen an Absätzen der Seite und an der entsprechenden Seitenreferenz) an. Mit einem Klick auf die Statusleiste kann der betreffende Arbeitsablauf direkt gestartet bzw. weitergeschaltet werden (siehe Kapitel 4.3.4 Seite 66).

Darüber hinaus wurden die Icons für die Sprachauswahl und die Hilfe an den rechten Rand unter das Suchfeld verschoben.

4.3.2 Neue Quick-Edit-Leiste auf Seitenebene

Ab Version 4.2R4 werden die Funktionen der Quick-Edit-Leiste von der neuen vertikalen Quick-Edit-Leiste übernommen. Dabei kann per Klick auf das Doppelpfeil-Icon ein Untermenü unterhalb von "Extras" geöffnet werden:



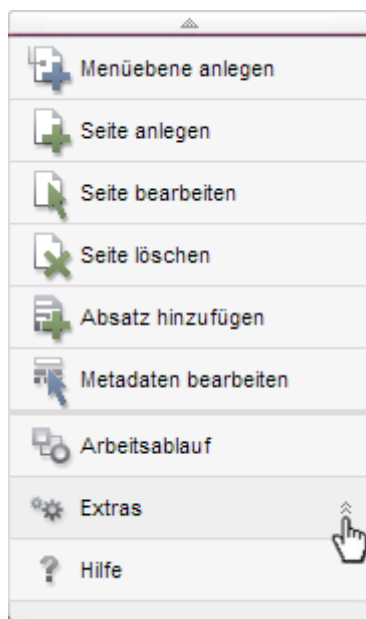



Abbildung 4-18: Vertikale Quick-Edit-Leiste

Es werden nur die Funktionen angezeigt, für die der jeweilige Benutzer die entsprechenden Rechte hat. Aus Gründen der Kompatibilität wird die bislang bekannte (horizontale) Quick-Edit-Leiste weiterhin unterstützt. Ab Version 5.0 wird sie nicht mehr unterstützt.

4.3.3 Globale Suche

Zur Volltextsuche innerhalb eines Projekts steht nun ein Suchfenster auf jeder Vorschau-Seite des WebClients zur Verfügung. Mit diesem können Struktur- und Inhalte-Verwaltung durchsucht werden. Suchergebnisse können für einen bestimmten Bearbeitungszeitraum gefiltert ausgegeben werden oder / und es können nur die vom aktuellen Benutzer erstellten oder geänderten Objekte angezeigt werden (über das Icon "Meine Änderungen" der WebEdit Symbolleiste, siehe Kapitel 4.3.1 Seite 61).

Dabei werden allerdings nur die Inhalte- und Struktur-Verwaltung durchsucht. Es werden in erster Linie Treffer aus der Struktur-Verwaltung ausgegeben. Nur wenn es zu einem Treffer keine Seitenreferenz gibt, wird die betreffende Seite aus der Inhalte-Verwaltung ausgegeben. Dabei werden neben Texten auf den Seiten des Projekts auch Metadaten und auf den Seiten referenzierte Medien durchsucht.

Die Suche wird über das Icon  oder <Enter> gestartet. Dabei kann das Feld auch leer bleiben. Auf diese Weise kann in Verbindung mit den zur Verfügung stehenden Filtern (z. B. zeitlich oder nach aktuellem Benutzer, siehe unten) z. B. eine Liste der



letzten Änderungen angefordert werden.

Unterhalb des Suchfensters wird die Trefferliste angezeigt:

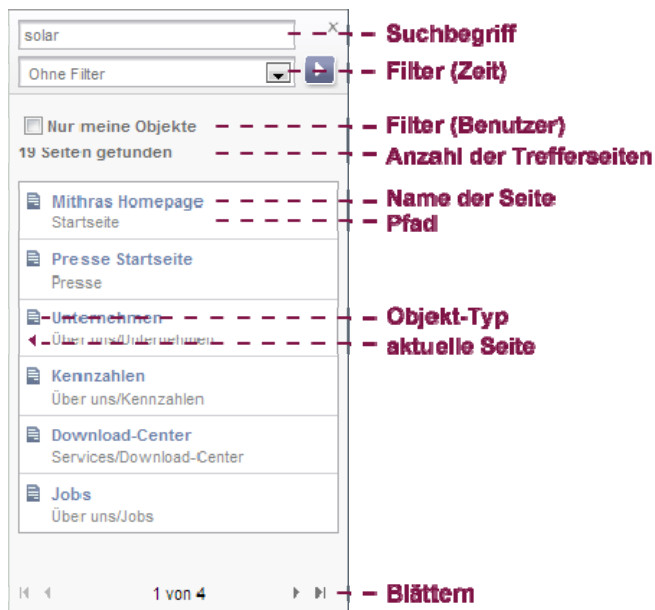



Abbildung 4-19: Trefferliste zum Suchbegriff "solar"

Über der Trefferliste wird die Anzahl der gefundenen Seiten ausgegeben. Werden keine Ergebnisse gefunden, wird der Text "0 Seiten gefunden" angezeigt.

Die Trefferliste hat eine fixe Breite. Werden mehr Objekte gefunden, als in der Höhe darstellbar sind, kann über die Icons unterhalb der Trefferliste zur ersten/vorherigen/nächsten/letzten Suchergebnis-Seite gewechselt werden.

Die Ergebnisse werden jeweils mit Objekt-Icon (in der Regel ) , Namen und Pfad angezeigt. Mit einem Klick auf ein Ergebnis wechselt die Ansicht der WebEdit-Vorschau zur Seite mit dem gesuchten Text.



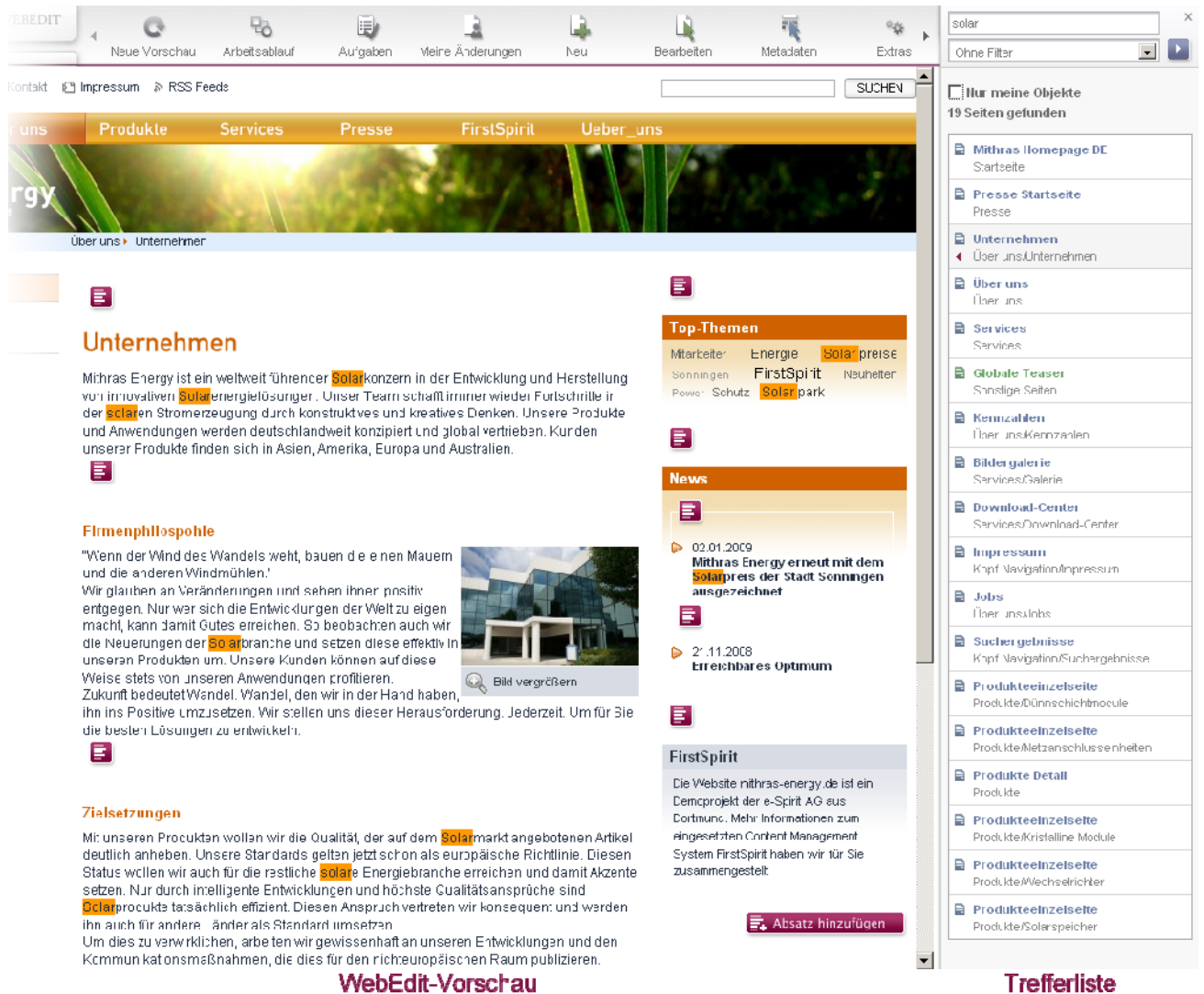





Abbildung 4-20: Suchergebnis-Seite in der Vorschau

Das Icon  in der Trefferliste zeigt an, dass sich der markierte Treffer auf der aktuellen Seite links in der WebEdit-Vorschau befindet. Der oder die Treffer werden in der Vorschau farbig hervorgehoben.

Filter: Das Suchergebnis kann mithilfe von Filtern zeitlich eingeschränkt werden. Wird aus der Dropdown-Liste der Filter "Geändert, letzte Woche" gewählt, werden nur Seitenreferenzen angezeigt, die innerhalb der letzten Woche angelegt oder geändert wurden. Wird der Filter "Geändert, letzten Monat" gewählt, werden nur Seitenreferenzen angezeigt, die innerhalb des letzten Monats angelegt oder geändert wurden. Diese Filter können mit dem Filter "Nur meine Objekte" (siehe unten) kombiniert werden. Zum Aktivieren des Filters muss das Such-Icon  angeklickt werden. Die Standard-Einstellung ist "Ohne Filter".



Nur meine Objekte: Über diese Checkbox kann das Suchergebnis auf Objekte eingeschränkt werden, die vom aktuell angemeldeten Benutzer angelegt oder geändert worden sind. Zum Aktivieren dieses Filters muss das Such-Icon  angeklickt werden. Standardmäßig ist diese Checkbox nicht aktiviert (Ausnahme: Anzeige von "Meine Änderungen", Kapitel 4.3.1 Seite 61).

Die Trefferliste bleibt so lange geöffnet, bis sie über das X-Icon rechts oben geschlossen oder eine neue Vorschau angefordert wird. Wurde ein Suchbegriff eingegeben, bleibt dieser so lange bestehen, bis er gelöscht wird, ein anderer Suchbegriff eingegeben oder eine neue Vorschau angefordert wird.

4.3.4 Statusleiste für Arbeitsabläufe

Um besser erkennen zu können, in welchem Arbeitsablauf-Status sich eine Seite befindet und welcher Schritt als nächstes ausgeführt werden muss, wurde unterhalb der WebEdit-Symboleiste eine Statusleiste für Arbeitsabläufe eingeführt.

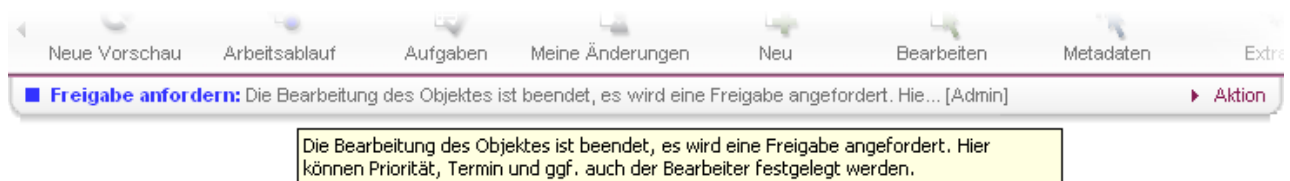


Abbildung 4-21: Neue Arbeitsablauf-Statusleiste

Hier wird angezeigt, welcher Schritt zur Bearbeitung des Arbeitsablaufs als nächstes erforderlich ist. Dieser wird in der jeweiligen Status-Farbe, die der Projektentwickler definiert hat, dargestellt (hier: "Freigabe anfordern"). Ebenfalls wird der Benutzer, der die Seite in den aktuellen Status geschaltet hat, angezeigt (hier: "Admin") sowie optional ein Kommentar des Benutzers, den er beim Weiterschalten eingetragen hat. Hat der Benutzer keinen Kommentar eingegeben, wird eine Beschreibung, die der Projektentwickler für den Status hinterlegt hat, angezeigt. Sollte der Text länger sein, als der zur Verfügung stehende Platz, kann der komplette Text einem Tooltip entnommen werden. Dieser wird eingeblendet, wenn mit dem Mauszeiger über den Kommentar bzw. Beschreibungstext gefahren wird.

An der rechten Seite der Statusleiste kann der Arbeitsablauf mit einem Klick auf "Aktion" weitergeschaltet werden. Es öffnet sich ein Dialogfenster, über das der nächste Arbeitsablauf-Schritt durchgeführt werden kann.

Detaillierte Informationen zu Arbeitsabläufen in WebEdit siehe FirstSpirit Handbuch für Redakteure (WebEdit), Kapitel "Arbeitsablauf starten / Arbeitsablauf Aktion".





Anders als in früheren Versionen, in denen die betreffenden Icons lediglich anzeigten, dass ein Arbeitsablauf auf der Seite gestartet wurde, wird nun durch die neue Statusleiste auch visualisiert, dass sie geändert wurde. Auch Änderungen an der betreffenden Seitenreferenz werden visualisiert.




Der Arbeitsablauf-Status wird darüber hinaus wie bisher über die Arbeitsablauf-Icons der WebEdit Symbolleiste und der bisherigen Quick-Edit-Leiste sowie der neuen Quick-Edit-Leiste auf Seitenebene (siehe Kapitel 4.3.2 Seite 62) angezeigt.

4.3.5 Überarbeitete Auswahldialoge

Folgende Eingabekomponenten verfügen in WebEdit über einen Auswahldialog, mit dessen Hilfe Referenzen zu FirstSpirit-Objekten erstellt werden können:

- Dateiauswahl (CMS_INPUT_FILE)
- Seitenreferenz (CMS_INPUT_PAGEREF)
- Bildauswahl (CMS_INPUT_PICTURE)
- Datensatzauswahl (CMS_INPUT_OBJECTCHOOSER, FS_DATASET)
- Referenzauswahl (FS_REFERENCE)
- Verweise, die eine Auswahl von FirstSpirit-Objekten ermöglichen (CMS_INPUT_LINK)

Diese Auswahldialoge werden über das "Öffnen"-Icon (z. B. ) geöffnet. In Version 4.2R4 wurde einerseits das Design überarbeitet, andererseits enthalten die Auswahldialoge nun auch – ähnlich wie im FirstSpirit-JavaClient – eine Suchmöglichkeit.



Ausnahme: Die Suche steht nicht für die Datenquellen-Verwaltung zur Verfügung.

Mit dieser Suche können die Verwaltungen, aus denen eine Referenz ausgewählt werden kann, durchsucht werden, z. B. nach Bildern und Dateien aus der Medien-Verwaltung oder Seitenreferenzen aus der Struktur-Verwaltung. Diese Suchmöglichkeit erleichtert das Arbeiten mit größeren Projekten und umfangreichen Inhalten.



Für die Eingabekomponente CMS_INPUT_FILE sieht der Auswahldialog mit Suche z. B. folgendermaßen aus:

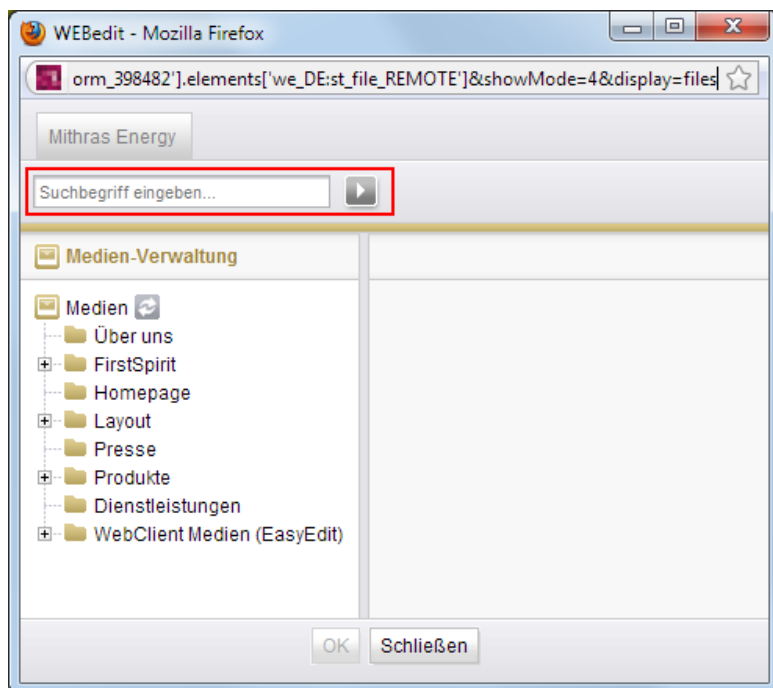


Abbildung 4-22: Suchfeld im Datei-Auswahldialog

Im Folgenden wird das Arbeiten mit der Suche exemplarisch für die Eingabekomponente CMS_INPUT_FILE vorgestellt. In den anderen Eingabekomponenten funktioniert die Suche im Wesentlichen analog.

Mit dem Suchfeld im Datei-Auswahldialog kann die Medien-Verwaltung des jeweiligen Projekts per Volltextsuche nach Dateien durchsucht werden. Dabei werden auch redaktionelle Inhalte einbezogen, z. B. Texte in PDF- oder Word-Dateien.

Nach Eingabe des gesuchten Begriffs kann die Suche mit dem Pfeil-Icon oder <Enter> gestartet werden. Im folgenden Dialog werden die Suchergebnisse in der linken Spalte angezeigt. Mit einem Klick auf ein Ergebnis werden in der linken Spalte die Detailinformationen eingeblendet:



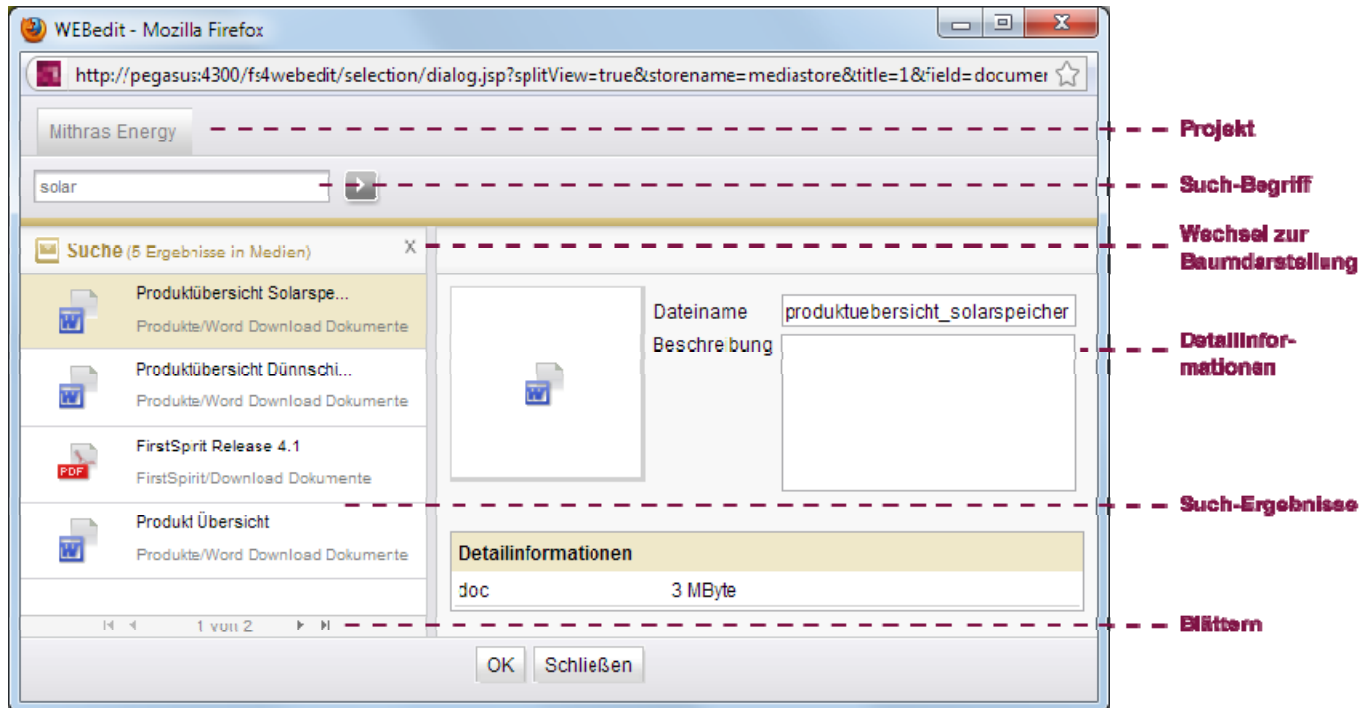


Abbildung 4-23: Such-Ergebnisse

Die Ergebnisse werden mit Anzeigenamen, Dateisymbol (z. B. Word- oder PDF-Symbol) und Pfadangabe aufgelistet. Oberhalb der Ergebnisliste wird die Anzahl an Such-Ergebnissen dargestellt. Auf der rechten Seite des Dialogs werden die Detailinformationen zum gewählten Objekt angezeigt.

Liegen mehr Ergebnisse vor, als aufgrund der eingestellten Höhe des Dialogs angezeigt werden können, kann mithilfe der Pfeil-Icons unterhalb der Ergebnisliste zu weiteren Ergebnisseiten gewechselt werden.

Mit einem Klick auf das x-Icon neben der Suchergebnis-Zahl wird statt der Ergebnisliste die Baumstruktur der Medien-Verwaltung dargestellt. Diese ist bis zu der Ebene aufgeklappt, an der sich das zuvor markierte Medium befindet. Diese Darstellung kann auch ohne Auslösen der Suche zur Navigation durch die Medien-Verwaltung verwendet werden. Wurde dieses Icon betätigt, kann nicht mehr zur Suchergebnis-Liste gewechselt werden, die Suche muss bei Bedarf erneut gestartet werden.

Mit "OK" kann die gewählte Datei in die Eingabekomponente übernommen werden.

Ermöglicht eine Eingabekomponente die Auswahl aus **mehreren Verwaltungen** (z. B. FS_REFERENCE oder interne Verweise), werden alle Verwaltungen durchsucht. Über Schaltflächen am unteren Ende des Dialogs kann zwischen den Ergebnissen der beiden Verwaltungen gewechselt werden. Dort wird auch die



jeweilige Trefferanzahl dargestellt:



Abbildung 4-24: Suchergebnisse in der Struktur- und Medien-Verwaltung

Ist für die Eingabekomponente der Zugriff auf ein oder mehrere **Remote-Projekte** konfiguriert, werden die Remote-Projekte als weitere Register neben dem Register des lokalen Projekts (in Abbildung 4-22: "Mithras Energy"). Soll die Medien-Verwaltung eines Remote-Projekts durchsucht werden, muss dazu auf das jeweilige Register gewechselt werden und die Suche von dort gestartet werden.

Detaillierte Informationen zur Arbeit mit diesen Eingabekomponenten in WebEdit siehe FirstSpirit Handbuch für Redakteure (WebEdit), Kapitel "Die Standard-Eingabelemente", speziell Unterkapitel zur Eingabekomponente CMS_INPUT_FILE, "Eine bestehende Datei aus der Medien-Verwaltung auswählen".

4.3.6 Überarbeitete "Bearbeiten"- und "Neu"-Dialoge

Einige Dialoge, z. B. zum Anlegen oder Bearbeiten von Menüebenen, Seiten oder Absätzen, sind in Version 4.2R4 geringfügig überarbeitet worden. Die Benutzerführung ist dabei aber prinzipiell unverändert geblieben.

"Bearbeiten"-Dialoge, die zuvor eine zweistufige Auswahl des zu bearbeitenden Elements erforderten, enthalten nun nur noch eine Dropdown-Liste, aus der das gewünschte Element direkt ausgewählt werden kann, z. B.



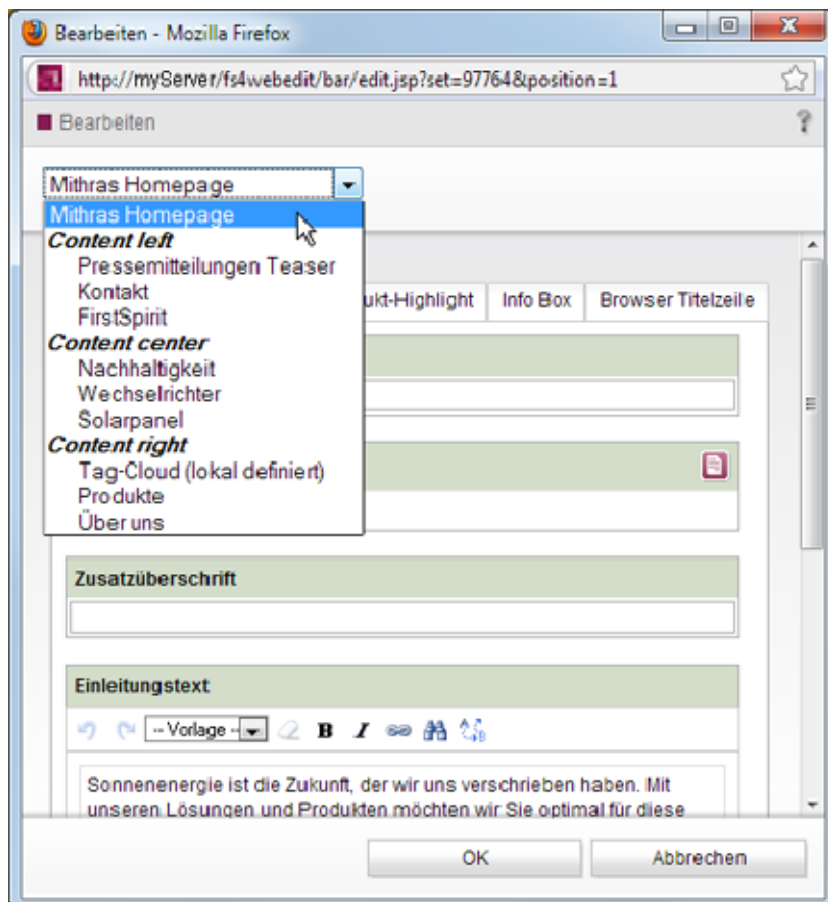


Abbildung 4-25: Dialog "Bearbeiten"

Wurde bisher zunächst ausgewählt, ob die Eingabekomponenten der Seite oder der Absätze bearbeitet werden sollen und dann der gewünschte Absatz, wird nun über die Dropdown-Liste die Seite oder ein bestimmter Absatz direkt ausgewählt. Im unteren Bereich des Dialogfensters werden alle für die über die Dropdown-Liste gewählte Seite oder den gewählten Absatz zur Verfügung stehenden Eingabekomponenten angezeigt. Auch für Datensätze wird diese Auswahl analog genutzt.

Dies betrifft die Dialoge, die sich über die Icons

- "Bearbeiten" und "Metadaten" in der WebEdit-Symbolleiste
- "Seite bearbeiten" in der Quick-Edit-Leiste

öffnen, sowie den Folgedialog bei Wahl des Icons

- "Neu" in der WebEdit-Symbolleiste.



Beispiel Dialog "Metadaten":

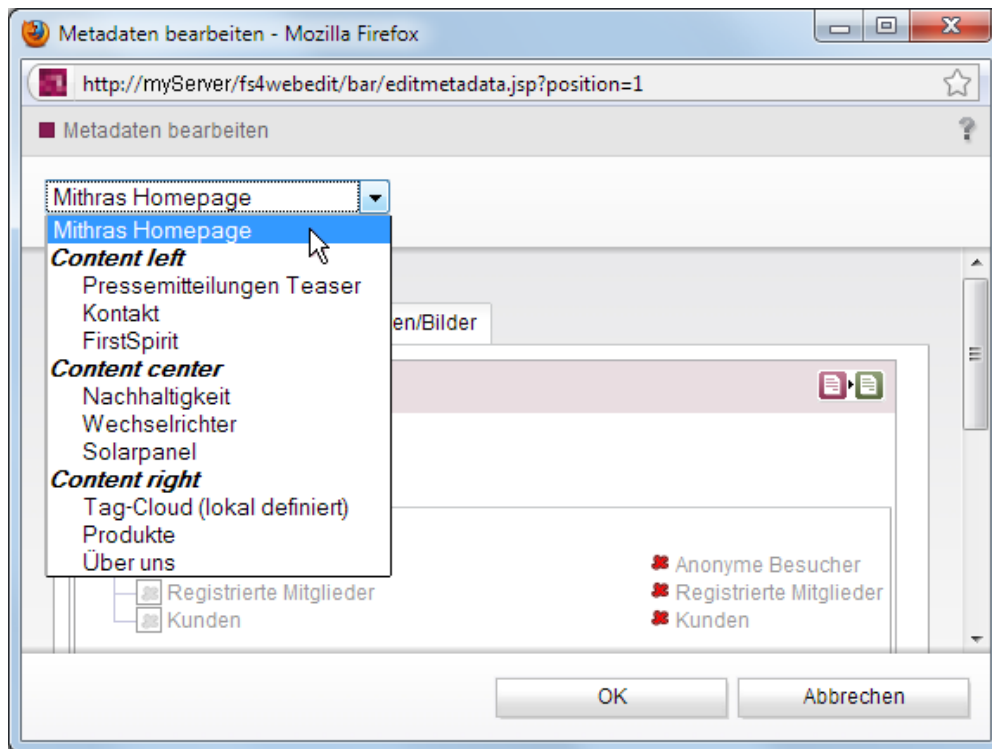


Abbildung 4-26: Dialog "Metadaten"

Die "Neu"-Dialoge, die über die Quick-Edit-Leiste (horizontal oder vertikal), geöffnet werden, wurden hinsichtlich des Layouts etwas verändert und bieten nun eine komfortablere Auswahl der gewünschten Vorlage, z. B.



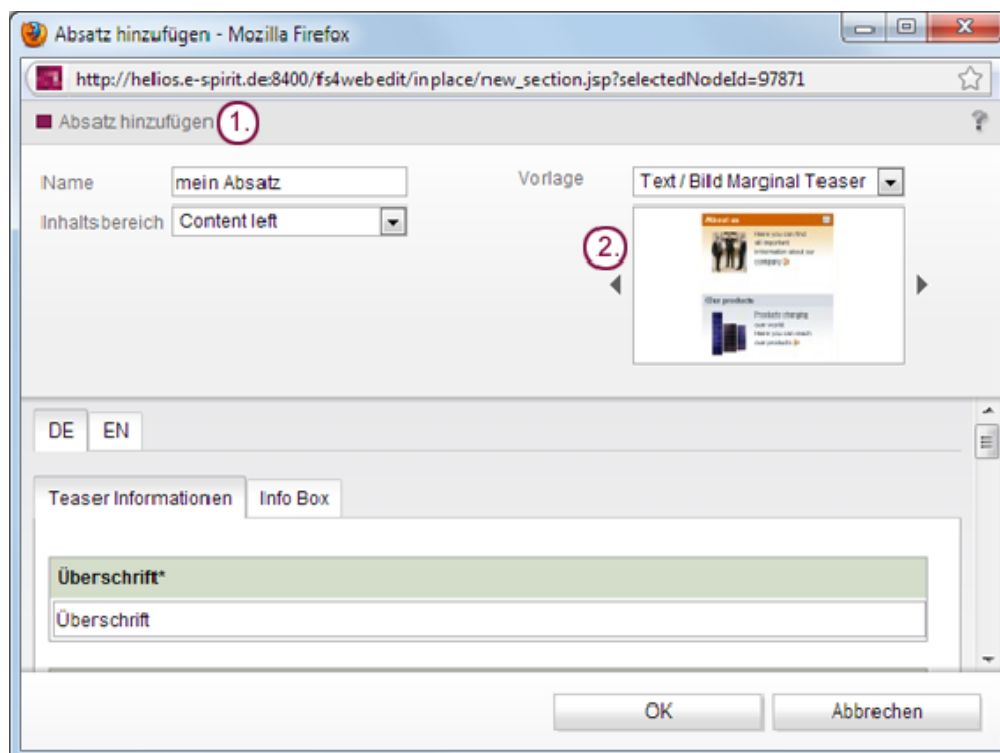


Abbildung 4-27: Dialog "Absatz hinzufügen"

1. Oben links wird nun immer die Aktion bzw. der Dialogname dargestellt.
2. Über die Scroll-Symbole kann zwischen den zur Verfügung stehenden Vorlagen anhand des Vorschaubildes (soweit vom Vorlagen-Entwickler für die jeweilige Vorlage hinterlegt) ausgewählt werden. Im unteren Bereich werden direkt die zugehörigen Eingabekomponenten eingeblendet. Alternativ kann die Vorlage wie bisher auch über die Dropdown-Liste nach Name ausgewählt werden.

Dies betrifft die Dialoge, die über

- die Icons / Schaltflächen "Menüebene anlegen", "Seite anlegen" und "Absatz hinzufügen" in der Quick-Edit-Leiste (horizontal und vertikal)
- die Easy-Edit-Funktion "Absatz hinzufügen"

geöffnet werden.

Wird das Icon "Menüebene anlegen" in der Quick-Edit-Leiste gewählt, werden Menüebene und Seite in einem zweistufigen Dialog angelegt. Der aktuelle Schritt wird dabei nun am oberen Rand des Dialogs visualisiert:



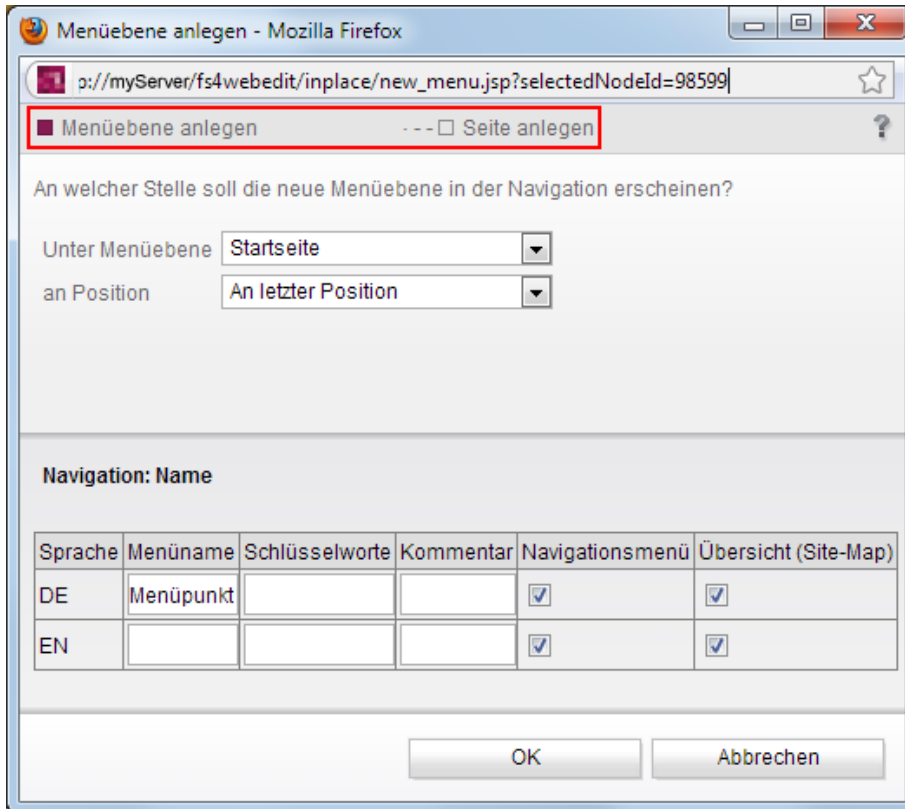


Abbildung 4-28: Dialog "Menüebene anlegen"

4.3.7 Überarbeiteter DOM-Editor

Aufgrund der in 4.2R4 neu eingeführten Kompatibilität mit dem Mozilla Firefox Version 3.x musste aus technischen Gründen auch das Design der Eingabekomponente DOM-Editor leicht verändert werden. Davon sind in erster Linie die Icons betroffen, ihre Funktion ist aber grundsätzlich unverändert geblieben:



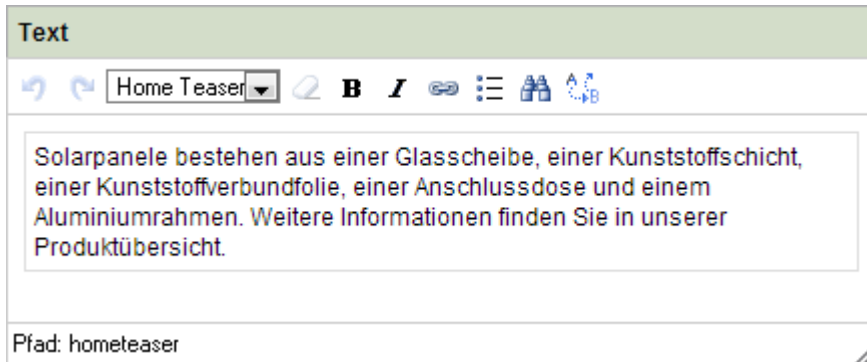



Abbildung 4-29: Überarbeiteter DOM-Editor

Neu dagegen ist, dass nun **Listenpunkte** über den Kontextmenü-Eintrag "Listenpunkt einrücken" eingerückt werden können, so dass geschachtelte Listen erzeugt werden können.

Darüber hinaus kann mit der ebenfalls neuen **Suchen/Ersetzen-Funktion** der Text im jeweiligen DOM-Editor durchsucht und bei Bedarf ersetzt werden. Mit einem Klick auf das Icon  wird dazu das Dialogfenster für eine Volltextsuche geöffnet:

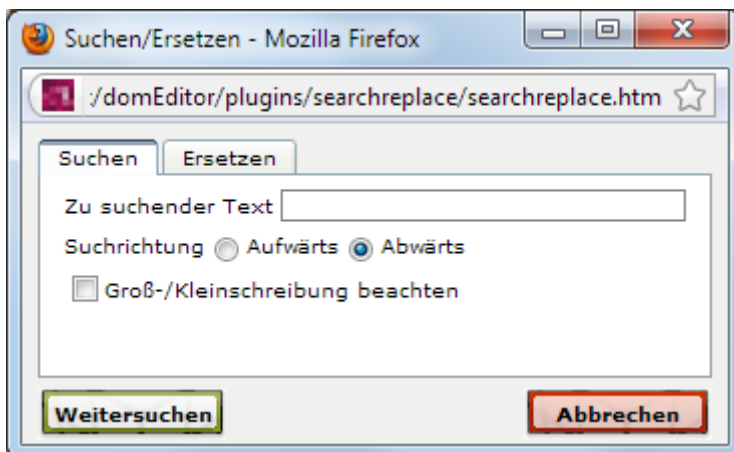




Abbildung 4-30: Such-Dialog

Der Text des DOM-Editors wird immer nur in die Richtung durchsucht, wie bei der Option Suchrichtung angegeben ist. Soll gesuchter Text durch einen anderen ersetzt werden, kann dazu auf das Register "Ersetzen" gewechselt werden. Wählt man das Icon , öffnet sich das gleiche Dialogfenster wie bei . Allerdings wird in diesem Fall initial das Register "Ersetzen" angezeigt.

Bei Verwendung des Rechtschreibprüfung-Moduls im DOM-Editor steht nun zusätzlich die Funktion **Wort ignorieren** auf unbekanntem (rot markierten) Wörtern im Kontextmenü zur Verfügung. Wird diese Funktion gewählt, wird die




Rechtschreibprüfung nur für das aktuelle Vorkommen des Worts in der Eingabekomponente ignoriert, die rote Markierung wird entfernt. Andere Vorkommen im Editor desselben (unbekannten) Wortes werden weiterhin rot markiert. Das markierte Wort wird nicht ins Wörterbuch übernommen, bei einem erneuten Aktivieren der Rechtschreibprüfung wird es erneut markiert.

Einige Funktionen, die bisher über Icons oder Tastaturkürzel ausgeführt werden konnten, können nun über ein **Kontextmenü** erreicht werden:

- Ausschneiden, Kopieren und Einfügen von markiertem Text
- Listenpunkt ein- oder ausrücken, Liste bearbeiten
- Verweis bearbeiten oder löschen

4.3.8 Datums-/Zeitauswahl mit Sekundenangabe über Kalenderansicht

Ab Version 4.2R4 können nun im WebClient in der Eingabekomponente zur Datums- und Zeitauswahl (CMS_INPUT_DATE) Sekunden auch über die Kalenderansicht eingegeben werden, die sich mithilfe des Icons  öffnet:

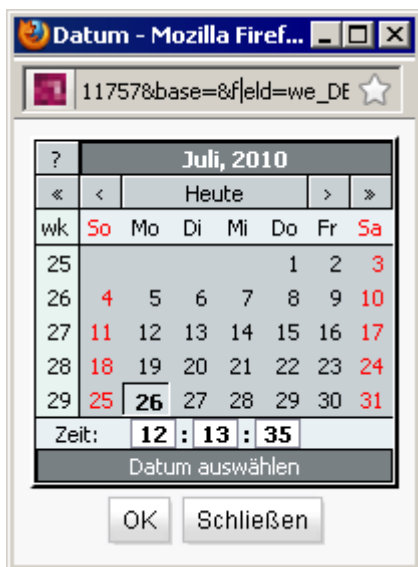


Abbildung 4-31: Kalenderansicht mit Sekundenfeld

4.3.9 Umstellung der WebEdit-Hilfe

An vielen Stellen in der WebEdit-Oberfläche kann über einen Hilfe-Button oder <F1> die kontextsensitive WebEdit-Hilfe aufgerufen werden. Mit Version 4.2R4 öffnet sich die Hilfe nicht mehr wie bisher in einem separaten Fenster, sondern es wird das *FirstSpirit Handbuch für Redakteure (WebEdit)* im PDF-Format geöffnet, immer in



der aktuellsten Version und in der Sprache, die auf der FirstSpirit-Startseite gewählt ist. Je nach Kontext, in dem die Hilfe aufgerufen wird, wird direkt das passende Kapitel dargestellt.



5 Neue/geänderte Funktionen für Vorlagenentwickler

In Version 4.2R4 wurden das FirstSpirit Komponenten- und Modul-Modell großflächig überarbeitet ("Refactoring") und flexibler gestaltet. Dabei wurde u.a. die technische Basis aller Eingabekomponenten ausgetauscht. In diesem Zusammenhang wurde auch das *FirstSpirit Entwicklerhandbuch für Komponenten* überarbeitet. Darüber hinaus ermöglicht das FirstSpirit AppCenter (siehe Kapitel 3 Seite 18) jetzt die Entwicklung und Integration eigener Applikationen für den JavaClient.

Neben dem Zugriff auf die bereits bekannte FirstSpirit Access-API wird in diesem Rahmen nun auch der Zugriff auf die so genannte Developer-API und die Applikationsintegrations-API ("Application-API") ermöglicht. Der Unterschied zwischen Access- und Developer-API ergibt sich hauptsächlich aus verschiedenen Stabilitätsauflagen: Die **Access-API** unterliegt höheren Stabilitätsauflagen, u.a. ist sie innerhalb einer Major-Versionslinie stabil; bevor Änderungen an Methode, Klassen und Funktionen vorgenommen werden, wird dies in einer Minor-Version zuvor angekündigt (*deprecation*). In der **Developer-API** können Änderungen hingegen bereits bei einem Minor-Versionswechsel durchgeführt werden. Die beiden API-Dokumentationen sind an unterschiedlichen Farben zu erkennen: Access-API = blau, Developer-API = brombeer. Die **Application-API** hingegen ist der wichtigste Einstiegspunkt in die Applikationsintegration. Über diese API können Applikations-Tabs im Applikationsbereich des FirstSpirit-JavaClients gesteuert werden. Die Application-API ist Bestandteil der Developer-API.

Siehe dazu auch FirstSpirit Online Dokumentation, Bereich "Vorlagenentwicklung" / "FirstSpirit API" sowie Kapitel 5.13 ab Seite 115.

Mit FirstSpirit Version 4.2 hat eine grundlegende Überarbeitung und Konsolidierung des Eingabekomponenten-Modells von FirstSpirit begonnen, die in 4.2R4 weitgehend abgeschlossen ist. Die in diesem Rahmen neu eingeführten Eingabekomponenten mit dem Prefix *FS_* haben dabei die Funktionen bereits bestehender Eingabekomponenten übernommen. Die Vorteile der neuen Eingabekomponenten sind, dass nun oft nur noch eine Eingabekomponente für mehrere Anwendungsfälle eingesetzt werden kann, die Zahl der unterschiedlichen Seiten- und Absatzvorlagen im Projekt kann dadurch verringert werden. Soll dem Redakteur z. B. die Auswahl von Bildern **und** PDF-Dateien ermöglicht werden, ist nun statt zweier Eingabekomponenten `CMS_INPUT_PICTURE` und `CMS_INPUT_FILE` nur noch `FS_REFERENCE` erforderlich.



Sollen Verwendungen von CMS_INPUT_PICTURE und / oder CMS_INPUT_FILE in einem Projekt durch FS_REFERENCE ersetzt werden, ist dazu eine Migration der betreffenden Formulare und Ausgabekanäle erforderlich. Wie Kapitel 5.1 Seite 80 zeigen wird, kann die Eingabekomponente FS_REFERENCE dabei je nach Bedarf so konfiguriert werden, dass sie (nur) genau dieselben Funktionen bietet wie CMS_INPUT_PICTURE oder CMS_INPUT_FILE. Bei einer Umstellung auf FS_REFERENCE sollten aber auch die neue Möglichkeiten, die diese Eingabekomponente bietet, in Betracht gezogen werden: Können z. B. zwei Vorlagen mit jeweils einem CMS_INPUT_PICTURE- und einem CMS_INPUT_FILE-Formular zu einer Vorlage mit einem FS_REFERENCE-Formular zusammengefasst werden? Ein weiterer Vorteil ist, dass die Syntax konsolidiert wurde und damit jetzt logischer aufgebaut ist.

Im folgenden Kapitel wird dargestellt, welche Schritte notwendig sind, um ein Projekt auf die Verwendung von FS_DATASET bzw. FS_REFERENCE umzustellen. Dabei wird zunächst gezeigt, wie die jeweiligen Komponenten in der Standard-Konfiguration aussehen, welche Funktionsunterschiede grundsätzlich bestehen und welche Parameter angepasst werden müssen.



5.1 Eingabekomponenten: Migration zu FS_REFERENCE

Für FirstSpirit-Version 5.0 ist geplant, dass, wie bereits in Version 4.2 angekündigt, die Eingabekomponenten

- CMS_INPUT_PICTURE
- CMS_INPUT_FILE
- CMS_INPUT_PAGEREF

The image displays three separate input component windows. The first window, titled 'PICTURE', contains a 'Referenz' field with a small image icon and the text 'image', a 'Kommentar' text area, and four input fields for 'hAbstand', 'vAbstand', 'Breite' (with value '723'), and 'Höhe' (with value '326'), plus a 'Rand' field. The second window, titled 'FILE', has a 'Referenz' field with a document icon and the text 'Word-Dokument', and a 'Kommentar' text area. The third window, titled 'PAGEREF', has a 'Referenz' field with a document icon and the text 'Seite 1'.

Abbildung 5-1: CMS_-Komponenten – Standardkonfiguration

entfallen und durch FS_REFERENCE

The image shows the 'FS_REFERENCE' input component window. It features a 'Referenz' field with an image icon and the text 'image'. Below this field is a detailed information box containing the following text: 'Status: Freigegeben (Barbara Gutknecht)' and 'Letzte Veränderung: 08.09.2010 09:24:38 (Barbara Gutknecht)'.

Abbildung 5-2: FS_REFERENCE – Standardkonfiguration

ersetzt werden.

Die folgenden Kapitel zeigen, was bei einer Umstellung zu beachten ist. Die Beispiele beziehen sich nur auf den Fall, in dem FS_REFERENCE exakt die Funktionalität der jeweiligen CMS_INPUT_-Eingabekomponenten übernehmen soll. FS_REFERENCE bietet darüber hinaus natürlich noch viele weitere Möglichkeiten.



5.1.1 Funktionsunterschiede FS_REFERENCE und entsprechende CMS_INPUT_-Eingabekomponenten

Die Eingabekomponenten CMS_INPUT_PICTURE, CMS_INPUT_FILE und CMS_INPUT_PAGEREF ermöglichen standardmäßig die Auswahl jeweils eines FirstSpirit-Objekt-Typs (Bilder, Dateien oder Seitenreferenzen). Der jeweilige Auswahldialog zeigt standardmäßig nur die Verwaltung an, aus der eine Referenz ausgewählt werden kann, z. B. die Medien-Verwaltung im Falle von CMS_INPUT_PICTURE und CMS_INPUT_FILE und die Struktur-Verwaltung im Falle von CMS_INPUT_PAGEREF. FS_REFERENCE hingegen ermöglicht prinzipiell die Auswahl jedes beliebigen Objekt-Typs und der Auswahldialog zeigt standardmäßig alle Verwaltungen mit allen Ordnern. Soll in FS_REFERENCE die **Auswahl** auf bestimmte Objekt-Typen eingeschränkt werden, wird dies über die Tags FILTER / ALLOW gesteuert. Soll die **Anzeige im Auswahldialog** auf bestimmte Objekt-Typen, Verwaltungen und / oder Ordner eingeschränkt werden, wird dies über die Tags FILTER / HIDE und die Tags SOURCES / FOLDER gesteuert. Detaillierte Informationen zu diesen Tags siehe auch in der *FirstSpirit Online Dokumentation*, Bereich "Vorlagenentwicklung" / "Formulare" / "Eingabekomponenten (neu)" / "FS_REFERENCE".

Beispiel Standardkonfiguration CMS_INPUT_FILE:

```
<CMS_INPUT_FILE name="st_file">
  <LANGINFOS>
    <LANGINFO lang="*" label="Datei-Auswahl"/>
  </LANGINFOS>
</CMS_INPUT_FILE>
```

In der mittleren Spalte des Auswahldialogs werden nur Dateien aus der Medien-Verwaltung angezeigt, keine Ordner.



Die Abbildung über FS_REFERENCE sieht folgendermaßen aus:

```
<FS_REFERENCE name="st_file">
  <FILTER>
    <ALLOW type="file"/>
    <HIDE type="picture"/>
    <HIDE type="mediafolder"/>
  </FILTER>
  <LANGINFOS>
    <LANGINFO lang="*" label="Datei-Auswahl"/>
  </LANGINFOS>
  <PROJECTS>
    <LOCAL name=".">
      <SOURCES>
        <FOLDER name="root" store="mediastore"/>
      </SOURCES>
    </LOCAL>
  </PROJECTS>
</FS_REFERENCE>
```

Über SOURCES / FOLDER wird hier die Anzeige auf die Medien-Verwaltung (store="mediastore") eingeschränkt, über FILTER / HIDE wird die Anzeige auf Dateien eingeschränkt, indem Bilder und Medien-Ordner ausgeblendet werden (type="picture" und type="mediafolder") und über FILTER / ALLOW wird die Auswahl auf Dateien (type="file") eingeschränkt.

Beispiel Standardkonfiguration CMS_INPUT_PAGEREF:

```
<CMS_INPUT_PAGEREF name="st_pageref" hFill="yes">
  <LANGINFOS>
    <LANGINFO lang="*" label="Auswahl einer Seitenreferenz"/>
  </LANGINFOS>
</CMS_INPUT_PAGEREF>
```

In der mittleren Spalte des Auswahldialogs werden Menüebenen, Seitenreferenzen und Dokumentengruppen aus der Struktur-Verwaltung angezeigt, es können aber nur Seitenreferenzen und Dokumentengruppen ausgewählt werden.



Die Abbildung über FS_REFERENCE sieht folgendermaßen aus:

```
<FS_REFERENCE name="st_file" hFill="yes">
  <FILTER>
    <ALLOW type="pageref"/>
    <ALLOW type="documentgroup"/>
  </FILTER>
  <LANGINFOS>
    <LANGINFO lang="*" label="Auswahl einer Seitenreferenz"/>
  </LANGINFOS>
  <PROJECTS>
    <LOCAL name=".">
      <SOURCES>
        <FOLDER name="root" store="sitestore"/>
      </SOURCES>
    </LOCAL>
  </PROJECTS>
</FS_REFERENCE>
```

Über SOURCES / FOLDER wird hier die Anzeige auf die Struktur-Verwaltung (store="sitestore") eingeschränkt, über FILTER / ALLOW wird die Auswahl auf Seitenreferenzen (type="pageref") und Dokumentengruppen (type="documentgroup") eingeschränkt.

Die Umstellung der Formulare in den folgenden Kapiteln basieren grundsätzlich auf diesen Standardkonfigurationen.

5.1.2 Umstellung von bestehenden Formularen

Ab Version 4.2R4 kann die FS_REFERENCE-Eingabekomponente Werte übernehmen und verarbeiten, die mit den Eingabekomponenten CMS_INPUT_PICTURE, CMS_INPUT_FILE und CMS_INPUT_PAGEREF erfasst wurden. Die Umstellung der Formulare von "CMS_INPUT_"- auf FS_REFERENCE ist mithilfe weniger manueller Änderungen möglich.

Für die Migration muss dazu in der Syntax das CMS_INPUT_-Tag durch FS_REFERENCE ersetzt werden. Der Bezeichner der Eingabekomponente, der über den Parameter name angegeben ist, muss hingegen bestehen bleiben. Nur so kann gewährleistet werden, dass bereits durch den Redakteur eingepflegte Daten weiterhin gespeichert werden und bearbeitbar sind.

Beispiel:

```
<CMS_INPUT_PICTURE name="st_picture"...>
```

muss durch



```
<FS_REFERENCE name="st_picture"...>
```

ersetzt werden.

Die übrigen Tags und Parameter, die im Formular der CMS_INPUT_-Eingabekomponenten enthalten sind, sind bis auf die Fälle, die in Kapitel 5.1.3 Seite 84 beschrieben werden, auch für FS_REFERENCE gültig und können unverändert übernommen werden.

5.1.3 Ausnahmen

Einige Parameter der CMS_INPUT_-Eingabekomponenten werden in FS_REFERENCE nicht mehr unterstützt. Nur in wenigen Fällen ist eine entsprechende Abbildung bei der Umstellung auf FS_REFERENCE nicht möglich. Dieses Kapitel stellt dar, welche Parameter wie ersetzt werden müssen, um in FS_REFERENCE die bisherige Funktionalität beizubehalten. Wird der Variablenname, der über `name` definiert wurde (siehe Kapitel 5.1.2 Seite 83), beibehalten, können aber auch abweichende Funktionalitäten konfiguriert werden.

- **CMS_INPUT_PICTURE:** Der Parameter `lean`, mit dem die Anzeige der Eingabekomponente gesteuert wurde, wird nicht von FS_REFERENCE unterstützt und muss für eine erfolgreiche Migration ersatzlos aus dem Formular entfernt werden.
- **CMS_INPUT_FILE:** Der Parameter `allowFolder`, über den eine Auswahl von Ordnern ermöglicht wurde, wird von FS_REFERENCE nicht unterstützt. Der Parameter muss für eine erfolgreiche Migration aus dem Formular entfernt werden. War in der CMS_INPUT_FILE-Eingabekomponente zuvor `allowFolder="yes"` gesetzt, muss stattdessen das neu zu verwendende FS_REFERENCE-Formular folgendermaßen geändert werden:

```
<CMS_INPUT_FILE name="st_file" allowFolder="yes">  
  <LANGINFOS>  
    <LANGINFO lang="*" label="Datei-Auswahl" />  
  </LANGINFOS>  
</CMS_INPUT_FILE>
```



```
<FS_REFERENCE name="st_file">
  <FILTER>
    <ALLOW type="file"/>
    <HIDE type="picture"/>
    <ALLOW type="mediafolder"/>
  </FILTER>
  <LANGINFOS>
    <LANGINFO lang="*" label="Datei-Auswahl"/>
  </LANGINFOS>
  <PROJECTS>
    <LOCAL name=".">
      <SOURCES>
        <FOLDER name="root" store="mediastore"/>
      </SOURCES>
    </LOCAL>
  </PROJECTS>
</FS_REFERENCE>
```



FS_REFERENCE stellt einige Felder, die in den CMS_INPUT_-Eingabekomponenten enthalten waren, aufgrund seltener Verwendung nicht mehr zur Verfügung. Dies betrifft die Felder für die Erfassung eines Kommentars, von horizontalem und vertikalem Abstand, Höhe und Breite des Bildes sowie Breite eines Randes bei CMS_INPUT_PICTURE bzw. CMS_INPUT_FILE. Wurden diese Felder in Projekten doch verwendet und Daten eingepflegt, muss dies vor der Umstellung auf FS_REFERENCE entsprechend berücksichtigt werden.

- **CMS_INPUT_PAGEREF:** Der Parameter `mediaStore`, über den eine Auswahl von Medien aus der Medien-Verwaltung ermöglicht wurde, wird von FS_REFERENCE nicht unterstützt. Der Parameter muss für eine erfolgreiche Migration aus dem Formular entfernt werden. War in der CMS_INPUT_PAGEREF-Eingabekomponente `mediastore="yes"` gesetzt, muss das neu zu verwendende FS_REFERENCE-Formular folgendermaßen geändert werden:

```
<CMS_INPUT_PAGEREF name="st_pageref" hFill="yes" mediastore="yes">
  <LANGINFOS>
    <LANGINFO lang="*" label="Auswahl einer Seitenreferenz"/>
  </LANGINFOS>
</CMS_INPUT_PAGEREF>
```



```

<FS_REFERENCE name="st_file" hFill="yes">
  <FILTER>
    <ALLOW type="pageref"/>
    <ALLOW type="documentgroup"/>
    <ALLOW type="media"/>
  </FILTER>
  <LANGINFOS>
    <LANGINFO lang="*" label="Auswahl einer Seitenreferenz"/>
  </LANGINFOS>
  <PROJECTS>
    <LOCAL name=".">
      <SOURCES>
        <FOLDER name="root" store="sitestore"/>
        <FOLDER name="root" store="mediastore"/>
      </SOURCES>
    </LOCAL>
  </PROJECTS>
</FS_REFERENCE>

```

Der Parameter `sections`, über den in `CMS_INPUT_PAGEREF` vorgegeben werden konnte, ob in der Eingabekomponente nur Seitenreferenzen, nur Absätze oder Seitenreferenzen **und** Absätze ausgewählt werden können, ist auch für `FS_REFERENCE` verfügbar. Im Gegensatz zu `CMS_INPUT_PAGEREF` haben sich allerdings die Werte geändert:

	CMS_INPUT_PAGEREF	FS_REFERENCE
nur Seite	page (Standard)	no
Seite und Absätze	both	yes (Standard)
nur Absätze	section	only

War in der `CMS_INPUT_PAGEREF`-Eingabekomponente `sections="page"` oder kein Wert für `sections` gesetzt, muss im neu zu verwendenden `FS_REFERENCE`-Formular stattdessen `sections="no"` gesetzt werden, war `sections="section"` gesetzt, muss im neu zu verwendenden `FS_REFERENCE`-Formular stattdessen `sections="only"` gesetzt werden.

```

<CMS_INPUT_PAGEREF name="st_pageref" hFill="yes"
  sections="section">
  <LANGINFOS>
    <LANGINFO lang="*" label="Auswahl einer Seitenreferenz"/>
  </LANGINFOS>
</CMS_INPUT_PAGEREF>

```



```

<FS_REFERENCE name="st_file" hFill="yes" sections="only">
  <FILTER>
    <ALLOW type="pageref"/>
    <ALLOW type="documentgroup"/>
  </FILTER>
  <LANGINFOS>
    <LANGINFO lang="*" label="Auswahl einer Seitenreferenz"/>
  </LANGINFOS>
  <PROJECTS>
    <LOCAL name=".">
      <SOURCES>
        <FOLDER name="root" store="sitestore"/>
      </SOURCES>
    </LOCAL>
  </PROJECTS>
</FS_REFERENCE>

```

War `sections="both"` gesetzt, braucht im `FS_REFERENCE`-Formular kein Wert für `sections` gesetzt werden.

Der Parameter `sortOrder`, über den die Sortierreihenfolge der Absätze im Auswahldialog beeinflusst werden kann, wird von `FS_REFERENCE` nicht unterstützt. Der Parameter muss für eine erfolgreiche Migration aus dem Formular entfernt werden. In `FS_REFERENCE` werden auswählbare Absätze immer in der Reihenfolge angezeigt, in der sie in der Baumstruktur vorhanden sind.

5.1.4 Umstellung der Ausgabekanäle

Soll von den Eingabekomponenten `CMS_INPUT_PICTURE`, `CMS_INPUT_FILE` und `CMS_INPUT_PAGEREF` auf `FS_REFERENCE` umgestellt werden, muss neben der Formulardefinition ggf. auch der Inhalt der Ausgabekanal-Register angepasst werden (Anweisungen und Methoden). Denn die `CMS_INPUT_`-Eingabekomponenten liefern Objekte eines anderen Datentyps zurück als `FS_REFERENCE`:

CMS_INPUT_PICTURE: Datentyp <i>GraphicalMedium</i>	FS_REFERENCE: Datentyp <i>TargetReference</i>
CMS_INPUT_FILE: Datentyp <i>BinaryMedium</i>	
CMS_INPUT_PAGEREF: Datentyp <i>ElementReference</i>	



Daher ist zu prüfen, ob die Methoden, Funktionen und Anweisungen, die auf Objekten der Datentypen *GraphicalMedium*, *BinaryMedium* und *ElementReference* auch auf ein Objekt vom Datentyp *TargetReference* angewendet werden können. Der Bezeichner der Eingabekomponente, der in der Formulardefinition über den Parameter `name` angegeben ist, muss übernommen werden.

In folgenden Beispielen kann die Syntax des HTML-Kanals beibehalten werden:

- `getClass`:
 - Datentyp *GraphicalMedium*: `$CMS_VALUE(pt_picture.class)$`
 - Datentyp *BinaryMedium*: `$CMS_VALUE(pt_file.class)$`
 - Datentyp *ElementReference*: `$CMS_VALUE(pt_pageref.class)$`
 - Datentyp *TargetReference*: `$CMS_VALUE(pt_reference.class)$`
- `$CMS_REF(...)$`:
 - Datentyp *GraphicalMedium*: `$CMS_REF(pt_picture)$`
 - Datentyp *BinaryMedium*: `$CMS_REF(pt_file)$`
 - Datentyp *ElementReference*: `$CMS_REF(pt_pageref)$`
 - Datentyp *TargetReference*: `$CMS_REF(pt_reference)$`
- `ref(...)`-Funktion und `getUrl`:
 - Datentyp *GraphicalMedium*:
`$CMS_VALUE(ref(pt_picture).url)$`
 - Datentyp *BinaryMedium*: `$CMS_VALUE(ref(pt_file).url)$`
 - Datentyp *ElementReference*:
`$CMS_VALUE(ref(pt_pageref).url)$`
 - Datentyp *TargetReference*:
`$CMS_VALUE(ref(pt_reference).url)$`
- `getSectionName`:
 - Datentyp *ElementReference*:
`$CMS_VALUE(pt_pageref.sectionName)$`
 - Datentyp *TargetReference*:
`$CMS_VALUE(pt_reference.sectionName)$`



In folgenden Fällen muss beispielsweise die Ausgabe angepasst werden:

- `getUid`:
 - Datentyp *GraphicalMedium*: `$CMS_VALUE(pt_picture.medium.uid)$`
 - Datentyp *BinaryMedium*: `$CMS_VALUE(pt_file.medium.uid)$`
 - Datentyp *ElementReference*: `$CMS_VALUE(pt_pageref.pageRef.uid)$`
 - Datentyp *TargetReference*: `$CMS_VALUE(pt_reference.uid)$`
- `getRemoteName` / `getRemote`:
 - Datentyp *GraphicalMedium*:
`$CMS_VALUE(pt_picture.remoteName)$`
 - Datentyp *BinaryMedium*: `$CMS_VALUE(pt_file.remoteName)$`
 - Datentyp *ElementReference*: `$CMS_VALUE(pt_pageref.remote)$`
 - Datentyp *TargetReference*:
`$CMS_VALUE(pt_reference.remoteSymName)$`
- `ref(...)`-Funktion und `getRemoteName` / `getRemote`:
 - Datentyp *GraphicalMedium*:
`$CMS_REF(pt_file, remote:pt_file.remoteName)$`
 - Datentyp *BinaryMedium*:
`$CMS_REF(pt_picture, remote:pt_picture.remoteName)$`
 - Datentyp *ElementReference*:
`$CMS_REF(pt_pageref, remote:pt_pageref.remote)$`
 - Datentyp *TargetReference*:
`$CMS_REF(pt_reference, remote:pt_reference.remoteSymName)$`



Für alle anderen Verwendungen muss geprüft werden, ob Methoden, Anweisungen und Funktionen angepasst werden müssen. Siehe dazu auch FirstSpirit Online Dokumentation.



5.2 Eingabekomponenten: Migration zu FS_DATASET

Für FirstSpirit-Version 5.0 ist geplant, dass, wie bereits in Version 4.2 angekündigt, die Eingabekomponente CMS_INPUT_OBJECTCHOOSER

The screenshot shows a web form titled "OBJECTCHOOSER". At the top, there are several icons for actions like refresh, close, and list. Below the icons, a status message reads "Der Datensatz wird an 3 weiteren Stellen verwe..." followed by a button labeled "Verwendungen anzeigen". The form contains three input fields: "Lfd. Nummer" (empty), "Vorname" (containing "Charly"), and "Nachname" (containing "Chef").

Abbildung 5-3: CMS_INPUT_OBJECTCHOOSER – Standardkonfiguration

entfallen und durch FS_DATASET ersetzt wird:

The screenshot shows a web form titled "FS_DATASET". At the top, there is a search input field containing "Kontakte" and several action icons. Below this, the form contains three input fields: "Lfd. Nummer" (empty), "Vorname" (containing "Charly"), and "Nachname" (containing "Chef"). At the bottom, a status message reads "Der Datensatz wird an 3 weiteren Stellen verwendet." followed by a button labeled "Verwendungen anzeigen".

Abbildung 5-4: FS_DATASET – Standardkonfiguration



5.2.1 Funktionsunterschiede FS_DATASET und CMS_INPUT_OBJECTCHOOSER

Im Gegensatz zur Eingabekomponente CMS_INPUT_OBJECTCHOOSER wird in FS_DATASET die Auswahl standardmäßig nicht auf Datensätze einer definierten Zieltabelle eingeschränkt. Stattdessen kann die gewünschte Tabelle über einen Auswahldialog gewählt werden. Dabei werden gleichzeitig die Zieltabelle und der gewünschte Datensatz ausgewählt.

5.2.2 Umstellung von bestehenden Formularen

Ab Version 4.2R4 werden Werte, die bereits mit CMS_INPUT_OBJECTCHOOSER von Redakteuren eingepflegt wurden, von FS_DATASET übernommen und verarbeitet. Die Umstellung des Formulars von CMS_INPUT_OBJECTCHOOSER auf FS_DATASET ist mithilfe weniger manueller Änderungen möglich.

Für die Migration muss dazu in der Syntax das CMS_INPUT_OBJECTCHOOSER-Tag durch das FS_DATASET-Tag ersetzt werden. Der Bezeichner der Eingabekomponente, der über den Parameter `name` angegeben ist, muss hingegen bestehen bleiben. Nur so kann gewährleistet werden, dass bereits durch den Redakteur eingepflegte Daten weiterhin gespeichert werden und bearbeitbar sind.

Beispiel:

```
<CMS_INPUT_OBJECTCHOOSER name="st_dataset"...>
```

muss durch

```
<FS_DATASET name="st_dataset"...>
```

ersetzt werden.

Soll die Auswahl nur aus der Tabelle möglich sein, die in CMS_INPUT_OBJECTCHOOSER vorgegeben war, muss die Referenz zur Tabellenvorlage, die in CMS_INPUT_OBJECTCHOOSER über den Pflichtparameter `table` angegeben wurde, in FS_DATASET mithilfe des Parameters `name` innerhalb der Tags `SOURCES` und `CONTENT` angegeben werden. Anders als für CMS_INPUT_OBJECTCHOOSER muss in FS_DATASET der Referenzname der gewünschten Datenquelle verwendet werden, z. B.

```
<SOURCES>  
  <CONTENT name="produkte"/>  
</SOURCES>
```





Damit die Daten, die durch den Redakteur bereits über `CMS_INPUT_OBJECTCHOOSER` eingegeben wurden, weiterhin gespeichert und bearbeitbar bleiben, muss zusätzlich der eindeutige Bezeichner der Tabellenvorlage aus dem zu migrierenden `CMS_INPUT_OBJECTCHOOSER`-Formular über das Tag `LEGACYTEMPLATE` angegeben werden, z. B.

```
<LEGACYTEMPLATE>Products.products</LEGACYTEMPLATE>
```

5.2.3 Umstellung der Ausgabekanäle

Soll von der Eingabekomponente `CMS_INPUT_OBJECTCHOOSER` auf `FS_DATASET` umgestellt werden, muss neben der Formulardefinition ggf. auch der Inhalt der Ausgabekanal-Register angepasst werden. Denn `CMS_INPUT_OBJECTCHOOSER` liefert Objekte eines anderen Datentyps zurück als `FS_DATASET`:

<code>CMS_INPUT_OBJECTCHOOSER</code> : Datentyp <i>Entity</i>	<code>FS_DATASET</code> : Datentyp <i>DatasetContainer</i>
--	---

Daher ist zu prüfen, ob die Methoden und Anweisungen, die auf Objekten des Datentyps *Entity* auch auf ein Objekt vom Datentyp *DatasetContainer* angewendet werden können. Der Bezeichner der Eingabekomponente, der in der Formulardefinition über den Parameter `name` angegeben ist, muss übernommen werden.

Während `get`-Methoden, die auf dem Datentyp-Objekt *Entity* aufgerufen werden, direkt auf einem Datensatz ausgeführt werden, umfasst das Datentyp-Objekt *DatasetContainer* neben dem Datensatz zusätzlich die zugehörige Tabellenvorlage. Daher muss bei der Umstellung von `CMS_INPUT_OBJECTCHOOSER` auf `FS_DATASET` die Methode `.getDataset().getEntity()` (in Bean-Syntax: `.dataset.entity`) ergänzt werden.

Beispiel:

So würde z. B. die Ausgabe des Wertes der "spalte1" eines über `CMS_INPUT_OBJECTCHOOSER` ausgewählten Datensatzes über

```
$CMS_VALUE(st_dataset.spalte1)$
```

bei der Umstellung auf `FS_DATASET` folgendermaßen lauten:



```
$CMS_VALUE(st_dataset.dataset.entity.spalte1)$
```

Für alle anderen Verwendungen muss geprüft werden, ob Methoden, Anweisungen und Funktionen angepasst werden müssen. Siehe dazu auch FirstSpirit Online Dokumentation.

5.3 Eingabekomponenten: Weiterentwicklung FS_LIST

Die Eingabekomponente FS_LIST bildet seit ihrer Einführung in der initialen Release-Version zu 4.2 die Funktionen der Eingabekomponente CMS_INPUT_CONTENTAREALIST ab. Ziel der in 4.2 begonnenen Konsolidierung der FirstSpirit-Eingabekomponenten soll sein, alle Eingabekomponenten, die Listencharakter haben, in FS_LIST zusammenzufassen. Daher wurden in Version 4.2R4 die Funktionen der folgenden Eingabekomponenten hinzugefügt:

- CMS_INPUT_CONTENTLIST
- CMS_INPUT_LINKLIST
- CMS_INPUT_SECTIONLIST
- CMS_INPUT_TABLIST

Dazu kann FS_LIST auf Daten verschiedener Herkunft bzw. verschiedener Typen zugreifen. Der Datentyp wird über den Parameter `type` im Tag `DATASOURCE` festgelegt. Je nach Datentyp können weitere Tags zur Konfiguration der Eingabekomponente verwendet werden. Beispielsweise kann der Vorlagen-Entwickler auch vorgeben, welche Icons dem Redakteur zur Aufnahme und Bearbeitung von Listen-Inhalten zur Verfügung stehen sollen (z. B. "Hinzufügen", "Löschen", "Bearbeiten").



In der FirstSpirit Online Dokumentation wird ein Beispielprojekt bereitgestellt, das u.a. beispielhafte Konfigurationen von FS_LIST und die Umstellung der CMS_INPUT-Eingabekomponenten zu FS_LIST (Formular und Ausgabe) illustriert. Siehe dazu "Vorlagenentwicklung" / "Formulare" / "Eingabekomponenten (neu)" / "FS_LIST" / "Beispielprojekt".





In **WebEdit** wird `FS_LIST` in 4.2R4 nur rudimentär unterstützt: aktuell stehen nur die Funktionen der Eingabekomponente `CMS_INPUT_CONTENTAREALIST` (Liste zur Verwaltung von Absatzinhalten) zur Verfügung. Um mehr Funktionen bereitzustellen, kann der sogenannte (nicht freigegebene!) Applet-Modus für `FS_LIST` aktiviert werden (Projekteinstellungen, "WebEdit-Einstellungen", Option "Editor-Applet benutzen" aktivieren, in Feld "Applet-Editoren" "`FS_LIST`" eintragen). Dies ist aber nur eine Übergangslösung, eine native `FS_LIST`-Unterstützung wird es ab FirstSpirit Version 5.0 geben.

5.3.1 Konfiguration

Welche Tags und Parameter zur Konfiguration von `FS_LIST` innerhalb des Formularbereichs zur Verfügung stehen, wird ausführlich in der *FirstSpirit Online Dokumentation* vorgestellt, siehe "Vorlagenentwicklung" / "Formulare" / "Eingabekomponenten (neu)" / "`FS_LIST`".



`FS_LIST` unterstützt für den `DATASOURCE`-Typ `INLINE` mit `source="LINKTEMPLATES"` nur die Auswahl von Verweisen des Typs "genericLink"! Daher wird empfohlen, **vor** der Verwendung von `FS_LIST` oder der Migration zu `FS_LIST` Verweise vom Typ "internalLink", "externalLink" und "contentLink" zu konvertieren. Dazu kann die Konvertierungsfunktion verwendet werden (Kontextmenü von Verweissvorlagen, "Extras" / "Verweissvorlage konvertieren").



In **WebEdit** wird `FS_LIST` nur für den `DATASOURCE`-Typ `INLINE` mit `source="SECTIONTEMPLATES"` unterstützt (entspricht `CMS_INPUT_CONTENTAREALIST`).

5.3.2 Umstellung von bestehenden Formularen

Die Migration der Eingabekomponenten, die von `FS_LIST` ersetzt werden sollen (`CMS_INPUT_CONTENTAREALIST`, `CMS_INPUT_CONTENTLIST`, `CMS_INPUT_LINKLIST`, `CMS_INPUT_SECTIONLIST` und `CMS_INPUT_TABLIST`), nach `FS_LIST` ist möglich, indem die Formulare angepasst werden.





Die Syntax von `FS_LIST` ist mit Version 4.2R4 überarbeitet worden, daher müssen auch Verwendungen von `FS_LIST`, die mit einer FirstSpirit-Version unterhalb von 4.2R4 erstellt worden sind, angepasst werden. Hier gelten die gleichen Hinweise hinsichtlich der Umstellung wie für `CMS_INPUT_CONTENTAREALIST`.

Für die Migration muss in der Syntax das `CMS_INPUT_-`Tag durch das `FS_LIST-`Tag ersetzt werden, z. B. muss im Formular `<CMS_INPUT_LINKLIST ...>` durch `<FS_LIST ...>` ersetzt werden. Der Bezeichner der Eingabekomponente, der über den Parameter `name` angegeben ist, muss hingegen bestehen bleiben. Viele der in den `CMS_INPUT-`Eingabekomponenten verwendeten Parameter können für `FS_LIST` übernommen werden, folgende Tags und Parameter müssen hingegen ersetzt oder hinzugefügt werden:

Für alle Eingabekomponenten:

- Die Parameter `maxEntries`, über den die Anzahl der auswählbaren bzw. hinzufügbaren Einträge definiert wird, und `useLanguages`, über den eine mehrsprachige Pflege von Daten ermöglicht bzw. unterbunden wird, wurden in den `CMS_INPUT_-`Eingabekomponenten innerhalb des `CMS_INPUT_-`Tags angegeben. In `FS_LIST` können diese Parameter übernommen werden, allerdings müssen sie dort im `DATASOURCE-`Tag angegeben werden, z. B. muss statt

```
<CMS_INPUT_CONTENTLIST name="st_datasets"
  table="Products.contacts" maxEntries="5"
  useLanguages="no" ...>
```

```
<FS_LIST name="st_sections" ...>
  <DATASOURCE type="inline" maxEntries="5"
    useLanguages="no">
  <TABLE>Products.contacts</TABLE>
  ...
```

definiert werden.

- `FS_LIST`-Eingabekomponenten sind standardmäßig 500 Pixel hoch und damit wahrscheinlich höher als zuvor verwendete `CMS_INPUT-`Eingabekomponenten. Soll die Höhe reduziert werden, muss dafür der Parameter `height` mit der gewünschten Pixelzahl gesetzt werden, z. B.

```
<FS_LIST name="st_sections" height="300" .../>
```



5.3.2.1 Anpassungen bei CMS_INPUT_CONTENTAREALIST bzw. FS_LIST

- Es muss `<DATASOURCE type="INLINE">` ergänzt werden.
- Das Tag `SOURCES` muss durch das Tag `TEMPLATES` ersetzt werden und zusätzlich `source="sectiontemplates"` ergänzt werden.
- Waren über `SOURCES / TEMPLATE` in der Eingabekomponente auswählbare Absatzvorlagen definiert, muss das Tag `TEMPLATE` erhalten bleiben, der Parameter `name` muss durch den Parameter `uid` ersetzt werden, der Wert bleibt allerdings bestehen.
- Um eine Bearbeitungsmöglichkeit der Listeneinträge wie in `CMS_INPUT_CONTENTAREALIST` zur ermöglichen, muss folgende Syntax ergänzt werden:

```
<ACTIONS>
    <ACTION name="ADD" />
    <ACTION name="REMOVE" />
    <ACTION name="UP" />
    <ACTION name="DOWN" />
    <ACTION name="EDIT" />
</ACTIONS>
```

- War über die Tags `VARIABLES / VARIABLE` eine Beschriftung der Absätze in der Liste definiert, muss diese in `FS_LIST` über die Tags `LABELS / LABEL` nachgestellt werden. Hier kann mithilfe von `#item` auf den in einer Eingabekomponente des Absatzes gespeicherten Wert zugegriffen werden. Dazu wird die Variable (Bezeichner) der betreffenden Eingabekomponente durch einen Punkt getrennt angehängt, z. B.

```
<LABELS>
    <LABEL lang="*">
        #item.VARIABLE
    </LABEL>
</LABELS>
```

- War in `CMS_INPUT_CONTENTAREALIST` innerhalb der `VARIABLES / VARIABLE` Tags die Variable `template` definiert, mit der der Referenzname der verwendeten Vorlage für einen Eintrag ausgegeben wurde, wird in `FS_LIST` dafür nun das Systemobjekt `#template` innerhalb der Tags `LABELS / LABEL` eingesetzt. Aus der Definition

```
<VARIABLES>
    <VARIABLE name="template" />
</VARIABLES>
```

würde dann in `FS_LIST`

```
<LABELS>
    <LABEL lang="*">#template.uid</LABEL>
```




```
</LABELS>
```

- Um das Layout von CMS_INPUT_CONTENTAREALIST zu erhalten, kann folgende Syntax ergänzt werden:

```
<COLUMNS>
    <COLUMN show="no">#identifizier</COLUMN>
</COLUMNS>
<LAYOUT>
    <ADD component="toolbar" constraint="top"/>
    <ADD component="overview" constraint="center"/>
    <ADD component="stackedview" constraint="hide"/>
</LAYOUT>
```



Für die Funktionalität in CMS_INPUT_CONTENTAREALIST, Absätzen manuell einen Namen zuzuweisen bzw. umzubenennen (z. B. Icon ) , der dann in der Übersicht angezeigt wird, gibt es in FS_LIST keine Entsprechung. Für die Listenübersicht und auf den Balken wird, wenn der gewählte Absatz ausschließlich "leere" Eingabekomponenten enthält (z. B. bei Neuanlage, wenn noch keine Werte eingepflegt sind), standardmäßig der Anzeigename der Absatzvorlage angezeigt, bei "gefüllten" Eingabekomponenten die Beschriftung und der Inhalt der Eingabekomponenten.



Damit bereits manuell zugewiesene Absatznamen auch in FS_LIST erhalten und gespeichert bleiben, muss in der oder den betreffenden Absatzvorlage/n (die über die CMS_INPUT_CONTENTAREALIST ausgewählt wurden) zusätzlich eine Eingabekomponente mit der Variablen (Bezeichner) `displayName` (Groß- und Kleinschreibung beachten!) definiert werden, in der dieser Absatzname gespeichert werden soll. Die Eingabekomponente muss vom Typ CMS_INPUT_TEXT oder CMS_INPUT_TEXTAREA sein.

Ein entsprechendes FS_LIST-Formular würde in diesem Fall folgendermaßen aussehen:

```
<FS_LIST name="st_cal">
    <DATASOURCE type="inline" useLanguages="no">
        <LABELS>
            <LABEL lang="*">#item.displayName</LABEL>
        </LABELS>
        <ACTIONS>
            <ACTION name="ADD">
                <PARAM name="select-ui">popup</PARAM>
            </ACTION>
            <ACTION name="REMOVE"/>
        </ACTIONS>
    </DATASOURCE>
</FS_LIST>
```



```

        <ACTION name="UP" />
        <ACTION name="DOWN" />
        <ACTION name="EDIT" />
    </ACTIONS>
    <COLUMNS>
        <COLUMN show="no">#identifier</COLUMN>
    </COLUMNS>
    <LAYOUT>
        <ADD component="toolbar" constraint="top" />
        <ADD component="overview" constraint="center" />
        <ADD component="stackedview" constraint="hide">
        <PARAM name="show-language-tabs">yes</PARAM>
        </ADD>
    </LAYOUT>
    <TEMPLATES source="sectiontemplates" />
</DATASOURCE>
<LANGINFOS>
    <LANGINFO lang="*" label="Absätze" />
</LANGINFOS>
</FS_LIST>

```

Bei Verwendung von CMS_INPUT_CONTENTAREALIST konnte darüber hinaus z. B. für eine Art Inhaltsverzeichnis am Anfang einer Seite über folgende Syntax eine Liste der Absätze erstellt werden:

```

$CMS_FOR(entry, st_cal)$
    <a href="#"$CMS_VALUE(entry.name)$">Section
$CMS_VALUE(#for.index + 1)$</a>
$CMS_END_FOR$

```

Zur Verwendung mit FS_LIST sollte diese folgendermaßen angepasst werden:

```

$CMS_FOR(entry, st_cal)$
    <a href="#"$CMS_VALUE(if(entry.displayName.isEmpty, "st_cal_" +
#for.index, entry.displayName))$">Section $CMS_VALUE(#for.index +
1)$</a>
$CMS_END_FOR$

```

In der Absatzvorlage, die vom Redakteur ausgewählt wird, sollte folgende Syntax als Verweisziel (Anker) ergänzt werden:

```

$CMS_SET(anchorName, if(displayName.isEmpty, "st_cal_" + #index,
displayName))$
    <a id="$CMS_VALUE(anchorName)$"
name="$CMS_VALUE(anchorName)$"></a>

```



5.3.2.2 Anpassungen bei CMS_INPUT_LINKLIST

- Es muss `<DATASOURCE type="INLINE">` ergänzt werden.
- Das Tag `LINKEDITORS` muss durch das Tag `TEMPLATES` ersetzt werden, zusätzlich muss `source="linktemplates"` ergänzt werden.
- Waren über `LINKEDITORS` / `LINKEDITOR` in der Eingabekomponente auswählbare Verweissvorlagen definiert, muss das Tag `LINKEDITOR` durch das Tag `TEMPLATE` ersetzt werden, der Parameter `name` muss durch den Parameter `uid` ersetzt werden, der Wert bleibt allerdings bestehen
- Um eine Bearbeitungsmöglichkeit der Listeneinträge wie in `CMS_INPUT_LINKLIST` zur ermöglichen, muss folgende Syntax ergänzt werden:

```
<ACTIONS>
  <ACTION name="ADD" />
  <ACTION name="REMOVE" />
  <ACTION name="UP" />
  <ACTION name="DOWN" />
  <ACTION name="EDIT" />
</ACTIONS>
```

- Damit für die Beschriftung in der Listenübersicht und der Balken wie bei `CMS_INPUT_LINKLIST` der Wert verwendet wird, der als Verweis-Text eingegeben wird, müssen in `FS_LIST` die Tags `LABELS` / `LABEL` verwendet werden. Hier kann mithilfe von `#item` auf den in einer Eingabekomponente der Verweissvorlage gespeicherten Wert zugegriffen werden. Dazu wird die Variable (Bezeichner) der betreffenden Eingabekomponente durch einen Punkt getrennt angehängt, z. B.

```
<LABELS>
  <LABEL lang="*">
    #item.VARIABLE
  </LABEL>
</LABELS>
```

- Um das Layout von `CMS_INPUT_LINKLIST` zu erhalten, kann folgende Syntax ergänzt werden:

```
<COLUMNS>
  <COLUMN show="no">#identifier</COLUMN>
</COLUMNS>
<LAYOUT>
  <ADD component="toolbar" constraint="top"/>
  <ADD component="overview" constraint="center"/>
  <ADD component="stackedview" constraint="hide"/>
</LAYOUT>
```





FS_LIST unterstützt für den DATASOURCE-Typ INLINE mit `source="LINKTEMPLATES"` nur die Auswahl von Verweisen des Typs "genericLink"! Daher wird empfohlen, vor der Verwendung von FS_LIST oder der Migration zu FS_LIST Verweise vom Typ "internalLink", "externalLink" und "contentLink" zu konvertieren. Dazu kann die Konvertierungsfunktion verwendet werden (Kontextmenü von Verweissvorlagen, "Extras" / "Verweissvorlage konvertieren").

5.3.2.3 Anpassungen bei CMS_INPUT_CONTENTLIST

- Es muss `<DATASOURCE type="DATABASE">` ergänzt werden.
- Die über den Parameter `table` definierte Tabellenvorlage muss innerhalb eines öffnenden und schließenden `TABLE`-Tags angegeben werden:

```
<TABLE>Products.contacts</TABLE>
```

- Um eine Bearbeitungsmöglichkeit der Listeneinträge wie in `CMS_INPUT_CONTENTLIST` zur ermöglichen, muss folgende Syntax ergänzt werden:

```
<ACTIONS>
  <ACTION name="ADD" />
  <ACTION name="REMOVE" />
  <ACTION name="UP" />
  <ACTION name="DOWN" />
  <ACTION name="GOTO" />
  <ACTION name="VIEW" />
</ACTIONS>
```

- War in `CMS_INPUT_CONTENTLIST` `allowEdit="yes"` gesetzt, muss innerhalb der `ACTIONS`-Tags zusätzlich `<ACTION name="EDIT" />` ergänzt werden, war `allowNew="yes"` gesetzt, muss `<ACTION name="NEW" />` ergänzt werden.
- Wurde für `CMS_INPUT_CONTENTLIST` der Parameter `indexAttribute` angegeben (z. B. bei der Verwendung als Medien-Galerie, Tag `MEDIAMODE`), muss der Wert, der für den Parameter `indexAttribute` angegeben war, innerhalb eines öffnenden und eines schließenden `PERSISTENCEINDEX`-Tags angegeben werden, z. B. wird dann aus
`CMS_INPUT_CONTENTLIST: indexAttribute="Order_Index"`
`FS_LIST: <PERSISTENCEINDEX>Order_Index</PERSISTENCEINDEX>`
- Während die Definition einer Abfrage, mithilfe derer die Menge der auswählbaren Datensätze eingeschränkt werden kann, in `CMS_INPUT_CONTENTLIST` über das Tag `QUERY` innerhalb des Formulars erfolgte, erfolgt sie bei `FS_LIST` in



einer Abfrage in der Vorlagen-Verwaltung. Im FS_LIST-Formular muss auf den Referenznamen dieser Abfrage folgendermaßen zugegriffen werden:

```
<QUERY name="query_products" />
```

- Um das Layout von CMS_INPUT_CONTENTLIST zu erhalten, kann folgende Syntax ergänzt werden:

```
<COLUMNS>
    <COLUMN show="no">#identifizier</COLUMN>
</COLUMNS>
<LAYOUT>
    <ADD component="toolbar" constraint="top" />
    <ADD component="overview" constraint="center" />
    <ADD component="stackedview" constraint="hide" />
</LAYOUT>
```

5.3.2.4 Anpassungen bei CMS_INPUT_TABLIST

- Es muss `<DATASOURCE type="DATABASE">` ergänzt werden.
- Die über den Parameter `table` definierte Tabellenvorlage muss innerhalb eines öffnenden und schließenden `TABLE`-Tag angegeben werden:

```
<TABLE>Products.contacts</TABLE>
```

- War in CMS_INPUT_TABLIST über `tabAttribute` oder `expression` eine flexible Beschriftung der Register definiert, müssen in FS_LIST die Tags `LABELS` / `LABEL` verwendet werden. Hier kann mithilfe von `#item` auf den in einer Spalte gespeicherten Wert zugegriffen werden. Dazu wird die Variable der Spalte (Bezeichner der Eingabekomponente) durch einen Punkt getrennt angehängt, z. B.

```
<LABELS>
    <LABEL lang="*">
        #item.VARIABLE
    </LABEL>
</LABELS>
```

- Um eine Bearbeitungsmöglichkeit der Listeneinträge wie in CMS_INPUT_TABLIST zur ermöglichen, muss folgende Syntax ergänzt werden:

```
<ACTIONS>
    <ACTION name="ADD" />
    <ACTION name="REMOVE" />
    <ACTION name="EDIT" />
</ACTIONS>
```

- War in CMS_INPUT_TABLIST `allowEdit="yes"` gesetzt, muss innerhalb der `ACTIONS`-Tags zusätzlich `<ACTION name="EDIT" />` ergänzt werden, war



allowNew="yes" gesetzt, muss `<ACTION name="NEW" />` ergänzt werden. War allowChoose="no" gesetzt, muss `<ACTION name="ADD" />` entfernt werden. War allowEdit="no" gesetzt, muss `<ACTION name="EDIT" />` entfernt werden. War delete="yes" gesetzt, muss `<ACTION name="DELETE" />` ergänzt werden.

- Um die Register-Ansicht von CMS_INPUT_TABLIST zu erhalten, kann folgende Syntax ergänzt werden:

```
<COLUMNS>
    <COLUMN show="no">#identifizier</COLUMN>
</COLUMNS>
<LAYOUT>
    <ADD component="toolbar" constraint="top" />
    <ADD component="overview" constraint="hide" />
    <ADD component="tabbedview" constraint="bottom" />
</LAYOUT>
```

5.3.2.5 Anpassungen bei CMS_INPUT_SECTIONLIST

- Es muss `<DATASOURCE type="PAGE">` ergänzt werden.

Grundsätzlich müssen alle geöffneten Tags auch wieder entsprechend geschlossen werden.

5.3.3 Umstellung der Ausgabekanäle

Soll von den Eingabekomponenten CMS_INPUT_LINKLIST, CMS_INPUT_CONTENTAREALIST, CMS_INPUT_CONTENTLIST, CMS_INPUT_TABLIST, CMS_INPUT_SECTIONLIST und FS_LIST (erstellt vor FirstSpirit Version 4.2R4) auf FS_LIST umgestellt werden, muss neben der Formulardefinition ggf. auch der Inhalt der Ausgabekanal-Register angepasst werden (Anweisungen und Methoden). Denn die CMS_INPUT_-Eingabekomponenten liefern Objekte eines anderen Datentyps zurück als FS_LIST:

CMS_INPUT_CONTENTAREALIST: Datentyp <i>SectionList</i> <code><Section ></code>	FS_LIST (ab 4.2R4): Datentyp <i>FormDataList</i> <code><IdProvidingFormData></code>
CMS_INPUT_CONTENTLIST: Datentyp <i>List</i> <code><Entity ></code>	
CMS_INPUT_LINKLIST: Datentyp <i>List</i> <code><Link ></code>	



CMS_INPUT_SECTIONLIST: Datentyp <i>List</i> < <i>SectionListEntry</i> >	
CMS_INPUT_TABLIST: Datentyp <i>List</i> < <i>Entity</i> >	
FS_LIST (bis 4.2R2): Datentyp <i>SectionList</i> < <i>Section</i> >	



FS_LIST befindet sich in FirstSpirit Version 4.2 im Status "in Entwicklung". Syntax und API zu FS_LIST wurden gegenüber 4.2R2 leicht geändert, um eine konsistentere Syntax zu schaffen. Es ist nicht ausgeschlossen, dass sich die Syntax zu FS_LIST erneut ändern kann.

5.3.3.1 Beispiel CMS_INPUT_CONTENTAREALIST (FS_LIST, Datasource-Typ **INLINE**)

CMS_INPUT_CONTENTAREALIST liefert eine Liste von *Section*-Objekten zurück. Ein *Section*-Objekt repräsentiert dabei den Inhalt eines über CMS_INPUT_CONTENTAREALIST gepflegten Absatzes. Mithilfe einer `§CMS_FOR(...)`-Anweisung kann auf die einzelnen *Section*-Objekte zugegriffen werden:

```
§CMS_FOR(_obj, pt_cal)§
  §CMS_VALUE(_obj)§
§CMS_END_FOR§
```

Dabei ist `pt_cal` der Bezeichner einer CMS_INPUT_CONTENTAREALIST-Eingabekomponente, `_obj` einer der in der CMS_INPUT_CONTENTAREALIST angelegten Absätze. Die Daten der einzelnen Absätze werden dann mithilfe der jeweiligen Absatzvorlage ausgegeben.

Bei Verwendung der neuen FS_LIST-Komponente kann diese Ausgabe-Syntax übernommen werden. Darüber hinaus kann auf die in FS_LIST gespeicherten Daten einer einzelnen Eingabekomponente einfach zugegriffen werden, indem der Bezeichner der Eingabekomponente angegeben wird:

```
§CMS_FOR(_obj, pt_cal)§
  §CMS_VALUE(_obj.st_text)§
§CMS_END_FOR§
```

Dabei ist `st_text` der Bezeichner der Eingabekomponente in der Absatzvorlage,



die mithilfe von FS_LIST ausgewählt wird.



Hinweise zur Verwendung von CMS_INPUT_CONTENTAREALIST in Verbindung mit WebEdit "Easy-Edit" siehe "FirstSpirit Online Dokumentation", "Vorlagenentwicklung" / "WebEdit" / "Easy-Edit", Absatz "Sonderfall CMS_INPUT_CONTENTAREALIST und FS_LIST".

Die Ausgabe der FS_LIST-Eingabekomponente als Ersatz für CMS_INPUT_LINKLIST erfolgt analog, wobei hier der Bezeichner der Eingabekomponente der ausgewählten Verweissvorlage verwendet wird.

5.3.3.2 Beispiel CMS_INPUT_TABLIST (FS_LIST, Datasource-Typ **DATABASE**)

Die Eingabekomponenten CMS_INPUT_TABLIST und CMS_INPUT_CONTENTLIST liefern eine Liste von Objekten vom Datentyp *Entity* zurück (vgl. dazu auch Kapitel 5.2.3 Seite 92). Mithilfe von `$CMS_FOR(...)` können die Daten der in diesen Eingabekomponenten ausgewählten Datensätze ausgegeben werden, z. B. die ID über die Methode `.getId()`:

```
$CMS_FOR(_obj, pt_tablist)$  
    $CMS_VALUE(_obj.getId())$  
$CMS_END_FOR$
```

Daten einer Spalte eines ausgewählten Datensatzes können mithilfe der Angabe des Spaltennamens ausgegeben werden:

```
$CMS_FOR(_obj, pt_tablist)$  
    $CMS_VALUE(_obj.headline)$  
$CMS_END_FOR$
```

Dabei ist `pt_tablist` der Bezeichner der CMS_INPUT_TABLIST-Eingabekomponente, `_obj` einer der in der CMS_INPUT_TABLIST ausgewählten Datensätze und `headline` der Spaltenname (dieser kann je nach Konfiguration auch für verschiedene Sprachen definiert sein, z. B. `headline_DE` und `headline_EN`).

Bei Verwendung der neuen FS_LIST-Komponente wird statt der Variablen der Tabellenspalte, deren Daten ausgegeben werden sollen, die Variable der Eingabekomponente angegeben, die dieser Tabellenspalte zugeordnet ist (Register "Mapping" der Tabellenvorlage). Die ID kann über den Schlüsselbegriff `fs_id` ermittelt werden.



Ausgabe der ID der ausgewählten Datensätze:

```
$CMS_FOR(_obj, pt_tablist)$
  Datensatz-ID: $CMS_VALUE(_obj.fs_id)$
$CMS_END_FOR$
```

Ausgabe der Daten einer Eingabekomponente:

```
$CMS_FOR(_obj, pt_tablist)$
  Text: $CMS_VALUE(_obj.cs_headline)$
$CMS_END_FOR$
```

Dabei ist `cs_headline` der Bezeichner der Eingabekomponente in der Tabellenvorlage des Datensatzes, der mithilfe von `FS_LIST` ausgewählt wird. Hier ist daher keine sprachabhängige Unterscheidung notwendig.

Bei der Verwendung der `FS_LIST`-Eingabekomponente als Ersatz für `CMS_INPUT_CONTENTLIST` erfolgt die Umstellung analog.

5.3.3.3 Beispiel CMS_INPUT_SECTIONLIST (FS_LIST, Datasource-Typ **PAGE**)

Die Eingabekomponente `CMS_INPUT_SECTIONLIST` liefert eine Liste von `SectionListEntry`-Objekten zurück. Die Ausgabe der einzelnen Absatzeinträge erfolgt mit einer `$CMS_FOR(...)$`-Anweisung. Auf einem Objekt können dabei z. B. folgende Methoden angewendet werden:

- `getSectionName` (in Bean-Syntax: `.sectionName`)
Ausgabe des Namens des Absatzes
- `getAnchorName` (in Bean-Syntax: `.anchorName`)
Ausgabe des um Sonderzeichen bereinigten Namens des Absatzes (Sonderzeichen werden durch Unterstriche ersetzt)
Hinweis: Diese Methode ist potenziell nur für Projekte relevant, die mit einer FirstSpirit-Version unterhalb von 4.x erstellt worden sind. Ab FirstSpirit-Version 4.x sind keine Sonderzeichen in Referenznamen mehr möglich.
- `getText` (in Bean-Syntax: `.text`)
Ausgabe der (manuell vergebenen) Beschriftung des Absatzes

Beispiel:

```
<ul>
  $CMS_FOR(_obj, pt_sectionlist)$
    <li>
      <a href="#"#$CMS_VALUE(_obj.anchorName)$">
        Label: $CMS_VALUE(_obj.text)$ <br>
        UID: $CMS_VALUE(_obj.sectionName)$</a>
    </li>
```



```
$CMS_END_FOR$  
</ul>
```

Dabei ist `pt_sectionlist` der Bezeichner einer `CMS_INPUT_SECTIONLIST`-Eingabekomponente

Diese Methoden können bei einer Migration zu `FS_LIST` weiterhin verwendet werden. Wird `FS_LIST`, `DATASOURCE`-type `PAGE`, neu eingesetzt, sollten stattdessen die Schlüsselbegriffe `uid` und `label` verwendet werden: Mit `uid` wird der Name eines Absatzes ausgegeben (zuvor: `getSectionName`), mithilfe von `label` die Beschriftung (zuvor: `getText`). Da ab FirstSpirit-Version 4.x keine Sonderzeichen in Absatznamen mehr erlaubt sind, kann für Projekte, die mit dieser Version angelegt wurden, statt `getAnchorName` `getUid` verwendet werden, z. B.:

```
<ul>  
  $CMS_FOR(_obj, pt_sectionlist)$  
    <li>  
      <a href="#"$CMS_VALUE(_obj.uid)$">  
        Label: $CMS_VALUE(_obj.label)$      <br>  
        UID:   $CMS_VALUE(_obj.uid)$</a>  
    </li>  
  $CMS_END_FOR$  
</ul>
```

Für alle anderen Verwendungen muss geprüft werden, ob Methoden, Anweisungen und Funktionen angepasst werden müssen. Siehe dazu auch FirstSpirit Online Dokumentation.

5.4 Neue Eingabekomponente `FS_BUTTON`

Die Eingabekomponente `FS_BUTTON` stellt dem Redakteur ein Icon, eine Schaltfläche oder einen Verweis zur Verfügung, der an ein FirstSpirit-Skript oder an eine Klasse gebunden ist, die z. B. von einem FirstSpirit-Modul auf dem Server zur Verfügung gestellt wird. Klickt der Redakteur dieses Bedienelement an oder zieht er Objekte mit dem Mauszeiger darauf ("Drop"), wird das referenzierte Skript bzw. die Klasse ausgeführt. Auf diese Weise kann mit `FS_BUTTON` ein Icon oder eine Schaltfläche erstellt werden, die eine individuell implementierbare Funktion ausführt.

Die Eingabekomponente kann mit anderen Eingabekomponenten zusammenarbeiten. Auf diese Weise können z. B. Eingaben des Redakteurs für das Skript verwendet werden.

Zur Konfiguration von `FS_BUTTON` siehe *FirstSpirit Online Dokumentation, Bereich*



"Vorlagenentwicklung" / "Formulare" / "Eingabekomponenten (neu)" / "FS_BUTTON".

Wird FS_BUTTON für die Applikationsintegration eingesetzt, wird also das AppCenter genutzt, ist eine entsprechende Lizenz erforderlich (siehe Kapitel 6.5.2 Seite 145).



Die Eingabekomponente FS_BUTTON wird aktuell im FirstSpirit-WebClient nicht unterstützt. Eine entsprechende Unterstützung ist für FirstSpirit Version 5 geplant.

5.5 CMS_INCLUDE_OPTIONS: Auswahlmenge einschränken

Mithilfe des Datenelements CMS_INCLUDE_OPTIONS ist es möglich, die Wertemenge, die in den Eingabekomponenten

- CMS_INPUT_CHECKBOX
- CMS_INPUT_COMBOBOX
- CMS_INPUT_RADIOBUTTON und
- CMS_INPUT_LIST

zur Auswahl angezeigt werden sollen, dynamisch zu gestalten. So können diese Eingabekomponenten z. B. automatisch mit Datensätzen aus einer Fremdtabelle befüllt werden.

Das Tag `QUERY` kann nun dazu verwendet werden, die Menge der Datensätze anhand einer für das betreffende Datenbank-Schema definierten Abfrage einzuschränken. Über den Parameter `name` wird der Referenzname der gewünschten Abfrage angegeben, z. B.

```
<CMS_INPUT_COMBOBOX name="BEZEICHNER" useLanguages="no">
  <CMS_INCLUDE_OPTIONS type="DATABASE">
    <LABELS>
      <LABEL lang="*">#item.SPALTE</LABEL>
      <LABEL lang="DE">#item.SPALTE_DE</LABEL>
    </LABELS>
    <TABLE>EINDEUTIGER_NAME_TABELLENVORLAGE</TABLE>
    <QUERY name="REFERENZNAME_ABFRAGE" />
  </CMS_INCLUDE_OPTIONS>

  <LANGINFOS>
    <LANGINFO lang="*" label="TEXT" description="TEXT"/>
    <LANGINFO lang="DE" label="TEXT" description="TEXT"/>
  </LANGINFOS>
</CMS_INPUT_COMBOBOX>
```



```
</LANGINFOS>  
</CMS_INPUT_COMBOBOX>
```

Sind für die über `QUERY` definierte Abfrage Parameter vorhanden (Register "Parameter" der Abfrage), kann mit dem Tag `PARAM` auf diese Parameter zugegriffen werden. Der Name des Abfrage-Parameters wird mithilfe des Parameters `name` angegeben, der Wert des Abfrage-Parameters innerhalb eines öffnenden und schließenden `PARAM`-Tags, z. B.

```
<QUERY name="REFERENZNAME_ABFRAGE ">  
  <PARAM name="PARAMETERNAME">WERT</PARAM>  
</QUERY>
```

5.6 Funktionserweiterungen der Integrierten Vorschau

Die in FirstSpirit Version 4.2 im JavaClient eingeführte "Integrierte Vorschau" kann ab 4.2R4 nicht nur wie bisher für eine Vorschau der aktuell bearbeiteten Inhalte verwendet werden, sondern es können (bei korrekter Konfiguration (siehe Kapitel 4.2.2.3 Seite 41)) auch andere Dateiformate (z. B. Microsoft Word- oder Excel-Dateien) siehe Kapitel 4.2.2.1 Seite 38) und die FirstSpirit Online Hilfe (siehe Kapitel 4.2.2.4 Seite 43) nun dort dargestellt werden, und zwar mithilfe einer zum Format passenden Anwendung.

Für Vorlagenentwickler ist besonders auf die Anzeige der *FirstSpirit Online Dokumentation* (ODFS) in der Integrierten Vorschau hinzuweisen. Dabei wird innerhalb von Formularen und Ausgabekanälen der aktuelle Kontext für die Anzeige einer passenden Seite im ODFS verwendet. Befindet man sich beispielsweise in einem Formular auf einem `CMS_INPUT_TEXT`-Tag, wird beim Aufruf der Hilfe (z. B. über F1) direkt die Seite "CMS_INPUT_TEXT" (Bereich *Vorlagenentwicklung / Formulare / Eingabekomponenten* im ODFS angezeigt, beim Aufruf auf einem `$(CMS_REF(...))`-Tag im Ausgabekanal die Seite "`$(CMS_REF(...))`" (Bereich *Vorlagenentwicklung / Vorlagensyntax / Anweisungen*). Wird Text im Formular- oder einem Ausgabekanal-Register markiert und dann die Hilfe aufgerufen, wird automatisch eine Suche im ODFS mit dem markierten Text durchgeführt und die Ergebnisse wie gewohnt dargestellt. Aus der Online Hilfe können darüber hinaus Absätze (z. B. Beispiel-Code) einfach per Drag & Drop auf den Formular-Reiter von Vorlagen gezogen werden

Weitere Informationen zu diesem Thema siehe auch Kapitel 4.2.2 Seite 35.



5.7 Inline-Tabellen: CMS_INPUT_NUMBER in Stilvorlagen

Seit der Version 4.1 können für den JavaClient so genannte Inline-Tabellen definiert werden (DOM-Editor mit `table="yes"`). Diese basieren u.a. auf Stilvorlagen, über die Eingabekomponenten zur Pflege von Layout-Attributen angelegt werden. Bislang wurden dabei nur folgende Eingabekomponenten unterstützt:

- CMS_INPUT_TEXT / CMS_INPUT_TEXTAREA: Textfeld für die Angabe eines Wertes, z. B. für die Hintergrundfarbe einer Zelle.
- CMS_INPUT_COMBOBOX: Auswahl aus einer vorgegebenen Menge von Werten, z. B. für die Angabe einer Hintergrundfarbe oder der Text-Ausrichtung.
- CMS_INPUT_RADIOBUTTON: Auswahl aus einer vorgegebenen Menge von Werten, z. B. für die Angabe einer Hintergrundfarbe oder einer Text-Ausrichtung.
- CMS_INPUT_TOGGLE: Auswahl zwischen zwei möglichen Werten (z. B. an/aus, rechts/links) .

Seit 4.2R4 kann auch die Eingabekomponente CMS_INPUT_NUMBER verwendet werden. In dieser kann z. B. der Wert für die Hintergrundfarbe einer Zelle angegeben werden.



In diesem Rahmen wurde auch eine so genannte Sichtbarkeit von bestimmten Eingabekomponenten für Stilvorlagen eingeführt (siehe dazu Kapitel 5.8 Seite 109).

5.8 Sichtbarkeit von Eingabekomponenten einschränken

Im Rahmen der Änderungen zu Kapitel 5.7 wurde mit 4.2R4 auch eine so genannte Sichtbarkeit von bestimmten Eingabekomponenten für bestimmte Vorlagentypen eingeführt. So können auf dem Register "Formular" von Stilvorlagen z. B. nur noch die folgenden Tags / Eingabekomponenten verwendet werden:

- CMS_COMMENT
- CMS_GROUP
- CMS_INPUT_COMBOBOX
- CMS_INPUT_NUMBER
- CMS_INPUT_RADIOBUTTON
- CMS_INPUT_TEXT
- CMS_INPUT_TEXTAREA



- CMS_LABEL
- CMS_MODULE

Stilvorlagen, für die ein davon abweichender Tag verwendet wird, können nicht gespeichert werden. Es erscheint folgender Hinweis: "Ungültige GOM-Benutzung! Tag 'CMS_INPUT_CONTENTAREALIST' not allowed in this kind of template! Editor 'xyz'".

Auch das Speichern von ungültigen Formularen in Tabellen- und Verweissvorlagen wurde in diesem Rahmen angepasst: für diese können nun alle Komponenten außer CMS_INPUT_SECTIONLIST verwendet werden.

In eigenen Eingabekomponenten und Modulen ist die Sichtbarkeit über das Tag `<restrictions>` (bei Verwendung des `<editor>`-Tags) bzw. `<scope>` (in Verbindung mit so genannten Gadgets) in der Datei `fs-module.xml` entsprechend anzupassen, da diese sonst in Links und Stilvorlagen ohne Änderung abgewiesen werden. Dazu werden folgende Werte verwendet:

- *DATA*: Die Eingabekomponente kann in Seiten- und Absatzvorlagen verwendet werden.
- *CONTENT*: Die Eingabekomponente kann in Tabellenvorlagen verwendet werden.
- *LINK*: Die Eingabekomponente kann in Verweissvorlagen verwendet werden.
- *STYLE*: Die Eingabekomponente kann in Stilvorlagen verwendet werden.

Beispiele:

```
<restrictions>
  <kind>DATA</kind>
  <kind>CONTENT</kind>
  <kind>LINK</kind>
</restrictions>
```

bzw.

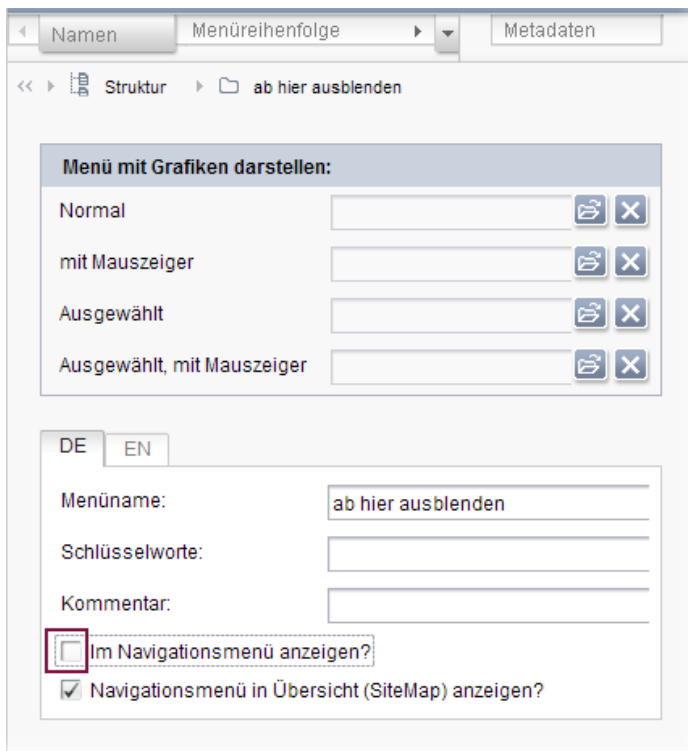
```
<scope data="yes" content="yes" link="yes" style="no"/>
```

Eine Eingabekomponente mit dieser Konfiguration wäre in Seiten-, Absatz-, Tabellen- und Verweissvorlagen, nicht aber in Stilvorlagen verwendbar. Wird für eine Eingabekomponente nichts angegeben, so ist sie für die Scopes *DATA* und *CONTENT* zugelassen.



5.9 Navigation: `parentpath` blendet Menüebenen bei Bedarf aus

Während mit dem Schlüsselbegriff `purepath` zum Parameter `expansionVisibility` der Navigation-Funktion (siehe dazu *FirstSpirit Online Dokumentation, Vorlagenentwicklung / Vorlagensyntax / Funktionen / im Header / Navigation*) immer der gesamte Pfad zum aktuellen Menüpunkt unabhängig von der Sichtbarkeit dargestellt wird, können mit dem in 4.2R4 eingeführten Schlüsselbegriff `parentpath` Menüebenen aus dem Pfad ausgeblendet werden. Und zwar werden in diesem Modus Menüebenen (Ordner in der Struktur-Verwaltung) berücksichtigt, bei denen auf dem Register "Namen" die Option "Im Navigationsmenü anzeigen?" deaktiviert ist:



The screenshot shows a configuration window for a menu. At the top, there are tabs for 'Namen', 'Menüreihenfolge', and 'Metadaten'. Below the tabs, there is a breadcrumb trail: '<< >> Struktur > ab hier ausblenden'. The main content area is titled 'Menü mit Grafiken darstellen:' and contains four rows of options, each with a text input field, a refresh icon, and a delete icon (X):

- Normal
- mit Mauszeiger
- Ausgewählt
- Ausgewählt, mit Mauszeiger

Below this section, there are language tabs for 'DE' and 'EN'. Under the 'DE' tab, there are three text input fields:

- Menüname: ab hier ausblenden
- Schlüsselworte:
- Kommentar:

At the bottom, there are two checkboxes:

- Im Navigationsmenü anzeigen? (This checkbox is highlighted with a red box in the original image)
- Navigationsmenü in Übersicht (SiteMap) anzeigen?

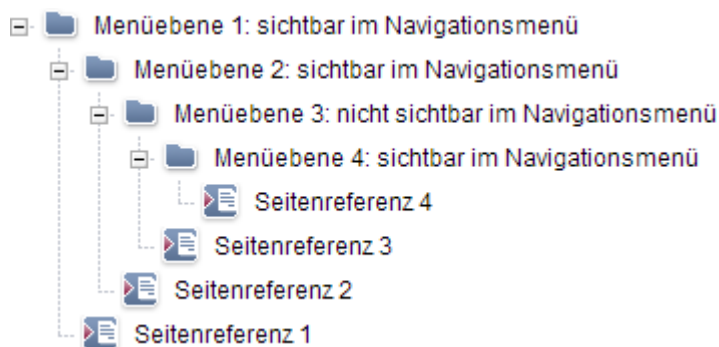
Abbildung 5-5: Aus dem Navigationsmenü ausgeblendete Menüebene

Wird eine Menüebene auf diese Weise aus dem Navigationsmenü herausgenommen und ist `expansionVisibility="parentpath"` gesetzt, wird der Pfad ausgehend vom Wurzelknoten (Struktur-Verwaltung) nur bis zur letzten sichtbaren Menüebene ausgegeben. D. h., die erste Menüebene vom Wurzelknoten ausgehend mit deaktivierter Option "Im Navigationsmenü anzeigen?" sowie alle darunter liegenden Menüebenen werden nicht im Pfad angezeigt.



Beispiel:

Von einer Struktur wie



wären die Menüebenen "Menüebene 1" und "Menüebene 2" sichtbar.

5.10 Tagging

Bereits seit Version 4.0 bietet FirstSpirit die Möglichkeit, Revisionen mit so genannten Tags zu versehen. Dabei handelt es sich um die Auszeichnung einer Revision. Ab FirstSpirit Version 4.2R4 werden Tags bei folgenden Operationen automatisch erzeugt:

- bei der Aktualisierung von Vorlagen
- beim Import eines Pakets (FirstSpirit Paket-Verwaltung)
- bei der Generierung

Allerdings sind Tags bisher nur für Entwickler von Bedeutung und spiegeln sich (noch) nicht in der FirstSpirit-Software-Oberfläche wider. Beispielsweise können Tags nach folgendem Schema in einem Skript über die API erzeugt werden:

```
Tag Project#createTag(String tag, String comment)
```

Die Tags eines Projekts können folgendermaßen angezeigt werden:

```
List<Tag> Project#getTags(Tag maxTag, int maxCount)
```

Die Revision zu einem Tag kann folgendermaßen ermittelt werden:

```
de.espirit.firstspirit.storage.Tag  
Revision getRevision()
```

Damit kann beispielsweise ein Generierungsskript erstellt werden, das zu einem Tag die Revision ermittelt und diese dann generiert.



Mit

```
setCreateGenerateTag(...)
```

kann das Tagging für die Generierung ein- bzw. ausgeschaltet werden, mit

```
getCreateGenerateTag()
```

kann abgefragt werden, ob Tags gesetzt werden.

Für zukünftige FirstSpirit-Versionen ist geplant, Tags in der Software-Oberfläche zu visualisieren (z. B. in der Versionshistorie) und möglicherweise redaktionelle Aktionen auf Tags zu realisieren.

5.11 Aufruf von Klassen aus Modulen

In (BeanShell-)Skripten können eigene Klassen und Methoden aus FirstSpirit-Modulen aufgerufen werden. Zusätzlich besteht in der FirstSpirit Version 4.2 die Möglichkeit, Klassen direkt auszuführen. Hierfür muss die Klasse das Interface `de.espirit.firstspirit.access.script.Executable` implementieren.

Die eigene Klasse muss dazu Implementierungen zu den Methoden `execute(Map<String, Object> context)` und `execute(Map<String, Object> context, Writer out, Writer err)` vorsehen (siehe FirstSpirit-API-Dokumentation).

Ist die Klasse über den Modul-Classloader auffindbar (d. h. als Bibliothek installiert), so kann in einem Skript in FirstSpirit die Klasse mit folgendem Aufruf ausgeführt werden:

```
-----  
#!executable-class  
NAME_DER_KLASSE  
-----
```

z. B.

```
-----  
#!executable-class  
de.test.meineKlasse  
-----
```





Der Klassenname muss in der zweiten Zeile angegeben werden.

Wird `executable-class` beispielsweise in einem Auftragskript verwendet, so können Übergabeparameter hinterlegt werden (in den Projekteigenschaften, "Auftragsverwaltung", Aktivität "Skript ausführen", Button "Eigenschaften" / "Parameter", siehe auch *FirstSpirit Handbuch für Administratoren*).

Auf diese Parameter kann in den `execute`-Methoden (siehe oben `Map<String, Object>`) zurückgegriffen werden.

Vorteil dieser Ausführungsart ist, dass der BeanShell-Interpreter nicht ausgeführt werden muss. Somit kann in vielen Fällen eine Performanzsteigerung (z. B. bei der Generierung) erzielt werden.

Innerhalb von FirstSpirit wird diese Funktionalität bereits an einigen Stellen eingesetzt. Ein Beispiel für die Verwendung ist das FirstSpirit-Modul "Portal SAP EP BP". Hier wird die Funktionalität im Skript "Portal_createLink" (siehe auch Kapitel 7.1.2 Seite 165) verwendet.

5.12 Neue bzw. geänderte Tastaturkürzel

In der Vorlagen-Verwaltung können nun Zeilen auf Formular- und Ausgabekanal-Registern mit Strg + Shift + D entfernt werden, zuvor wurde dazu Strg + Y verwendet. Mit Strg + Y (ebenso wie mit Strg + Shift + Z) können jetzt dagegen Änderungen, die durch Strg + Z rückgängig gemacht wurden, wiederhergestellt werden ("Redo").



5.13 API-Erweiterungen

In FirstSpirit Version 4.2R4 wurden im Vergleich zu 4.2R2 keinerlei inkompatiblen Änderungen an der öffentlichen API vorgenommen.

5.13.1 FirstSpirit Access-API

Einige Methoden wurden in FirstSpirit Version 4.2R4 auf den Status „deprecated“ gesetzt. Der FirstSpirit Access-API kann jeweils entnommen werden, um welche Methoden es sich handelt und welche Methoden als Ersatz verwendet werden sollen. Methoden mit diesem Status können zwar verwendet werden, da sie aber mit FirstSpirit Version 5.0 entfallen werden, wird von der Verwendung abgeraten.

Darüber hinaus wurden u.a. folgende API-Erweiterungen vorgenommen:

- Über die Revisions-Metadaten sind nun Veränderungen an ausgehenden **Referenzen** zu einer Revision verfügbar (siehe dazu auch *FirstSpirit Handbuch für Entwickler (Teil 1: Grundlagen)*, Kapitel "Änderungsverfolgung über Revisions-Metadaten"). Dazu können die Methoden des Interface `de.espirit.firstspirit.storage.ReferencesChangeDetail` genutzt werden.
- Wird ein **Auftrag**, für den bei "Parallele Ausführung" in den Projekteigenschaften unter "Auftragsverwaltung" / Schaltfläche "Konfigurieren" die Option "Nicht erlaubt (nacheinander ausführen)" gewählt ist, mehrfach gestartet, kann die Liste der noch nicht ausgeführten Aufträge über

```
de.espirit.firstspirit.access.schedule.ScheduleStorage#getQueuedEntries()
```

zurückgeliefert werden.

- Die **Rechteauswertung** bei Verwendung des Interface `de.espirit.firstspirit.access.AdminService` wurde mit 4.2R4 überarbeitet: Es gibt nun explizite Prüfungen, ob die jeweilige Aktion durch den aufrufenden Benutzer aufgrund seiner Rechte durchgeführt werden darf. Dabei werden zwei Aktions- bzw. Benutzertypen unterschieden:
 - Aktionen, für die Rechte eines Projekt-Administrators benötigt werden
 - Aktionen, für die Rechte eines Server-Administrators benötigt werdenHat der Benutzer nicht die erforderlichen Rechte, wird eine `SecurityException` ausgegeben. In der API sind Methoden, für die Projekt- oder Server-Administratorrechte erforderlich sind, mit einem entsprechenden Hinweis auf eine `SecurityException` ausgezeichnet. Die Methode `createLanguage` des Interface `de.espirit.firstspirit.access.AdminService` erfordert



beispielsweise Server-Administratorrechte.

- Mit 4.2R4 wurden die **Anlege-Operationen** für IDProvider-Objekte konsolidiert, so dass es nun möglich ist, bereits beim initialen Anlegen den Anzeigenamen mit zu übergeben. Vorteile:
 - Die Anlege-Operation ist eine locklose Operation, d. h. das Element wird nicht in den Bearbeitungszustand versetzt. Zur Vergabe von Anzeigenamen war es bisher nötig, nach der locklosen Anlege-Operation noch eine lockbasierte Änderung durchzuführen. Mit den neuen Methoden kann der Anzeigename bereits bei der locklosen Anlege-Operation mit übergeben werden.
 - Die zusätzliche Revision und zugehörige Daten, die durch das nachträgliche Setzen des Anzeigenamens erzeugt wurden, werden eingespart.

Wurde bisher ein Ordner in der Medien-Verwaltung mit einem Anzeigenamen für Deutsch und Englisch beispielsweise folgendermaßen angelegt:

```
MediaStore ms;
MediaFolder newFolder = ms.createMediaFolder("myfolder", true);
try {
    newFolder.setLock(true, false);
    // set displayname for german
    LanguageInfo deLangInfo = newFolder.getLanguageInfo(germanLanguage);
    deLangInfo.setDisplayName("mein Ordner");

    // set displayname for english
    LanguageInfo enLangInfo = newFolder.getLanguageInfo(englishLanguage);
    enLangInfo.setDisplayName("my folder");

    newFolder.save("display names defined", false);
} finally {
    try {
        myLockFolder.setLock(false, false);
    } catch (Exception unlockFailed){
        // error handling
    }
}
```

kann dies ab 4.2R4 folgendermaßen umgesetzt werden:

```
MediaStore ms;
Map<Language, String> lang2DisplayName = new HashMap<Language,
String>();
lang2DisplayName.put(germanLanguage, "mein Ordner");
lang2DisplayName.put(englishLanguage, "my folder");
MediaFolder newFolder = ms.createMediaFolder("myfolder",
lang2DisplayName, true);
```

5.13.2 FirstSpirit Developer-API

Mit FirstSpirit Version 4.2 Release 4 wird erstmalig eine Developer-API ausgeliefert. Im Gegensatz zur Access-API (siehe Kapitel 5.13.1 Seite 115) unterliegt die Developer-API geringeren Stabilitätsauflagen: Während die Access-API innerhalb einer Major-Versionslinie stabil ist, ist die Developer-API nur innerhalb einer **Minor-**



Versionslinie stabil, d. h. dass Änderungen bei einem Minor-Versionswechsel durchgeführt werden dürfen.

Die Dokumentation zur FirstSpirit Developer-API kann z. B. über die Seite "Vorlagenentwicklung" / "FirstSpirit API" / "API-Dokumentation" in der *FirstSpirit Online Dokumentation* geöffnet werden.

5.13.3 FirstSpirit Application-API

Mit FirstSpirit Version 4.2 Release 4 wird darüber hinaus eine erste Version der Application-API zur Verfügung gestellt, die es ermöglicht, völlig eigenständig individuelle Applikationsintegrationen für FirstSpirit zu realisieren. Diese ist aus den praktischen Erfahrungen der Applikationsintegration des Kernproduktes hervorgegangen und hat sich damit bereits in der Praxis bewährt.

In Verbindung mit der ebenfalls schon lange im Produkt FirstSpirit etablierten Erweiterungs-Schnittstelle (FirstSpirit-Module) können so Produkterweiterungen entwickelt werden, die sich nahtlos in die FirstSpirit Applikationsintegrations-Infrastruktur eingliedern.

Das eigentliche Potenzial der Applikationsintegration liegt ganz besonders im Bereich der FirstSpirit Entwicklungspartner, die mithilfe der Applikationsintegrations-API ein mächtiges Werkzeug zur Verfügung gestellt bekommen, um spezialisierte Integrationslösungen in ihren besonderen Kompetenzbereichen zu realisieren.

An dieser Stelle soll aber auch nicht verschwiegen werden, dass die Applikationsintegration ein hochkomplexes Thema ist, das auch nach mehr als zweijähriger Entwicklungszeit immer noch nicht vollständig erschlossen ist, da sich mit den gemachten Erfahrungen immer neue Anwendungsfelder eröffnen.

Die Entwicklung im Bereich der FirstSpirit Applikationsintegration ist also auch auf der Ebene des Kernproduktes noch lange nicht abgeschlossen, sondern befindet sich noch im vollen Gange. Dennoch wurde von e-Spirit die Entscheidung getroffen, bereits zu einem möglichst frühen Zeitpunkt eine Schnittstelle für die Applikationsintegration freizugeben, um es Partnern so schnell wie möglich zu erlauben, eigene Applikationsintegrationen zu realisieren.

Mit dieser Entscheidung für eine frühe API-Freigabe ging aber auch die Notwendigkeit einher, die eigentlich bei der Entwicklung von FirstSpirit explizit definierte Maxime zur langfristigen Wahrung der Schnittstellen-Stabilität in diesem Bereich aufzuweichen: Die mit FirstSpirit Version 4.2 Release 4 zur Verfügung gestellte Schnittstelle wird in den folgenden Versionen weiterentwickelt, und es wird sich mit sehr hoher Wahrscheinlichkeit die Notwendigkeit ergeben, dass



Veränderungen an der Modul-Implementierung vorgenommen werden müssen. Dieser Tatsache sollte sich jeder Entwicklungspartner bewusst sein, der auf der Basis der Applikationsintegrations-API von FirstSpirit Version 4.2 Release 4 eine Implementierung vornimmt!

Auf der anderen Seite bietet sich so die Gelegenheit, auf zusätzliche, bisher noch nicht angedachte Einsatzszenarien der Applikationsintegration auf der API-Ebene flexibel zu reagieren und diese optimal zu unterstützen.

Mit FirstSpirit Version 5.0 wird dann eine überarbeitete und den Kriterien der Langzeitstabilität genügende Version 2.0 der API für die Applikationsintegration zur Verfügung stehen, die dann nach den bekannten Strategien der Pflege stabiler APIs in FirstSpirit gewartet und weiterentwickelt wird.



6 Neue/geänderte Funktionen für Administratoren

6.1 Erstellen und Verwenden von JDBC-Treiber-Modulen

FirstSpirit speichert die stark strukturierten Inhalte der Datenquellen-Verwaltung in einer Datenbank ab, um effiziente, komplexe Suchanfragen auf diesen Daten zu ermöglichen.

Bis zur FirstSpirit-Version 4.2R2 einschließlich war es der in den meisten Fällen übliche Weg, die erforderlichen JDBC-Treiberdateien in den FirstSpirit-Server zu integrieren, indem sie im Verzeichnis `.../shared/lib` abgelegt wurden.

Ab FirstSpirit-Version 4.2R4 wird die Einbindung von JDBC-Treiberdateien als FirstSpirit-Modul empfohlen. Dies hat z. B. den Vorteil, dass verschiedene Versionen eines Treibers nebeneinander betrieben werden können. Darüber hinaus können die JDBC-Treiber im laufenden Betrieb des FirstSpirit-Servers ohne Neustart ausgetauscht werden. Aus lizenzrechtlichen Gründen werden diese FirstSpirit-Module nicht mit der Software ausgeliefert und müssen von jedem Kunden selbst erstellt werden. Dieses Kapitel erläutert, wie vorgegangen werden kann, um die Umstellung von der Verwendung von JDBC-Treibern im Verzeichnis `.../shared/lib` hin zur Einbindung als FirstSpirit-Modul vorzunehmen.

Dabei wird auch die Verwendung von Datenbanken

- auf dem in FirstSpirit integrierten Jetty-Web-Server,
- auf einem (externen) Tomcat-Web-Server
- mit Jetty- und Tomcat-Web-Server

sowie

- die Verwendung der Derby-Datenbank in Web-Anwendungen (z. B. FirstSpirit-Modul Integration)

thematisiert.



Die Verwendung der JDBC-Treiberdateien im Verzeichnis `.../shared/lib` ist auch nach FirstSpirit-Version 4.2R2 weiterhin möglich, aber nicht mehr empfohlen.





Liegen verschiedene Datenbank-Treiber unter `.../shared/lib` und als Modul vor, werden vorrangig die Treiber verwendet, die unter `.../shared/lib` liegen.



Informationen zur Modul-Entwicklung in FirstSpirit allgemein bietet das "FirstSpirit-Entwicklerhandbuch für Komponenten", weiterführende Informationen zur Konfiguration von Datenbanken, Webservern und zur Installation bzw. Aktualisierung von FirstSpirit-Modulen bietet die "FirstSpirit-Dokumentation für Administratoren".

6.1.1 Erstellen eines JDBC-Treiber-Moduls

Um ein JDBC-Treiber-Modul zu erstellen, werden folgende Dateien benötigt:

- Datei(en) des JDBC-Treibers, z. B. JAR-Datei, Lizenzen usw. (siehe Kapitel 6.1.1.1 Seite 120)
- `module.xml`-Datei (siehe Kapitel 6.1.1.2 Seite 121)

optional:

- `web.xml`-Datei (siehe Kapitel 6.1.1.2 Seite 121)

Diese Dateien werden in einer FSM-Datei (ZIP-Archiv) zu einem FirstSpirit-Modul zusammengefasst (siehe Kapitel 6.1.1.3 Seite 126).

6.1.1.1 Treiber-Dateien

Wurde bisher nicht mit einem Modul für JDBC-Treiber gearbeitet, befinden sich die relevanten Dateien des JDBC-Treibers, die zur Erstellung des Moduls benötigt werden, entweder im Verzeichnis `.../shared/lib` des FirstSpirit-Servers oder im entsprechenden Verzeichnis der Servlet-Engine (z. B. externer Tomcat-Web-Server).





Für Oracle sollte der JDBC-Treiber der Serie 10.1 (`ojdbc14_10.1.0.x.jar`) verwendet werden, da bei der Version 10.2 Probleme bei Verwendung des Datentyps `LONG` ab einer Länge von 4.000 Zeichen bei UTF-8-Kodierung vorkommen können. Alternativ muss bei Verwendung des 10.2-Treibers der Kompatibilitätsmodus für Oracle 9 `LONG` aktiviert werden, da `LONG` seit Oracle 9 deprecated ist. Dazu muss der Parameter

```
jdbc.property.oracle.jdbc.RetainV9LongBindBehavior=true
```

in der Datenbankkonfiguration ergänzt werden.

6.1.1.2 module.xml und web.xml

Die Datei `module.xml` enthält die Definition des Treiber-Moduls und muss nach dem folgenden Beispiel aufgebaut sein. Das Grundgerüst ist dabei immer gleich, einige Tags und Parameter variieren je nach eingesetzter Datenbank und -version.

Das folgende Beispiel stellt exemplarisch den Aufbau einer `module.xml` für eine **PostgreSQL 8.2**-Datenbank dar:

```
<!DOCTYPE module SYSTEM "../lib/module.dtd">
<module>
  <name>PostgreSQL_JDBC_Driver_8_2</name>
  <version>8.2.508</version>
  <description>JDBC Driver for PostgreSQL 8.2 databases</description>
  <vendor>PostgreSQL Global Development Group</vendor>
  <components>

    <public>
      <name>PostgreSQL_JDBC_Driver_8_2_Server</name>
      <description>Provides the JDBC Driver for the FirstSpirit
server.</description>
      <class>org.postgresql.Driver</class>
      <resources>
        <resource scope="module">lib/postgresql-8.2-
508.jdbc3.jar</resource>
      </resources>
    </public>

    <web-app>
      <name>PostgreSQL_JDBC_Driver_8_2_WebApp</name>
      <description>Provides the JDBC Driver in a web
application.</description>
      <web-xml>web.xml</web-xml>
      <web-resources>
        <resource name="postgresql" version="8.2.508"
minVersion="8.2.1" maxVersion="8.2.999">lib/postgresql-8.2-
508.jdbc3.jar</resource>
      </web-resources>
    </web-app>
  </components>
</module>
```



```
</components>

<configuration>
<layerclass>de.espirit.or.impl.postgres.PostgreSQLLayer</layerclass>
  <DRIVER>org.postgresql.Driver</DRIVER>
</configuration>
</module>
```

Das Modul ist zweigeteilt: ein Teil definiert die Ressourcen für den FirstSpirit-Server, die so genannte "Server-Komponente" (innerhalb des `<public>`-Tags), der andere für Web-Anwendungen, die so genannte "Web-Anwendungs-Komponente" (innerhalb des `<web-app>`-Tags). Dieser JDBC-Treiber kann somit im FirstSpirit-Server **und** in Web-Anwendungen genutzt werden. Wird der Treiber nur für den Server benötigt, kann die `<web-app>`-Definition entfallen.



Für den Einsatz der FirstSpirit-Web-Applikationen wird eine Servlet-Engine benötigt, die die Servlet-API in der Version 2.4 implementiert.

`<name>`: Über dieses Tag muss ein eindeutiger Name für die Komponenten vergeben werden. Dabei dürfen nur folgende Zeichen verwendet werden:

Groß- und Kleinbuchstaben (A-Z, a-z), Zahlen (0-9) und die Sonderzeichen ; , _ \ -

Der für die Server-Komponente vergebene Name wird in der Modul-Übersicht des FirstSpirit-Servers verwendet (siehe Abbildung 6-3) und muss auch in der Datenbank-Layer-Konfiguration angegeben werden (siehe Kapitel 6.1.2.2 Seite 127). Der für die Web-Anwendungs-Komponente vergebene Name wird auch in den Projekt-Eigenschaften, Bereich "Web-Komponenten" verwendet (siehe Abbildung 6-7). Für eine bessere Übersicht wurden im Beispiel `_Server` bzw. `_WebApp` an den Namen für die Server- bzw. die Web-Anwendungs-Komponente angehängt.

`<description>`: Über dieses Tag kann eine Beschreibung zur Komponente angegeben werden.

`<class>`: Über dieses Tag muss der vollständige Klassenname des verwendeten JDBC-Treibers angegeben werden.

`<resources>` / `<resource>`: Über diese Tags wird der Pfad zur JAR-Datei des JDBC-Treibers angegeben.



`scope`: Innerhalb des `<resources>` / `<resource>`-Tags sollte für diesen Parameter der Wert *module* angegeben werden. Er sorgt dafür, dass die JAR-Datei für das JDBC-Treiber-Modul und nicht den gesamten Server gilt.

`<webresources>` / `<resource>`: Über diese Tags wird innerhalb der Web-Anwendungs-Komponente der Pfad zur JAR-Datei des JDBC-Treibers angegeben. Folgende Parameter sollten zusätzlich verwendet werden:

`name`: Für die von FirstSpirit unterstützten Datenbanken sollten folgende Standardnamen für die jeweiligen JAR-Dateien verwendet werden:

postgresql (PostgreSQL)

oracle (Oracle)

mssql (Microsoft SQL Server)

mysql (MySQL)

db2 (IBM DB2)

derby (Apache Derby)

`version`: Mit diesem Parameter sollte die vollständige Version des Treibers angegeben werden, also z. B. *8.2.508* für Version 8.2 Build 508.

`minVersion` / `maxVersion`: Mit diesen Parametern sollten die minimale bzw. maximale Version angegeben werden, mit denen der Treiber verwendet werden kann. Im Beispiel bedeutet dies, dass der Treiber von Version 8.2.1 bis 8.2.999 genutzt werden kann. Wird nun ein zweiter Treiber durch ein weiteres Modul zur Verfügung gestellt, z. B. Build 509, so kann dieser auch von 8.2.1 bis 8.2.999 genutzt werden. In diesem Fall wird dann nur der höhere Treiber (also 509) in die Web-Anwendung kopiert bzw. übernommen.

`<configuration>`: Enthält Angaben zur Layer-Klasse und zum Klassennamen des verwendeten JDBC-Treibers.

`<layerclass>`: Über dieses Tag wird die Klasse angegeben, die den Datenbank-Layer für dieses spezielle Datenbanksystem implementiert, beispielsweise:

```
<layerclass>de.espirit.or.impl.postgres.PostgreSQLLayer</layerclass>
```

für PostgreSQL oder

```
<layerclass>de.espirit.or.impl.oracle.OracleLayer</layerclass>
```



für Oracle.

<DRIVER>: Enthält den vollständigen Klassennamen des verwendeten JDBC-Treibers, beispielsweise `org.postgresql.Driver` für PostgreSQL.

Soll der JDBC-Treiber in einer Web-Anwendung zur Verfügung stehen, ist die Datei `web.xml` erforderlich:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app id="PostgreSQL_JDBC_Driver_8_2"
  version="2.4"
  xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"/>
```

Der Wert des Parameters `id` sollte den Namen des JDBC-Moduls (Server-Komponente) angeben.

Soll die integrierte **Derby-Datenbank** in den Web-Anwendungen eines Tomcat-Web-Servers verwendet werden, wird ebenfalls eine `module.xml`-Datei benötigt. Eine exemplarische `module.xml` könnte so aussehen:

```
<!DOCTYPE module SYSTEM "../lib/module.dtd"> <module>
  <name>Derby_JDBC_Driver_10_2</name>
  <version>10.2.2.0</version>
  <description>JDBC Driver for Derby 10.2 databases</description>
  <vendor>Apache Software Foundation</vendor>
  <components>

  <public>
    <name>Derby_JDBC_Driver_10_2_Server</name>
    <description>Provides the JDBC Driver for the FirstSpirit
server.</description>
    <class>org.apache.derby.jdbc.ClientDriver</class>
    <resources>
      <resource
scope="module">lib/derbyclient.jar</resource>
    </resources>
  </public>

  <web-app>
    <name>Derby_JDBC_Driver_10_2_WebApp</name>
    <description>Provides the JDBC Driver in a web
application.</description>
    <web-xml>web.xml</web-xml>
    <web-resources>
      <resource name="derby" version="10.2.2.0"
minVersion="10.2.0"
maxVersion="10.2.999">lib/derbyclient.jar</resource>
    </web-resources>
  </web-app>
```



```
</components>

<configuration>
  <layerclass>de.espirit.or.impl.derby.DerbyLayer</layerclass>
  <DRIVER>org.apache.derby.jdbc.ClientDriver</DRIVER>
</configuration>
</module>
```

Der Server-Komponententeil ist nur dann notwendig, wenn sowohl der interne Jetty- als auch ein externer Tomcat-Web-Server gleichzeitig eingesetzt werden.

Die Datei `web.xml` ist ebenfalls erforderlich. Beispiel:

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app id="Derby_JDBC_Driver_10_2"
  version="2.4"
  xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd" />
```

Wird der Server-Komponententeil benötigt, also interner Jetty- und externer Tomcat-Web-Server gleichzeitig eingesetzt, ist der Datenbank-Layer anzupassen (siehe Kapitel 6.1.2.4.1 Seite 130).



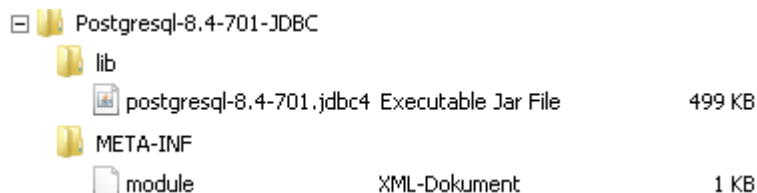
Die in FirstSpirit enthaltene Derby-Datenbank ist nicht für den Produktivbetrieb geeignet und sollte daher lediglich für Tests verwendet werden.

Eine Erläuterung der meisten der hier verwendeten Tags kann auch im *FirstSpirit-Entwicklerhandbuch für Komponenten* nachgeschlagen werden.



6.1.1.3 Verzeichnisstruktur eines JDBC-Treiber-Moduls

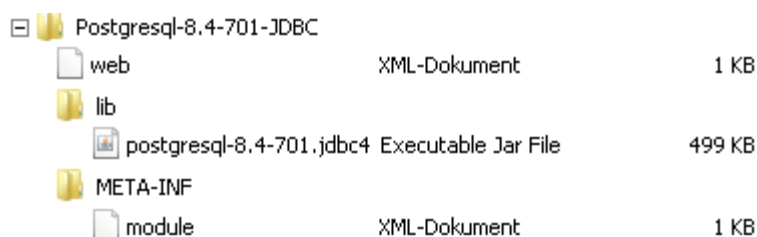
Wird der Treiber nur für den FirstSpirit-Server ("Server-Komponente") verwendet, müssen die Dateien in folgender Verzeichnisstruktur abgelegt werden (vgl. Kapitel 6.1.1.1 und 6.1.1.2):



Postgresql-8.4-701-JDBC		
lib		
postgresql-8.4-701.jdbc4 Executable Jar File		499 KB
META-INF		
module	XML-Dokument	1 KB

Abbildung 6-1: Verzeichnisstruktur Server-Komponente

Bei zusätzlicher Verwendung in Web-Anwendungen muss die Datei `web.xml` auf oberster Ebene eingebunden sein:



Postgresql-8.4-701-JDBC		
web	XML-Dokument	1 KB
lib		
postgresql-8.4-701.jdbc4 Executable Jar File		499 KB
META-INF		
module	XML-Dokument	1 KB

Abbildung 6-2: Verzeichnisstruktur Server- und Web-Anwendungs-Komponente

Um nun ein gültiges FirstSpirit-Modul zu erhalten, muss aus den Inhalten des übergeordneten Ordners ("Postgresql-8.4-701-JDBC") eine ZIP-Datei erstellt werden. Der übergeordnete Ordner darf nicht in der ZIP-Datei enthalten sein. Diese ZIP-Datei muss dann in `*.fsm` umbenannt werden. Würde im Beispiel von Abbildung 6-2 der Ordnername als Dateiname übernommen, sollte die Modul-Datei schließlich `Postgresql-8.4-701-JDBC.fsm` heißen.

6.1.2 Installation und Konfiguration des JDBC-Treiber-Moduls

6.1.2.1 Installation des JDBC-Treiber-Moduls

Wurde das JDBC-Treiber-Modul erfolgreich wie in Kapitel 6.1.1 ab Seite 120 beschrieben erstellt, muss es auf dem FirstSpirit-Server installiert werden. Dies wird über die Anwendung zur Server- und Projektkonfiguration vorgenommen.



Dazu wird in den Server-Eigenschaften im Bereich "Module" der Button "Installieren" angeklickt. Im folgenden Dialog kann die Treiber-Modul-Datei aus dem lokalen Dateiverzeichnis ausgewählt und hochgeladen werden. Die erfolgreich installierte Datei wird anschließend in der Übersicht als Modul mit Namen (hier: PostgreSQL_JDBC_Driver_8_4-701, siehe Kapitel 6.1.1.2 Seite 121, Tag <name>), Version (Tag <version>) und der enthaltenen Komponente (Tag <components>) angezeigt:

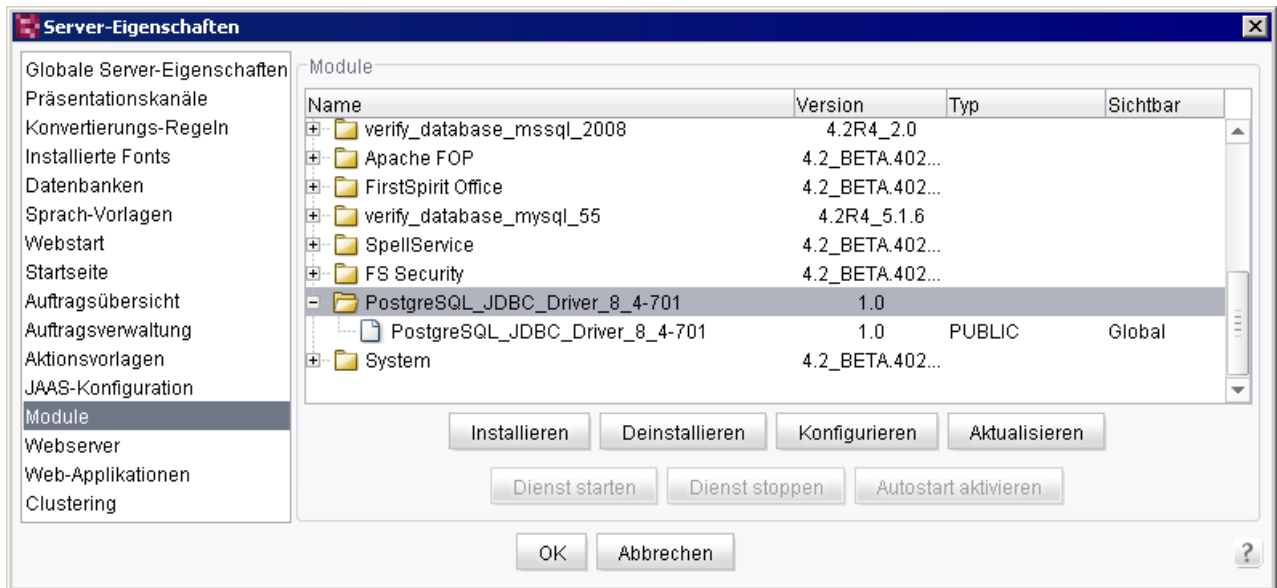


Abbildung 6-3: Server-Eigenschaften – JDBC-Treiber als Modul

Enthält die `module.xml`-Datei eine Definition für eine Web-Anwendung, wird diese hier ebenfalls angezeigt.

Hier braucht keine weitere Konfiguration vorgenommen werden.

6.1.2.2 Datenbank-Layer-Konfiguration

Anschließend muss in der Layer-Konfiguration der Datenbank, für die das Treiber-Modul erstellt wurde, über den Parameter `module` auf dieses Treiber-Modul verwiesen werden.

Dazu wird in den Server-Eigenschaften im Bereich "Datenbanken" die entsprechende Datenbank herausgesucht:



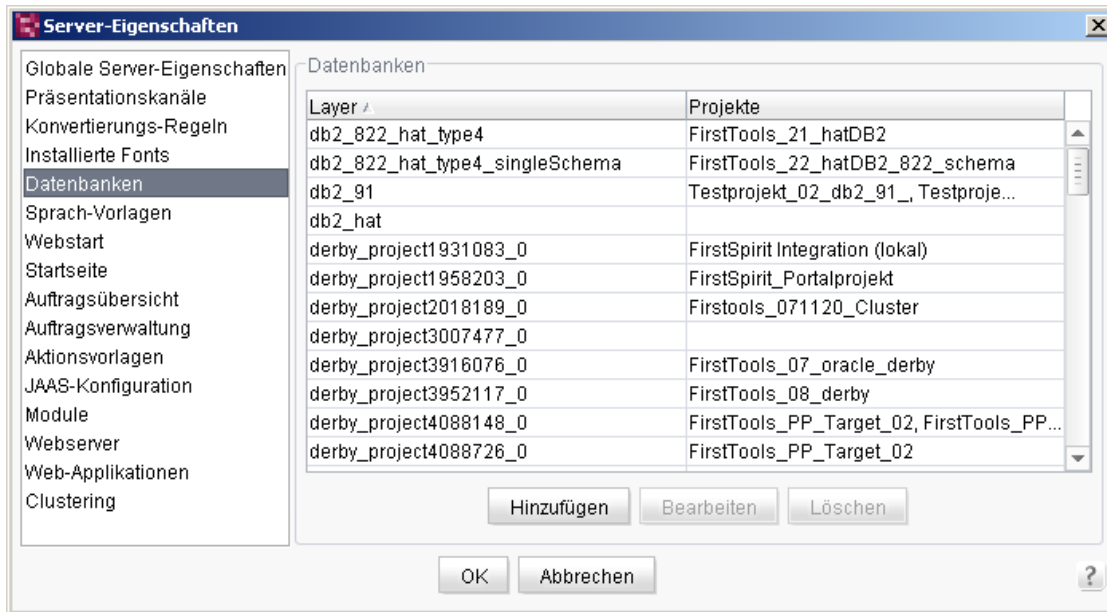


Abbildung 6-4: Server-Eigenschaften – Datenbanken

Mit einem Doppelklick auf den entsprechenden Eintrag öffnet sich der Dialog "Datenbank bearbeiten", in dem die JDBC-Parameter für die Datenbankanbindung bearbeitet werden können:

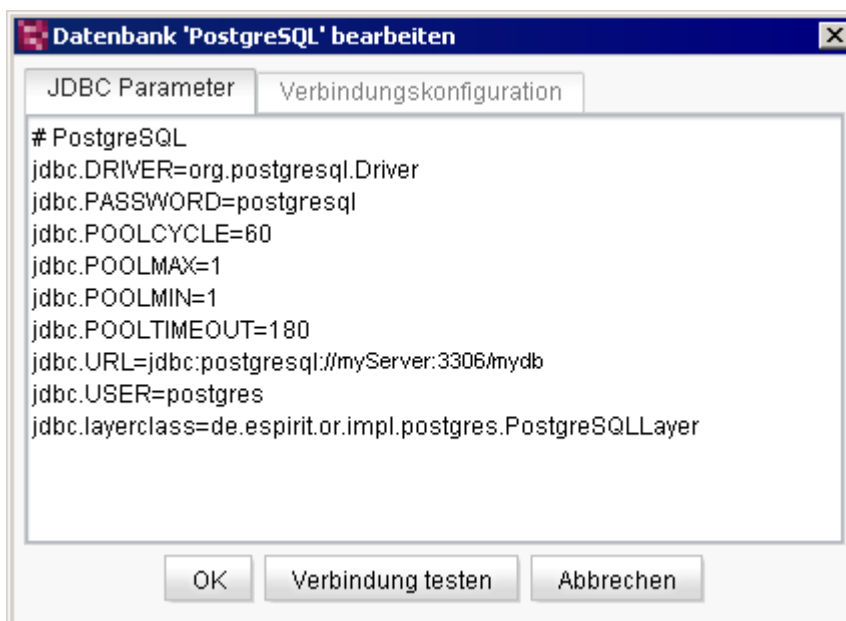


Abbildung 6-5: Datenbank-Konfiguration

Hier muss der Parameter `module` mit dem in der `module.xml`-Datei über `<name>` definierten Namen ergänzt werden, in unserem Beispiel:



```
module=PostgreSQL_JDBC_Driver_8_4-701
```

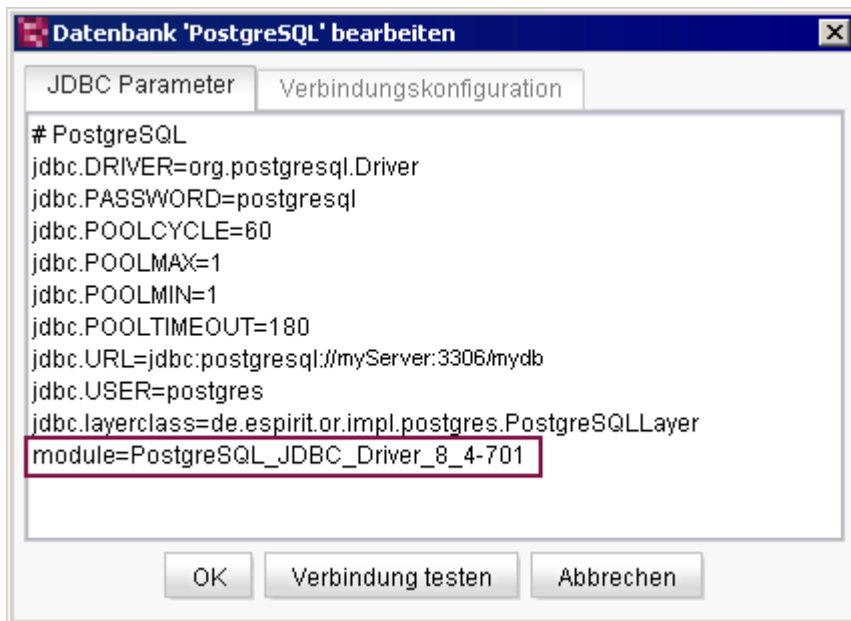


Abbildung 6-6: Datenbank-Konfiguration mit `module`-Parameter

Die Änderung kann nach einem erfolgreichen Verbindungstest mit einem Klick auf "OK" gespeichert werden.

6.1.2.3 Bei Verwendung in Web-Anwendungen

Soll die Datenbank in einer Web-Anwendung verwendet werden, muss das Modul in der gewünschten Web-Komponente hinzugefügt werden. Dies wird in den Projekteigenschaften im Bereich "Web-Komponenten" vorgenommen. Mit einem Klick auf "Hinzufügen" öffnet sich eine Liste, aus der die auf dem Server zur Verfügung stehenden Modul-Komponenten ausgewählt werden können. Hier muss die Komponente des JDBC-Treiber-Moduls ausgewählt werden. Der Name ergibt sich dabei aus dem in der `<web-app>`-Definition über `<name>` definierten Wert (siehe Kapitel 6.1.1.2 Seite 121):



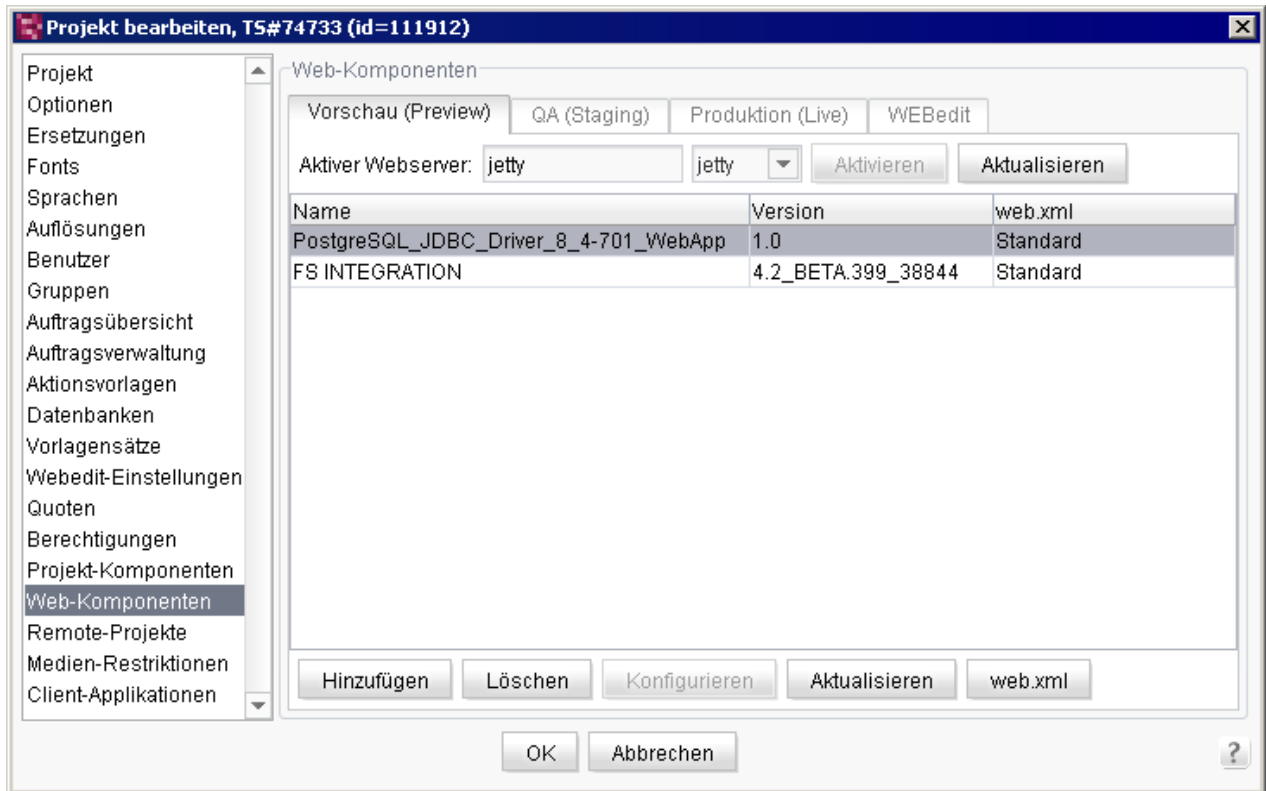


Abbildung 6-7: Projekt-Eigenschaften – JDBC-Treiber als Web-Komponente

Die so hinzugefügte Komponente des JDBC-Treiber-Moduls kann nicht weiter konfiguriert werden.

6.1.2.4 Bei Verwendung der Derby-Datenbank in Web-Anwendungen

Wird eine Derby-Datenbank in Web-Anwendungen (z. B. im FirstSpirit-Modul Integration) verwendet, muss das JDBC-Treiber-Modul ebenfalls, wie in Kapitel 6.1.2.3 Seite 129 beschrieben, der Web-Anwendung hinzugefügt werden.



Die in FirstSpirit enthaltene Derby-Datenbank ist nicht für den Produktivbetrieb geeignet und sollte daher lediglich für Tests verwendet werden.

6.1.2.4.1 Beispiel: Modul "FirstSpirit Integration"

Bei der Verwendung des Moduls FirstSpirit Integration mit einem Tomcat-Web-Server muss für jedes Schema die Verbindungskonfiguration angepasst werden. Der Zugriff kann in diesem Fall nur über den TCP-Port der Derby-Datenbank erfolgen.



Hierzu muss der Parameter `internalDB.port` in der Konfigurationsdatei `fs-server.conf` angegeben werden.

In der Datenbank-Layer-Konfiguration (siehe Kapitel 6.1.2.2 Seite 127) sind zusätzlich folgende Parameter für jedes Schema anzupassen:

`jdbc.URL`: Dieser Parameter muss statt auf ein lokales Verzeichnis auf den TCP-Port der Derby-Datenbank zeigen. Hierfür muss im bestehenden URL nur Host und Port ergänzt und `create` entfernt werden, z. B. wird aus

```
jdbc:derby:projects/project_29703/derby;create=true
```

dann

```
jdbc:derby://myServer:8455/projects/project_29703/derby
```

`jdbc.DRIVER`: Dieser Parameter muss bei Verwendung eines Tomcat-Web-Servers in `org.apache.derby.jdbc.ClientDriver` geändert werden, bei Verwendung eines Jetty-Web-Servers ist keine Anpassung erforderlich.

Bei Verwendung des FirstSpirit-Moduls Integration muss die Konfiguration aus dem JDBC-Treiber-Modul nach diesen Änderungen manuell aktualisiert werden, wenn in der Datenbank-Verbindung die Option "Benutzerspezifisch" aktiviert ist:



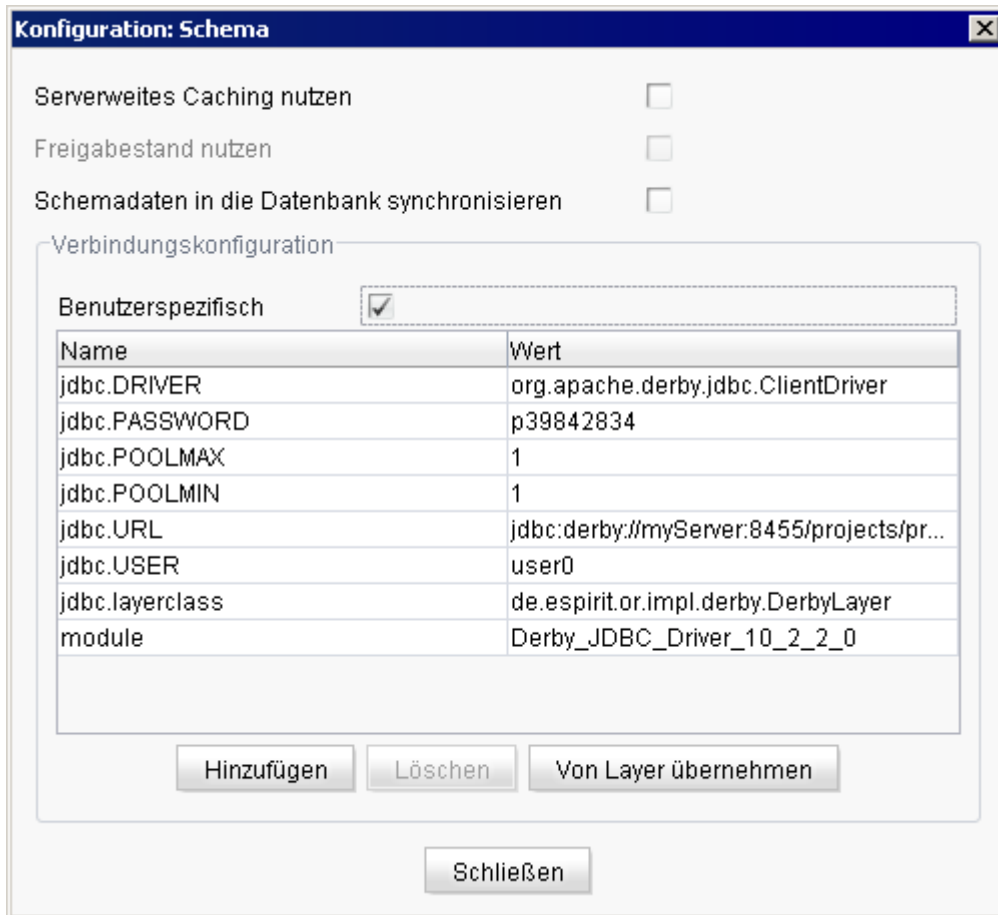


Abbildung 6-8: FirstSpirit Integration – Datenbank-Konfiguration

Ist die Option nicht aktiviert, werden die aktuell im JDBC-Treiber-Modul und in der Datenbank-Layer-Konfiguration gespeicherten Werte übernommen.

Wird neben dem externen Tomcat- auch der interne Jetty-Web-Server verwendet, so ist auch der Datenbank-Layer anzupassen (siehe Kapitel 6.1.2.2 Seite 127).

6.1.2.4.2 Individuelle Modul-Implementierungen

Die Anpassungen, die für das Beispiel "FirstSpirit Integration" (siehe Kapitel 6.1.2.4.1 Seite 130) vorgenommen werden müssen, müssen in jedem Modul, das individuell programmiert wurde und mit der Derby-Datenbank zusammenarbeitet, vorgenommen werden.



6.2 Funktionserweiterungen der Integrierten Vorschau

Die in FirstSpirit Version 4.2 im JavaClient eingeführte "Integrierte Vorschau" kann ab 4.2R4 nicht nur wie bisher für eine Vorschau der aktuell bearbeiteten Inhalte verwendet werden, sondern es können (bei korrekter Konfiguration (siehe Kapitel 4.2.2.3 Seite 41)) auch andere Dateiformate (z. B. Microsoft Word- oder Excel-Dateien) siehe Kapitel 4.2.2.1 Seite 38) und die FirstSpirit Online Hilfe (siehe Kapitel 4.2.2.4 Seite 43) nun dort dargestellt werden, und zwar mithilfe einer zum Format passenden Anwendung. Die jeweilige Anwendung muss dazu auf dem Arbeitsplatzrechner des jeweiligen Benutzers installiert sein. Zu weiteren Voraussetzungen und Einschränkungen siehe auch Kapitel 3.1 ab Seite 19.

Um die Integrierte Vorschau in der Medien-Verwaltung von der bisher bekannten Integrierten Vorschau für HTML-Inhalte abgrenzen zu können, wird die bisherige HTML-Vorschau in der rechten Spalte des JavaClients als "Integrierte Vorschau für Inhalte" bezeichnet, die neue Vorschau als "Integrierte Vorschau für Medien".

In den Projekteigenschaften können projektweite Anzeigeeoptionen voreingestellt werden, und zwar im Bereich "Client-Applikationen". Die hier vorgenommenen Einstellungen haben Einfluss auf das Menü "Ansicht" im JavaClient (siehe Kapitel 4.2.2.3 Seite 41). Damit die Änderungen übernommen werden, muss das betreffende Projekt neu gestartet werden.

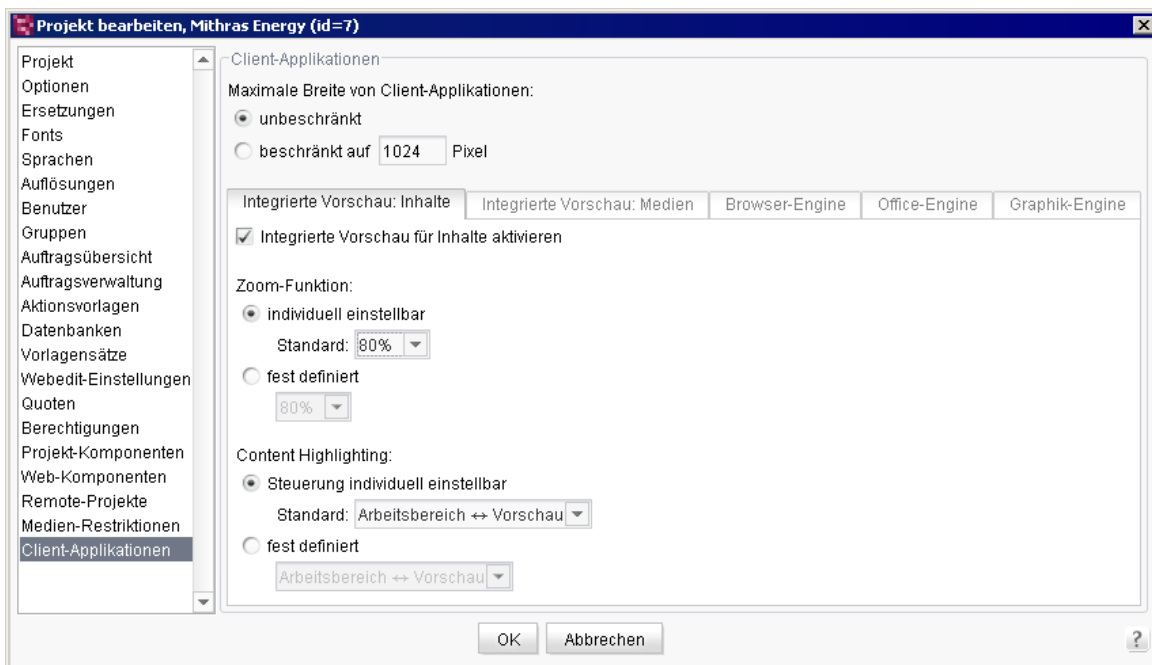


Abbildung 6-9: Projekteigenschaften – Client-Applikationen



Mit 4.2R4 ist der für die Integrierte Vorschau verwendete JxBrowser auf die Version 2.5 aktualisiert worden (mit Gecko Engine 1.9.2 / Firefox 3.6). In einigen Fällen kann es durch diese Aktualisierung zu Fehlern kommen (z. B. `java.lang.UnsatisfiedLinkError`), so dass die Integrierte Vorschau nicht angezeigt wird. Dieses Problem kann durch das Installieren von Microsoft Visual C++ 2008 SP1 Redistributable Package (x86) behoben werden.

Neu in Version 4.2R4 sind unter "Client-Applikationen" folgende Register:

6.2.1 Integrierte Vorschau: Medien

Hier wird festgelegt, ob die Integrierte Vorschau für Medien prinzipiell im jeweiligen Projekt verwendet werden können soll und welche Dateiformate mittels welcher Anwendung dargestellt werden sollen:

The screenshot shows the 'Integrierte Vorschau: Medien' tab in a settings window. The 'Integrierte Vorschau für Medien aktivieren' checkbox is checked. Below it are four text input fields for file extensions:

- Office-Engine verwenden für folgende Datei-Endungen (kommasepariert):
odt,ott,swx,doc,odg,otg,sxd,ods,ots,src,cls,xls,odf,sxm,odp,otp,sxi,ppt,odm,oth,odb
- Browser-Engine verwenden für folgende Datei-Endungen (kommasepariert):
pdf,html,htm,url,swf
- Integrierten Text-Editor verwenden für folgende Datei-Endungen (kommasepariert):
css,js,txt,xml,csv,json,as
- Microsoft Media Player (nur Windows) verwenden für folgende Datei-Endungen (kommasepariert):
avi,mpg,mpeg,wmv,asf,mp3,mp4,flv

A 'Restore Defaults' button is located at the bottom right of the dialog.

Abbildung 6-10: Projekteigenschaften – Voreinstellung Integrierte Vorschau/Medien

Integrierte Vorschau für Medien aktivieren: Diese Option hängt mit der Option "Office-Engine aktivieren" auf dem Register "Office-Engine" (siehe Kapitel 6.2.3 Seite 137) und "Grafik-Engine aktivieren" auf dem Register "Graphik-Engine" (siehe Kapitel 6.2.4 Seite 138) zusammen: wird diese Option deaktiviert, werden automatisch die Optionen "Office-Engine aktivieren" und "Grafik-Engine aktivieren" deaktiviert.

Ist die Option "Integrierte Vorschau für Medien aktivieren" deaktiviert, können auf



diesem Register keine weiteren Änderungen vorgenommen werden. Im Menü "Ansicht" / "Integrierte Vorschau" ist für dieses Projekt im JavaClient die Option "für Medien verwenden" ausgegraut und kann nicht ausgewählt werden. In neuen Projekten, die mit einer FirstSpirit-Server-Version ab 4.2.400 erstellt werden, ist die Option standardmäßig aktiviert, so dass mittels der folgenden Felder eine Zuordnung vorgenommen werden kann, welche Dateiformate von welchen Anwendungen in der Integrierten Vorschau dargestellt werden sollen. Für Projekte, die mit einer FirstSpirit-Server-Version unterhalb von 4.2.400 erstellt wurden, ist die Option standardmäßig deaktiviert.

Office-Engine verwenden für folgende Datei-Endungen (kommasepariert): In diesem Feld können Datei-Endungen komma-separiert angegeben werden, die über die Office-Engine (Microsoft Office, OpenOffice oder Google Docs) dargestellt werden sollen. Standardmäßig sind hier Dateiformate von Textverarbeitungs-, Tabellenkalkulations- und Präsentationsprogrammen vorgegeben.

Browser-Engine verwenden für folgende Datei-Endungen (kommasepariert): In diesem Feld können Datei-Endungen komma-separiert angegeben werden, die über die Browser-Engine (Microsoft Internet Explorer und Mozilla Firefox) dargestellt werden sollen. Standardmäßig sind hier Dateiformate vorgegeben, die von den Web-Browsern bzw. von Plugins dieser Browser dargestellt werden können.

Integrierten Text-Editor verwenden für folgende Datei-Endungen (kommasepariert): In diesem Feld können Datei-Endungen komma-separiert angegeben werden, die über den in FirstSpirit integrierten Text-Editor dargestellt werden sollen. Standardmäßig sind hier Dateiformate vorgegeben, die mit Text-Editoren erstellt und bearbeitet werden können.

Integrierte Bild-Ansicht verwenden für folgende Datei-Endungen (kommasepariert): In diesem Feld können die Datei-Endungen komma-separiert angegeben werden, die über die Grafik-Engine (einfache Bildbearbeitung, Java Image Editor, Picnik, Pixlr) dargestellt werden sollen. Standardmäßig sind hier Bild-Dateiformate vorgegeben.

Microsoft Windows Media Player (nur Windows) verwenden für folgende Datei-Endungen (kommasepariert): In diesem Feld können Datei-Endungen komma-separiert angegeben werden, die über den Windows Media Player abgespielt werden sollen. Standardmäßig sind hier Audio- und Video-Dateiformate vorgegeben. Der Windows Media Player kann nur in Verbindung mit Microsoft Windows verwendet werden, für andere Betriebssysteme kann dieses Feld ausgegraut sein.



Restore Defaults: Mit einem Klick auf diese Schaltfläche kann die Standard-Einstellung wiederhergestellt werden.

6.2.2 Browser-Engine

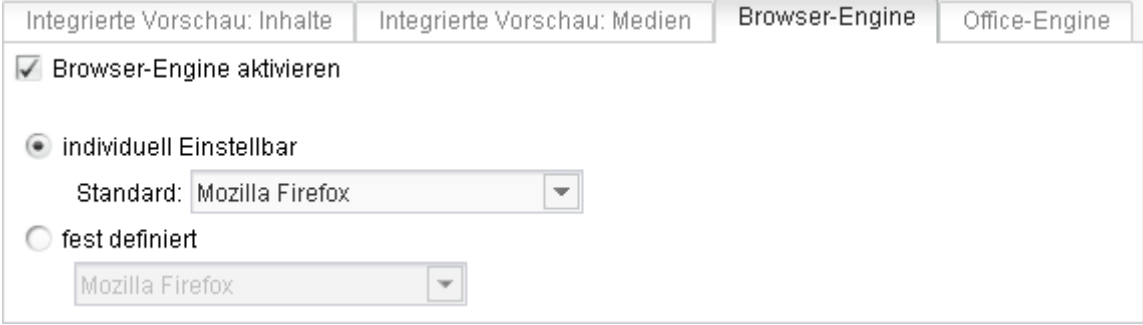


Abbildung 6-11: Projekteigenschaften – Voreinstellung Integrierte Vorschau/Browser

Browser-Engine aktivieren: Diese Option hängt mit der Option "Integrierte Vorschau für Inhalte" auf dem Register "Integrierte Vorschau: Inhalte" zusammen: es können nur beide Optionen gleichzeitig aktiviert oder deaktiviert sein. Ist die Option "Browser-Engine aktivieren" deaktiviert, können auf diesem Register keine Änderungen vorgenommen werden. Die Dateiformate, die auf dem Register "Integrierte Vorschau: Medien" im Feld "Browser-Engine" (siehe Kapitel 6.2.1 Seite 134) definiert sind, werden dann nicht über die in FirstSpirit integrierte Browser-Engine dargestellt, sondern in einer externen Anwendung. Standardmäßig ist die Option aktiviert, so dass mittels der folgenden Radio- und Comboboxen festgelegt werden kann, welche Browser-Engine für die im Feld "Browser-Engine" definierten Dateiformate verwendet werden soll:

individuell einstellbar: Ist diese Option aktiviert, können Benutzer im betreffenden Projekt die Browser-Engine selbst auswählen. Standardmäßig ist diese Option aktiviert. Über die Combobox "Standard:" kann vorgegeben werden, welche Browser-Engine im Projekt vorausgewählt sein soll. Standardmäßig ist dies Mozilla Firefox.

fest definiert: Ist diese Option aktiviert, kann für das Projekt eine Browser-Engine vorgegeben werden. Im JavaClient ist dann im Untermenü "Browser Engine" im Menü "Ansicht" in diesem Projekt keine Auswahl durch den Benutzer möglich.

Zu Systemvoraussetzungen und Einschränkungen hinsichtlich der Browser Engine siehe Kapitel 3.1.3 Seite 22.



6.2.3 Office-Engine



Integrierte Vorschau: Inhalte | Integrierte Vorschau: Medien | Browser-Engine | Office-Engine

Office-Engine aktivieren

individuell Einstellbar

Standard: Microsoft Office (nur Windows) ▼

fest definiert

Microsoft Office (nur Windows) ▼

Abbildung 6-12: Projekteigenschaften – Voreinstellung Integrierte Vorschau/Office



Für diese Funktion ist eine gültige Lizenz erforderlich: Der Parameter `license.OFFICE_INTEGRATION` in der Lizenzdatei `fs-license.conf` muss auf den Wert 1 gesetzt sein (siehe auch Kapitel 6.5 Seite 144). Ansonsten ist der Reiter "Office-Engine" ausgegraut, es können keine Einstellungen vorgenommen werden und Office-Dokumente können nicht in der Integrierten Vorschau des JavaClients dargestellt werden.

Office-Engine aktivieren: Diese Option hängt mit der Option "Integrierte Vorschau für Medien aktivieren" auf dem Register "Integrierte Vorschau: Medien" (siehe Kapitel 6.2.1 Seite 134) zusammen: wird die Option "Office-Engine aktivieren" aktiviert, wird auch automatisch die Option "Integrierte Vorschau für Medien aktivieren" aktiviert.

Ist die Option "Office-Engine aktivieren" deaktiviert, können auf diesem Register keine Änderungen vorgenommen werden. Die Dateiformate, die auf dem Register "Integrierte Vorschau: Medien" im Feld "Office-Engine" (siehe Kapitel 6.2.1 Seite 134) definiert sind, werden dann nicht in der Integrierten Vorschau für Medien dargestellt, sondern in einer externen Anwendung.

Mit gültiger Lizenz ist diese Option standardmäßig aktiviert, so dass mittels der folgenden Radio- und Comboboxen festgelegt werden kann, welche Anwendung für die im Feld "Office-Engine" definierten Dateiformate verwendet werden soll:

individuell einstellbar: Ist diese Option aktiviert, können Benutzer im betreffenden Projekt die Office-Engine selbst auswählen. Über die Combobox "Standard:" kann vorgegeben werden, welche Office-Engine im Projekt vorausgewählt sein soll. Standardmäßig ist dies Microsoft Office. Für Nicht-Windows-Betriebssysteme kann Microsoft Office nicht verwendet werden, daher sollte hier dann "OpenOffice" (aktuell



nur BETA-Test-Phase!) ausgewählt werden.

fest definiert: Ist diese Option aktiviert, kann für das Projekt eine Office-Engine vorgegeben werden. Im JavaClient ist dann im Untermenü "Office Engine" im Menü "Ansicht" in diesem Projekt keine Auswahl durch den Benutzer möglich. Bei Neu-Installationen ist diese Option standardmäßig aktiviert.


Hinweise zur Verwendung von Microsoft Office siehe Kapitel 3.1 Seite 19, zur Verwendung von OpenOffice und Google Docs siehe Kapitel 3.3 Seite 25.

6.2.4 Graphik-Engine



Abbildung 6-13: Projekteigenschaften – Voreinstellung Integrierte Vorschau/Grafik

Grafik-Engine aktivieren: Diese Option hängt mit der Option "Integrierte Vorschau für Medien aktivieren" auf dem Register "Integrierte Vorschau: Medien" (siehe Kapitel 6.2.1 Seite 134) zusammen: wird die Option "Grafik-Engine aktivieren" aktiviert, wird auch automatisch die Option "Integrierte Vorschau für Medien aktivieren" aktiviert.

Ist die Option "Grafik-Engine aktivieren" deaktiviert, können auf diesem Register keine Änderungen vorgenommen werden. Im JavaClient ist der Eintrag "Grafik Engine" im Menü "Ansicht" ausgegraut und deaktiviert. Bilder können weiterhin mit den bisher bekannten Funktionen (Icon ) bearbeitet werden.

Ist die Option aktiviert, kann mittels der folgenden Radio- und Comboboxen festgelegt werden, welche Anwendungen verwendet werden sollen:

individuell einstellbar: Ist diese Option aktiviert, können Benutzer im betreffenden Projekt die Grafik-Engine selbst auswählen. Über die Combobox "Standard:" kann vorgegeben werden, welche Grafik-Engine im Projekt vorausgewählt sein soll.

fest definiert: Ist diese Option aktiviert, kann für das Projekt eine Grafik-Engine vorgegeben werden. Im JavaClient ist dann im Untermenü "Grafik Engine" im Menü



"Ansicht" in diesem Projekt keine Auswahl durch den Benutzer möglich.

Weitere Informationen zu den Bildbearbeitungsfunktionen siehe Kapitel 4.2.3 Seite 44. Zu Systemvoraussetzungen und Einschränkungen hinsichtlich der Grafik Engine siehe Kapitel 3.2 Seite 24 (Java Image Editor) und Kapitel 3.3 Seite 25 (Picnik und Pixlr).

6.3 Globale Web-Applikationen

Ab Version 4.2R4 können über die Server-Eigenschaften eigene Web-Applikationen definiert und konfiguriert werden, die dann in allen Projekten des Servers zur Verfügung stehen ("global"). Damit können z. B. eigene Web-Anwendungen für das FirstSpirit AppCenter (siehe Kapitel 3 Seite 18) installiert werden.

Dazu werden die Server-Eigenschaften, Bereich "Web-Applikationen", geöffnet:

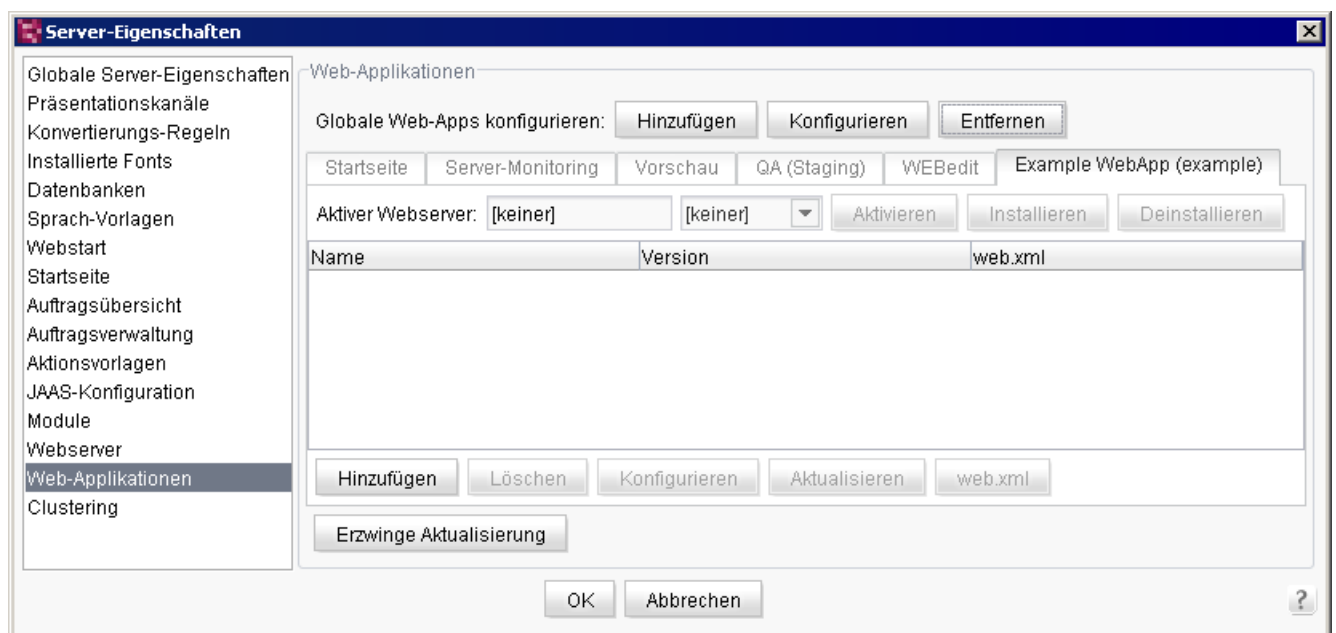


Abbildung 6-14: Server-Eigenschaften – Web-Applikationen

6.3.1 Installieren von globalen Web-Applikationen

Über den Button "Hinzufügen" kann eine neue Web-Applikation auf dem Server installiert werden:



Abbildung 6-15: Neue Web-Applikation hinzufügen

Id: Hier muss ein eindeutiger Bezeichner für die Web-Applikation eingegeben werden. Auf diesen ist ein Zugriff über die API möglich (siehe Kapitel 6.3.4 Seite 141). Außerdem wird auf dem Server für die Web-Applikation ein Unterverzeichnis mit diesem Bezeichner erstellt. Es dürfen nur Groß-, Kleinbuchstaben, Ziffern und Unterstriche verwendet werden. Dieser Bezeichner kann nach dem Speichern nicht mehr geändert werden.

Name: Hier muss ein Name angegeben werden, der als Anzeigename verwendet wird. Dieser kann später bei Bedarf geändert werden.

Web Context: Hier muss ein sogenannter Kontextname eingegeben werden, der einen Teil der URL zur Web-Applikation bildet. Er darf nicht mit den Namen der bereits standardmäßig vorhandenen FirstSpirit Web-Applikationen (also *fs4root*, *fs4preview*, *fs4staging*, *fs4webedit*, *fs4webmon*) übereinstimmen.

6.3.2 Konfiguration

Zu jeder Web-Applikation

- kann ein Webserver eingestellt werden:

- können auf dem Server zur Verfügung stehende Web-Komponenten hinzugefügt, entfernt, konfiguriert und aktualisiert werden:

- kann die Datei `web.xml` manuell bearbeitet werden:



Die Web-Komponenten können zu einer Web-Applikation zusammengefasst und auf den Web-Server installiert oder als WAR-Datei herunterzuladen.

Die Funktion ist dabei analog zu der von Web-Komponenten in den Projekteigenschaften. Siehe dazu auch *FirstSpirit Handbuch für Administratoren*, Kapitel "Web-Komponenten".

6.3.3 Aktualisierung

Zum Aktualisieren von globalen Web-Applikationen steht der Button "Verwendungen aktualisieren" in den Server-Eigenschaften, Bereich "Module", zur Verfügung. Beim Klick auf den Button öffnet sich ein Dialogfenster zur Aktualisierung der Web-Applikationen für alle Projekte, die diese Anwendung bisher verwenden. Die Aktualisierung muss also nicht mehr einzeln für jedes Projekt über die Projekteigenschaften ausgeführt werden, sondern kann zentral über die Server-Eigenschaften gesteuert werden.

6.3.4 Zugriff über die API

Über die FirstSpirit-Developer-API kann anhand der Id der Web-Applikation (siehe Abbildung 6-15) die URL zu einer globalen Web-Applikation über das Interface `de.espirit.firstspirit.agency.LegacyModuleAgent` ermittelt werden.



6.4 Performance und Sicherheit

In Version 4.2R4 wurden Veränderungen vorgenommen, mit denen die Effizienz und Geschwindigkeit von FirstSpirit gesteigert werden konnte, speziell im Falle von Generierungen: Für große Projekte konnte die Generierungsgeschwindigkeit um etwa ein Drittel gesteigert werden. Ebenso wurden performantere Datenformate für Referenzen (Referenzgraph) und Arbeitsabläufe eingeführt (siehe dazu auch Kapitel 2.2.1 Seite 14). Referenzen werden zudem beim Speichern komprimiert, wenn sie einen gewissen Schwellwert überschreiten.

Um die vorhandene Hardware besser auszunutzen, wurden für eine bessere Performance auch die Standardwerte für die Größe des Thread Pools und der Thread Queues in der Konfigurationsdatei `fs-server.conf` angepasst. So wird der FirstSpirit-Server nun nach Installation oder nach einem Wechsel der Hardware automatisch auf die Anzahl der vorhandenen Rechenkerne skaliert. Die Anzahl der verfügbaren Prozessoren wird dabei über

```
java.lang.Runtime.getRuntime().availableProcessors()
```

bestimmt.

Die Standardwerte im Bereich **Thread Pool** der Datei `fs-server.conf` sehen nun folgendermaßen aus:

```
#####  
# Thread Pool  
#####  
# minimum number of concurrent threads, if left empty the value is  
# set to #cores (= number of cores as delivered by  
#"java.lang.Runtime.availableProcessors()")  
ThreadPool.minSize=  
# maximum number of concurrent threads, if left empty the value is  
# set to (#cores * 8)  
ThreadPool.maxSize=
```

Werden für die Parameter

- `ThreadPool.minSize` (Minimalgröße des limitierten Thread Pools) und
- `ThreadPool.maxSize` (Maximalgröße des limitierten Thread Pools und damit die größtmögliche Anzahl parallel auszuführender Tasks)

keine expliziten Werte angegeben, werden automatisch Werte verwendet, die abhängig von der Anzahl der verfügbaren Prozessoren sind.



Der Bereich **Thread Queues** der Datei `fs-server.conf` sieht nun folgendermaßen aus:

```
#####
# Thread Queues:
# - LOW: Queue for resource-intensive tasks.
# - DEFAULT: Default queue for default tasks.
# - BOUNDED: Bounded queue with rejection strategy.
#   (queueCapacity: -1 = unbounded, 0 = no queueing allowed)
# Attributes:
# - maxRunning    maximum numbers of running tasks.
# - queueCapacity queue capacity (-1 = unbounded, 0 = no queueing
# allowed).
# - rejection     rejection strategy for incoming tasks if queue
# capacity is exceeded (REJECT, BLOCK, EXECUTE).
#                 REJECT reject task if queue capacity is exceeded
# (only allowed in BOUNDED queue!).
#                 BLOCK  wait until queue has free capacities.
#                 EXECUTE execute queue's oldest task inline.
#####
ThreadQueue.LOW.maxRunning=2
ThreadQueue.LOW.queueCapacity=128
ThreadQueue.LOW.rejection=BLOCK

# if left empty the value is set to (#cores * 6)
ThreadQueue.DEFAULT.maxRunning=
# if left empty the value is set to (#cores * 20)
ThreadQueue.DEFAULT.queueCapacity=
ThreadQueue.DEFAULT.rejection=BLOCK

# if left empty the value is set to (#cores * 6)
ThreadQueue.BOUNDED.maxRunning=
# if left empty the value is set to (#cores * 16)
ThreadQueue.BOUNDED.queueCapacity=
ThreadQueue.BOUNDED.rejection=REJECT
```

Werden für die Parameter

- `ThreadQueue.DEFAULT.maxRunning` (maximale Anzahl der parallel ausführbaren Tasks einer mit `DEFAULT` klassifizierten Queue),
- `ThreadQueue.DEFAULT.queueCapacity` (Aufnahmekapazität einer mit `DEFAULT` klassifizierten Queue),
- `ThreadQueue.BOUNDED.maxRunning` (maximale Anzahl der parallel ausführbaren Tasks einer mit `BOUNDED` klassifizierten Queue) und
- `ThreadQueue.BOUNDED.queueCapacity` (Aufnahmekapazität einer mit `BOUNDED` klassifizierten Queue)

keine expliziten Werte angegeben, werden automatisch Werte verwendet, die abhängig von der Anzahl der verfügbaren Prozessoren sind





Die Werte der Parameter `ThreadQueue.LOW.maxRunning` und `ThreadQueue.LOW.queueCapacity` sind nicht abhängig von der Anzahl der Rechenkerne und können nicht mit "leeren" Werten überschrieben werden.



Der Wert für `ThreadQueue.<name>.maxRunning` muss kleiner sein als der Wert für `ThreadPool.maxSize` (Bereich **Thread Pool**, siehe oben).



Weitere Informationen zu diesen Parametern siehe "FirstSpirit Handbuch für Administratoren", Kapitel "Konfiguration des FirstSpirit-Servers (fs-server.conf)", Unterkapitel "Bereich Thread Pool" und "Bereich Thread Queues".

Ebenso wurde die FirstSpirit-Sicherheit weiter erhöht.



Damit diese Änderung zum Tragen kommt, muss `fs4root` aktualisiert werden. Wird die Aktualisierung nicht vorgenommen, können Client und Server keine Verbindung per HTTP aufnehmen.

6.5 Neuerungen im Lizenz-Management

In Version 4.2R4 sind zwei neue Lizenzparameter eingeführt worden:

6.5.1 license.OFFICE_INTEGRATION

Hierbei handelt es sich um den Lizenzparameter für die Office-Integration. Er muss den Wert `1` haben, damit Microsoft Office-, OpenOffice- oder Google Docs-Text-Dokumente im FirstSpirit AppCenter verwendet werden können (`license.OFFICE_INTEGRATION=1`). Ohne gültige Lizenz ist der Reiter zur Office-Engine-Konfiguration in den Projekteigenschaften ausgegraut (siehe Kapitel 6.2.3 Seite 137), im JavaClient ist das Menü "Ansicht" \ "Office Engine" deaktiviert.



6.5.2 license.APPTAB_SLOTS

Hierbei handelt es sich um den Lizenzparameter für das FirstSpirit AppCenter (siehe dazu auch Kapitel 3 Seite 18). Er gibt an, wie viele verschiedene AppCenter-Anwendungen (also eigene Anwendungen, die auf die Applikations-API zugreifen) verwendet werden können. Mit `license.APPTAB_SLOTS=5` können z. B. fünf verschiedene Anwendungen verwendet bzw. URLs aufgerufen werden. Welche Anwendungen das sind, ist dabei unerheblich. Denn im Unterschied zur Lizenzierung einer FirstSpirit-(Modul-)Erweiterung wird hier nicht die Funktionalität lizenziert, sondern die Anzahl der integrierten Anwendungen. Zu jeder über diesen Parameter lizenzierten Anwendung können im AppCenter beliebig viele Register geöffnet werden. Im Falle von Web-Anwendungen wird pro integriertem Web-Dienst eine separate Lizenz benötigt. Um nachvollziehen zu können, welche Web-Anwendungen integriert wurden, werden in der AppCenter-Lizenzübersicht (siehe Abbildung 6-16) die von einer AppCenter-Anwendung verwendeten URLs angezeigt. Es ist dabei zu beachten, dass zu einer Anwendung nur URLs gehören dürfen, die auch "logisch" zu der integrierten Web-Anwendung gehören und nicht zu einem völlig anderen Web-Dienst.

Der erste Client, in dem eine AppCenter-Anwendung geöffnet wird, belegt eine Lizenz für diese Anwendung (es erfolgt eine "Registrierung" der Anwendung) und erhöht den Zähler des Lizenzparameters um 1. Die Registrierung bleibt auch nach dem Beenden des Clients weiterhin bestehen. Ist der Wert des `license.APPTAB_SLOTS`-Parameters erreicht, kann zu Test- und Demozwecken noch eine weitere Anwendung im AppCenter gestartet werden (es wird eine entsprechende Warnung im JavaClient angezeigt). Darüber hinaus können keine weiteren Anwendungen gestartet werden, die bereits registrierten Anwendungen können jedoch in weiteren Registern bzw. in weiteren Clients geöffnet werden.

Einige Anwendungen, die zwar im AppCenter dargestellt, aber standardmäßig mit dem FirstSpirit-Kernprodukt ausgeliefert werden, fallen nicht unter den Lizenzierungsparameter `license.APPTAB_SLOTS` und werden nicht als AppCenter-Anwendung gezählt, zurzeit

- integrierte Browser
 - Vorschau von Medien (siehe Kapitel 4.2.2.1 Seite 38)
 - FirstSpirit Online Hilfe (siehe Kapitel 4.2.2.4 Seite 43)
 - erweiterte Bildbearbeitung (siehe Kapitel 4.2.3 Seite 44)
 - Office-Integration (siehe Kapitel 6.2.3 Seite 137)
- WICHTIG: Die Office-Integration erfordert einen eigenen Lizenzparameter, siehe dazu Kapitel 6.5.1 Seite 144.



Eine einfache Möglichkeit, eigene AppCenter-Anwendungen zu integrieren, bietet die neue Eingabekomponente FS_BUTTON (siehe Kapitel 5.4 Seite 106). Bei Verwendung dieser Eingabekomponente wird jedes Skript und / oder jedes Modul, das von FS_BUTTON referenziert wird und ein eigenes Register im AppCenter öffnet, als Anwendung, die über den Parameter `license.APPTAB_SLOTS` lizenzpflichtig ist, gezählt.

Die Integrierte Vorschau ist keine lizenzpflichtige AppCenter-Anwendung sondern eine FirstSpirit-Standardfunktion. Erst wenn in die FirstSpirit-Vorschau über AppCenter-Technologie eine eigene Erweiterung integriert werden soll, wird hierfür eine zusätzliche AppCenter-Lizenz benötigt. Beispiel: FS_BUTTON in der Vorschau.

Wie viele und welche Anwendungen, die über den Parameter `license.APPTAB_SLOTS` lizenzpflichtig sind, aktuell registriert sind, kann im Server-Monitoring im Untermenü "AppCenter Lizenzen" unterhalb des Menüs "FirstSpirit" / "Steuerung" geprüft werden:

The screenshot shows the FirstSpirit™ web interface. At the top, there is a header with the logo, a language dropdown set to 'Sprache', and user information: 'Benutzer Admin' and 'Monitoring beenden'. On the right, system statistics are displayed: 'FirstSpirit 5.0_DEV.2.44524', 'Projekte 74', 'Aktionen 0', 'Sitzungen 2', and 'Vorschau 0'. Below the header is a navigation bar with tabs: 'Wartungs-Modus', 'Web-Applikationen', 'Aktualisierung', 'Dienste', and 'Server Neustart'. The 'AppCenter Lizenzen' tab is selected. The main content area is titled 'AppCenter Lizenzen' and contains a section 'Verwendete AppCenter-Lizenzen' with a list of three entries:

Icon	URL
[+]	Script-image_search_fotolia http://us.fotolia.com/search?k=modern+building
[-]	Script-image_search_pixelio http://www.pixelio.de/search.php?search[phrase]=modern+building&search[allow_edit]=%3C+2&search[license]=%3C2
[-]	Script-wikipedia_search http://de.wikipedia.org/wiki/Spezial:Suche/FirstSpirit

Below the list is a button labeled 'Verwendungen zurücksetzen'. The left sidebar contains a navigation menu with items like 'Übersicht', 'Status', 'Aktivitäten', 'Sitzungen', 'Projekte', 'Logdateien', 'kompletter Server nach Projekt nach Veröffentlichung', 'Zeitplanung', 'Benutzer', 'Suche', 'FirstSpirit', 'Konfiguration', and 'Steuerung'.

Abbildung 6-16: Server-Monitoring – AppCenter-Lizenzen

Über den Button "Verwendungen zurücksetzen" kann bei Bedarf die Zahl der registrierten Anwendungen auf 0 zurückgesetzt werden. Registrierte Anwendungen, die aktuell in Clients geöffnet sind, können so lange weiter verwendet werden, bis die Anwendung bzw. das zugehörige Register geschlossen wird. Der Server braucht



nicht neu gestartet werden.

6.6 NTLM-Authentifizierung und Windows 7

Bei den Betriebssystemen Windows Vista, Windows 7 und Windows Server 2008 R2 wird standardmäßig für die Authentifizierung das NTLMv2-Verfahren verwendet.

Die NTLM-Authentifizierung wird vom FirstSpirit-Server genutzt, wenn beim Anmeldevorgang das NTLM-Loginmodul verwendet wird (vgl. *FirstSpirit Dokumentation für Administratoren*, Kapitel "Anmeldung an Windows-Domäne"). Das NTLM-Loginmodul ist **nicht** kompatibel zu NTLMv2. Bei der Verwendung der o. g. Betriebssystem-Versionen und des NTLM-Loginmodules muss die Einstellung der LAN Manager-Authentifizierungsebene geändert und NTLM(v1) zugelassen werden.



Das NTLMv2-Verfahren ist der Nachfolger von NTLM(v1) und wurde aufgrund von Sicherheitsproblemen entwickelt. Der Einsatz von NTLM(v1) sollte daher sorgfältig überlegt werden.

Nach folgender Anleitung kann das Betriebssystem auf das vorherige Verhalten umgestellt werden:

1. <Windows-Taste> + <R> drücken
2. `secpol.msc` eingeben und <Enter> drücken
3. Wechsel zu "Lokale Richtlinien" / "Sicherheitsoptionen":

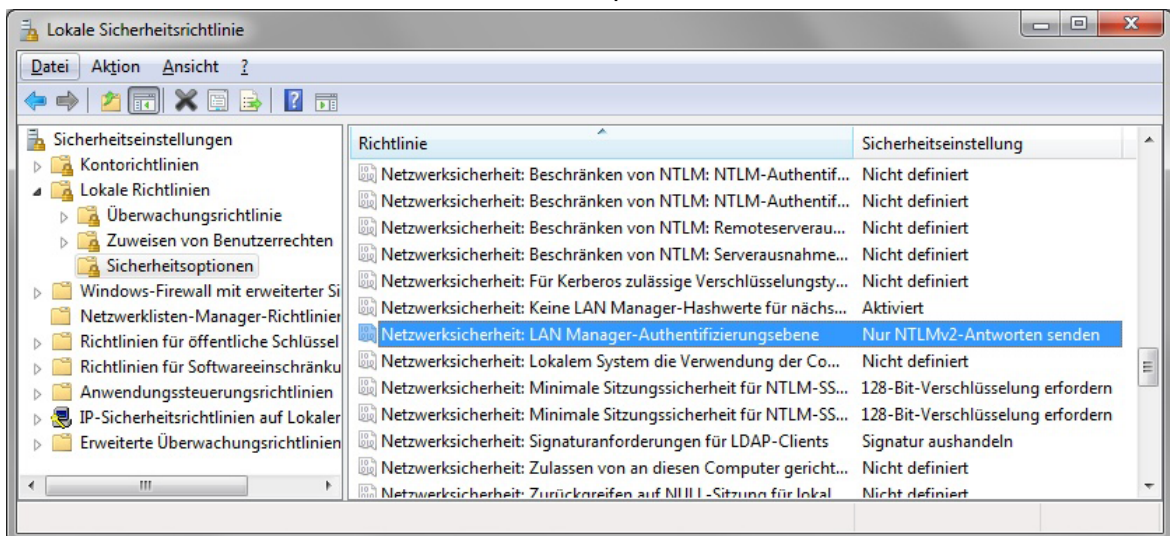


Abbildung 6-17: Netzwerksicherheit: LAN-Manager-Authentifizierungsebene



4. Beim Doppelklick auf den Eintrag "Netzwerksicherheit: LAN Manager-Authentifizierungsebene" öffnet sich ein Fenster.
5. Als Wert für die LAN-Authentifizierung muss in diesem Fenster NTLM zugelassen werden (im folgenden Screenshot wird die Standardeinstellung in Windows XP angezeigt):

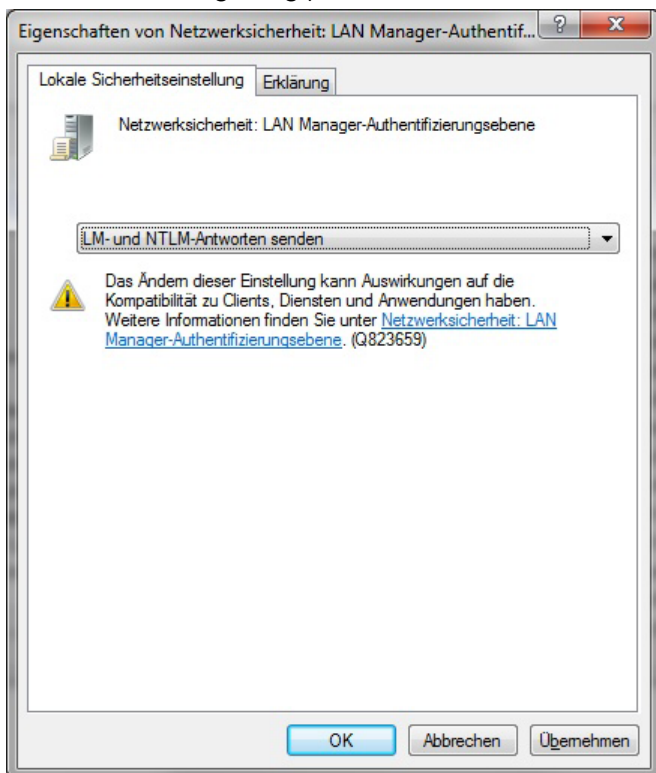


Abbildung 6-18: Standardeinstellung in Windows XP

6. Die Auswahl muss mit dem "OK"-Button bestätigt werden.



Ab der FirstSpirit Version 4.2 steht neben dem NTLM-Loginmodul auch das Kerberos-Loginmodul zur Verfügung. Für die Verwendung von Kerberos muss – im Gegensatz zu NTLM – keine Änderung der Einstellungen im Betriebssystem vorgenommen werden und ist die zu bevorzugende Variante.

6.7 WebEdit-Themes

Ab 4.2R4 wird nur noch das "xp"-Theme unterstützt. Bei diesem handelt es sich jetzt um das Default-Theme. Darüber hinaus werden keine Themes mit kundenspezifischen Einstellungen mehr unterstützt. Die Combobox "WebEdit Theme" in den Projekteigenschaften (Anwendung zur Server- und Projektkonfiguration / "Projekt" / Bereich "WebEdit-Einstellungen") ist ab 4.2R4 nicht



mehr vorhanden.

6.8 Überwachung des Festplattenplatzes

Mit Version 4.2R4 wurden einige Maßnahmen implementiert, um den bereits verbrauchten bzw. noch freien Speicherplatz der Festplatte besser überwachen und so eventuelle Datenverluste umgehen zu können.

Dazu können in der Konfigurationsdatei `fs-server.conf` oder in den Server-Eigenschaften (Anwendung zur "Server- und Projektkonfiguration", Menü "Server" / "Eigenschaften) unter "Globale Server-Eigenschaften" Verzeichnisse angegeben werden, die überwacht werden sollen, sowie Schwellenwerte für eine Warnung des Serveradministrators per E-Mail und das Herunterfahren des FirstSpirit-Servers. Der freie Speicherplatz in den angegebenen Verzeichnissen wird in einem Intervall von fünf Minuten ermittelt.



Aus technischen Gründen ist die Überwachung des Festplattenplatzes nur unter JDK 1.6 möglich.



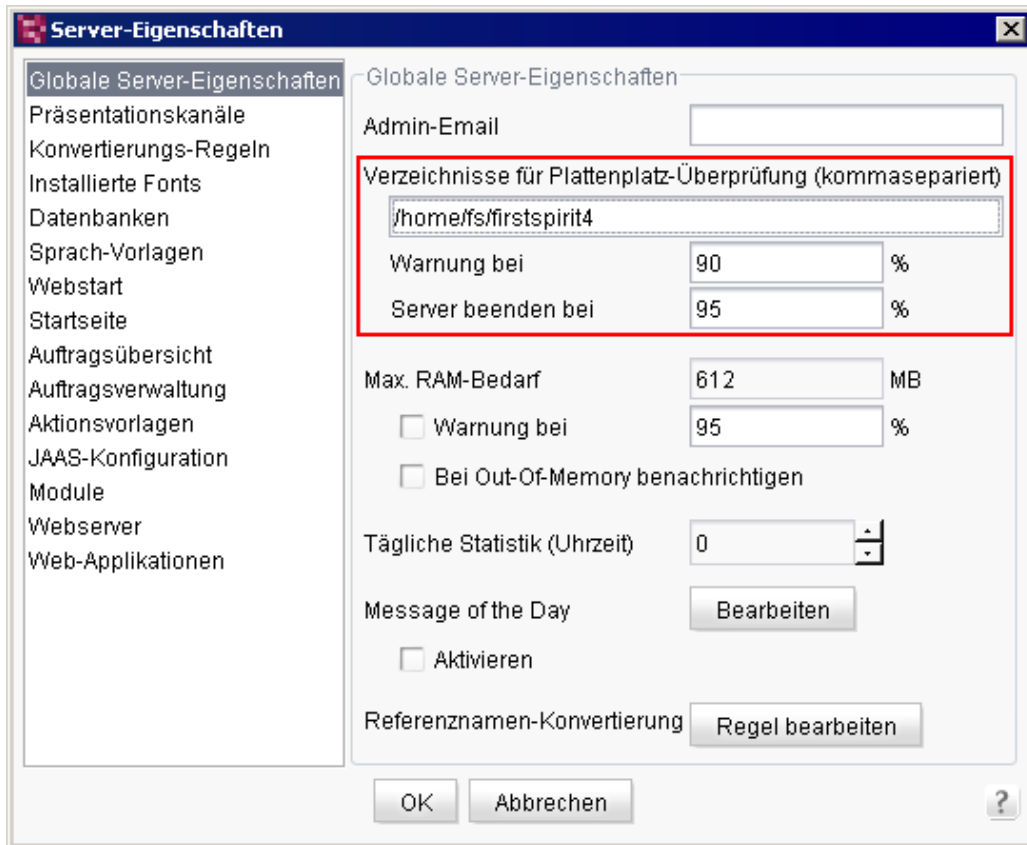


Abbildung 6-19: Server-Eigenschaften – Globale Server-Einstellungen

Verzeichnisse für Plattenplatz-Überprüfung (kommasepariert): In dieses Feld werden die Verzeichnisse eingetragen, die hinsichtlich des noch zur Verfügung stehenden Festplattenplatzes überwacht werden sollen. Standardmäßig ist hier das Basis-Verzeichnis des FirstSpirit-Servers angegeben. Wenn FirstSpirit über mehrere Mountpoints verteilt ist, müssen alle Mountpoints der Dateisysteme angegeben werden, die von FirstSpirit verwendet werden. Mehrere Verzeichnisse müssen durch Kommata voneinander getrennt angegeben werden.

Warnung bei: In diesem Feld wird die Prozentzahl angegeben, ab der eine Warnungs-E-Mail an den Serveradministrator versendet werden soll. Betreff und Inhalt der E-Mail: "FIRSTspirit server 'MYSERVER' disk space warning: /home/fs/firstspirit4, 32,01 GB free, 42,52 GB used".

Die Prozentzahl ergibt sich dabei aus dem Verhältnis des im angegebenen Verzeichnis zur Verfügung stehenden und des bereits belegten Speicherplatzes. Standardmäßig ist der Wert auf 90% gesetzt, d. h. der Serveradministrator erhält eine Warnungs-E-Mail, wenn 90% des zur Verfügung stehenden Speicherplatzes belegt sind (siehe auch unten, Parameter `hdd.limit`). Sind mehrere Verzeichnisse angegeben, wird eine E-Mail versendet, wenn in **einem** der Verzeichnisse die Prozentzahl überschritten ist. Der hier gewählte Wert sollte niedriger sein, als der



Wert für "Server beenden" (siehe unten). Es wird maximal alle 12 Stunden eine Warnungs-E-Mail versendet.

Server beenden bei: In diesem Feld wird die Prozentzahl angegeben, bei der der FirstSpirit-Server heruntergefahren und eine E-Mail an den Serveradministrator versendet werden soll. Betreff und Inhalt der E-Mail: "FIRSTspirit server 'MYSERVER' disk space shutdown limit reached: /home/fs/firstspirit4, 463.2 MB free, 5.54 GB used".

Die Prozentzahl ergibt sich dabei aus dem Verhältnis des im angegebenen Verzeichnis zur Verfügung stehenden und des bereits belegten Speicherplatzes. Standardmäßig ist der Wert auf 95% gesetzt, d. h. der Server wird heruntergefahren, wenn 95% des zur Verfügung stehenden Speicherplatzes belegt sind (siehe auch unten, Parameter `hdd.shutdown`). Sind mehrere Verzeichnisse angegeben, wird der Server heruntergefahren, wenn in **einem** der Verzeichnisse die Prozentzahl überschritten ist. Der hier gewählte Wert sollte höher sein, als der Wert für "Warnung bei" (siehe oben).

Die Angabe der zu überwachenden Verzeichnisse und die Schwellenwerte kann auch direkt in der Konfigurationsdatei `fs-server.conf` sowie über das Server-Monitoring ("FirstSpirit" / "Konfiguration" / "Server") vorgenommen werden:

`hdd.directories`: Über diesen Parameter werden die zu überwachenden Verzeichnisse angegeben. Standardwert ist das FirstSpirit-Basis-Verzeichnis, also `hdd.directories=${cmsroot}`. Dieser wird verwendet, wenn der Wert nicht manuell angegeben wird

`hdd.limit`: Über diesen Parameter wird die Prozentzahl angegeben, ab der eine Warnungs-E-Mail an den Serveradministrator versendet wird. Es können Werte von `1` bis `99` angegeben werden. Sollen keine Warnungs-E-Mails versendet werden, kann der Wert auf `-1` gesetzt werden.

Standardwert ist `90%`, also `hdd.limit=90`. Dieser wird verwendet, wenn der Wert nicht manuell angegeben wird oder wenn der angegebene Wert nicht zwischen `1` und `99` liegt bzw. `-1` ist.

`hdd.shutdown`: Über diesen Parameter wird die Prozentzahl angegeben, ab der eine Warnungs-E-Mail an den Serveradministrator versendet und der Server heruntergefahren wird. Es können Werte von `1` bis `99` angegeben werden. Soll der Server nicht heruntergefahren werden, kann der Wert auf `-1` gesetzt werden. Ist der gewählte Wert gleich oder kleiner als `hdd.limit`, wird `hdd.shutdown` vom System um 5% höher gesetzt als `hdd.limit`.



Standardwert ist 95%, also `hdd.shutdown=95`. Dieser wird verwendet, wenn der Wert nicht manuell angegeben wird oder wenn der angegebene Wert nicht zwischen 1 und 99 liegt bzw. -1 ist.

Bei fehlender oder ungültiger Konfiguration werden entsprechende Warnungen in der Log-Datei `fs-server.log` festgehalten, z. B.

- keine oder ungültige Verzeichnisse:

```
WARN 22.03.2011 14:53:47.804
(de.espirit.firstspirit.server.ServerManagerImpl): Empty fs-
server.conf property: hdd.directories
```

bzw.

```
WARN 22.03.2011 14:53:47.804
(de.espirit.firstspirit.server.ServerManagerImpl): Illegal fs-
server.conf property: hdd.directories, illegal directory: C:\test
```

- keine oder ungültige Werte für die Schwellenwerte:

```
WARN 22.03.2011 14:53:47.805
(de.espirit.firstspirit.server.ServerManagerImpl): Illegal fs-
server.conf property value: hdd.limit
```

```
WARN 22.03.2011 14:53:47.805
(de.espirit.firstspirit.server.ServerManagerImpl): Illegal fs-
server.conf property value: hdd.shutdown=120
```

- keine E-Mail-Adresse:

```
WARN 22.03.2011 15:11:30.739
(de.espirit.firstspirit.server.ServerManagerImpl): Empty fs-
server.conf property: mail.default-recipient
```



6.9 Referenznamen: Automatische Umwandlung von Sonderzeichen

Referenznamen dürfen in keiner Verwaltung Sonderzeichen enthalten, im FirstSpirit-JavaClient können darüber hinaus keine Großbuchstaben für Referenznamen verwendet werden. Im JavaClient werden diese ungültigen Zeichen noch während der Eingabe entfernt, Großbuchstaben werden in Kleinbuchstaben umgewandelt. Im FirstSpirit WebClient werden Sonderzeichen automatisch in Unterstriche umgewandelt (_) und auch Großbuchstaben sind erlaubt.

Mit FirstSpirit Version 4.2R4 kann serverweit ein Regelsatz definiert werden, nach dem ungültige Zeichen beim Anlegen neuer FirstSpirit-Objekte oder beim Ändern des Referenznamens (über das Kontextmenü "Extras" / "Referenznamen ändern") umgeformt werden sollen. Auf diese Weise kann z. B. jedes "Ä" bei der Eingabe automatisch in ein "ae" oder Sonderzeichen wie Punkte oder Kommas in Unterstriche transformiert werden. Dies ist z. B. sinnvoll, wenn Referenznamen bei der Generierung eines Projekts für die Erzeugung von URLs genutzt werden (siehe auch Kapitel 4.1.3 Seite 31).

Dazu wird in den Server-Eigenschaften (Anwendung zur Server- und Projektkonfiguration, Menü "Server", Eintrag "Eigenschaften") unter "Globale Server-Einstellungen" die Schaltfläche "Regel bearbeiten" angeklickt:

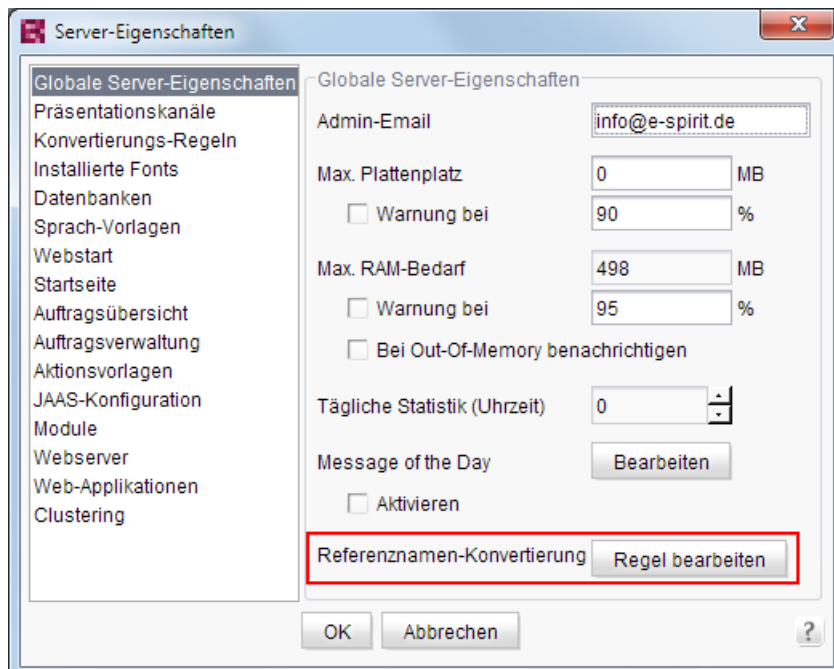


Abbildung 6-20: Server-Eigenschaften – Globale Server-Einstellungen



Es öffnet sich folgendes Fenster:

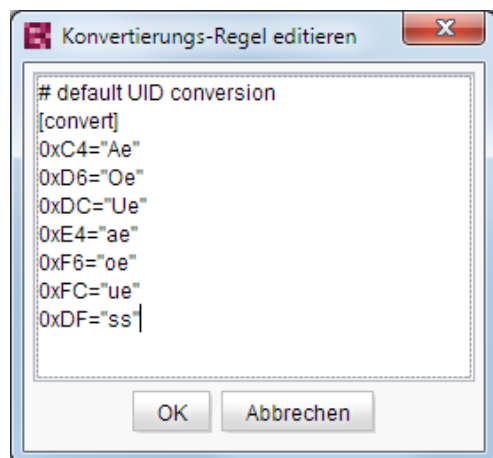


Abbildung 6-21: Referenznamen-Konvertierung

Hier sind standardmäßig bereits einige Regeln definiert, und zwar für die Umlaute "ä", "ö" und "ü" in Klein- und Großschrift sowie für "ß". Jede Regel muss in einer Zeile stehen und besteht aus zwei Werten, die durch ein Gleichheitszeichen getrennt werden:

- **links** das Sonderzeichen, das umgeformt werden soll, in ASCII-Codierung (hexadezimal)
- **rechts** das oder die gültige/n Zeichen, in das das Sonderzeichen bei der Verwendung in Referenznamen umgeformt werden soll, in doppelten Hochkommata.

Mit dem Standard-Regelsatz werden Umlaute bei der Verwendung in Referenznamen in 2 Zeichen umgeformt (Vokal in Kleinschrift + "e"), "ß" wird in "ss" umgeformt.



*Werden die Zeichen in diesem Dialog nicht in der richtigen Codierung bzw. Formatierung angegeben, wird beim Speichern des Regelsatzes eine Fehlermeldung in der folgenden Form angezeigt:
"Das Format der Konvertierungs-Regel ist fehlerhaft: Error parsing line 11:...".*

Die mitgelieferten Regeln können verändert oder gelöscht, und neue Regeln können hinzugefügt werden. Sie werden bei einem FirstSpirit-Update nicht zurückgesetzt oder ergänzt.



Mit einem Klick auf "OK" werden die Regeln gespeichert. Wird der darunterliegende Dialog mit den Server-Eigenschaften per "OK" geschlossen, werden die Regeln sofort angewendet, Server oder Projekt müssen nicht erst neu gestartet werden.

6.9.1 Konfiguration

Es können jeweils nur Regeln für einzelne Zeichen, nicht für Zeichenketten, definiert werden und nur sprachunabhängig, d. h. jede Regel gilt für alle Sprachen. Wird für ein Sonderzeichen (ASCII-Codierung) mehr als eine Definition angegeben, wird immer die unterste Definition angewendet. Sonderzeichen, für die keine Regel hinterlegt ist, werden bei der Eingabe eines Referenznamens weiterhin direkt entfernt.

Werden auf der rechten Seite Großbuchstaben definiert, werden diese im JavaClient automatisch in Kleinbuchstaben umgeformt.

Enthält eine Definition auf der rechten Seite (Sonder)zeichen, für die wiederum eine Konvertierungs-Regel definiert ist, werden diese (Sonder)zeichen im Feld "Anzeigenamen" (vgl. Abbildung 6-22) entsprechend ersetzt, z. B. wird bei folgender Definition (mit Leerzeichen bei 0x2E)

```
0x2E=" _ "  
0x20=" _ "
```

ein Punkt . in einen doppelten Unterstrich umgewandelt:

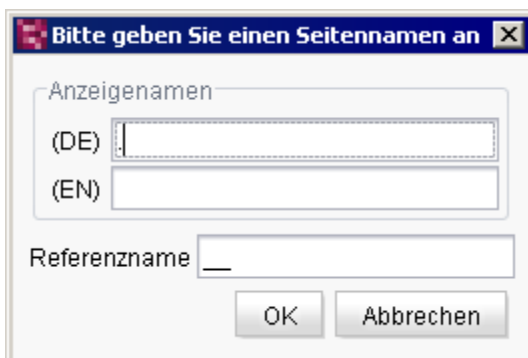


Abbildung 6-22: Umgeformter Referenzname

Im Feld "Referenzname" und / oder wenn keine Definition für das Sonderzeichen vorhanden ist, wird es bei der Eingabe direkt entfernt.

Zusätzlich können Kommentare eingefügt werden, die mit # eingeleitet werden müssen.



6.9.2 Symbole und äquivalente ASCII-Codierung (hexadezimal)

Bezeichnung	Zeichen	Hexadezimalwert
Leerzeichen		0x20
Ausrufungszeichen	!	0x21
Anführungszeichen	"	0x22
Rauten-Zeichen	#	0x23
Dollarzeichen	\$	0x24
Prozentzeichen	%	0x25
kaufm. Und-Zeichen	&	0x26
Apostroph	'	0x27
runde Klammer links	(0x28
runde Klammer rechts)	0x29
Stern	*	0x2A
Plus-Zeichen	+	0x2B
Komma	,	0x2C
Minus-Zeichen	-	0x2D
Punkt	.	0x2E
Semikolon	;	0x3B
Doppelpunkt	:	0x3A
Kleiner-als-Zeichen	<	0x3C
Gleichheitszeichen	=	0x3D
Größer-als-Zeichen	>	0x3E
Fragezeichen	?	0x3F



Bezeichnung	Zeichen	Hexadezimalwert
Schrägstrich	/	0x2F
At-Zeichen	@	0x40
Unterstrich	_	0x5F



Bei der Definition von Konvertierungs-Regeln für das Symbol . (Punkt) sollte berücksichtigt werden, dass dieser in FirstSpirit für die Erzeugung von Referenznamen für Tabellen- und Verweisvorlagen verwendet und ebenfalls bei entsprechender Regel-Definition konvertiert wird.

6.10 Referenznamen: Änderung projektweit unterbinden

Seit FirstSpirit Version 4.1.24 können Referenznamen im JavaClient über das Kontextmenü "Extras" / "Referenznamen ändern" von jedem Redakteur, der das Recht "Ändern" auf dem betreffenden Objekt hat, geändert werden. Mit Einführung der FirstSpirit Version 4.2R4 kann diese Möglichkeit projektweit in den Projekteigenschaften durch den Administrator unterbunden werden, und zwar im Bereich "Optionen":



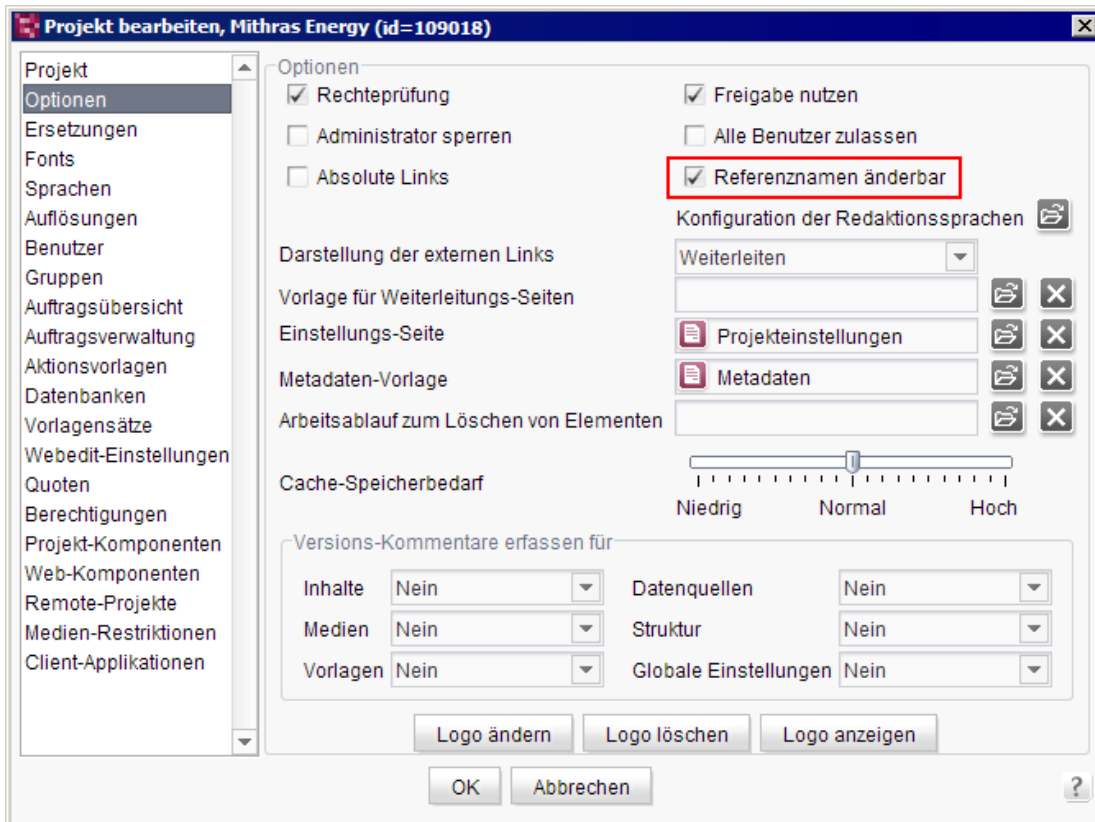


Abbildung 6-23: Projekteigenschaften – Referenznamen

Ist die Checkbox "Referenznamen änderbar" aktiviert, können Referenznamen im betreffenden Projekt geändert werden. Ist die Checkbox deaktiviert, können Referenznamen nicht geändert werden, der Menüpunkt "Referenznamen ändern" ist ausgegraut. Standardmäßig ist die Option beim Anlegen neuer Projekte aktiviert, so dass Referenznamen wie bisher geändert werden können. Unabhängig von der Einstellung in den Projekteigenschaften können Server- und Projekt-Administratoren Referenznamen jederzeit ändern.



Von der Konfiguration der Option "Referenznamen änderbar" in den Projekt-Eigenschaften ist die "Umbenennen"-Funktion (F9) im JavaClient nicht betroffen: Referenznamen von Elementen ohne UID (z. B. Absätze in der Inhalte-Verwaltung) können weiterhin über "Umbenennen" geändert werden. Änderungen an Referenznamen von Absätzen können jedoch unterbunden werden, in dem man die Option "Referenznamen anzeigen" in den Projekteigenschaften, Bereich "Optionen", "Konfiguration der Redaktionssprachen", deaktiviert. Dann ist eine Änderung nur über BeanShell (API) möglich.



6.11 Vereinfachte Encoding-Definition für Sprachen

In den Projekteigenschaften kann unter dem Punkt "Sprachen" für jede Projektsprache ein HTML-Encoding definiert werden, z. B. ISO-8859-1 für westeuropäische Sprachen. Dieses wurde bisher im Feld "HTML-Encoding" direkt eingetragen. Die Eingabe musste einer von FirstSpirit vorgehaltenen Whitelist entsprechen, nicht unterstützte Encodings wurden dabei ignoriert, d. h. es wurde immer das zuvor gespeicherte Encoding verwendet.

Mit 4.2R4 wurde das Textfeld "HTML-Encoding" in eine Combobox geändert. In dieser Combobox werden nun alle Encodings angezeigt, die von der jeweiligen Java-Version, unter der der FirstSpirit-Server läuft, unterstützt werden. Mit einem Klick auf das Feld öffnet sich die Combobox, und es kann das gewünschte Encoding ausgewählt werden:

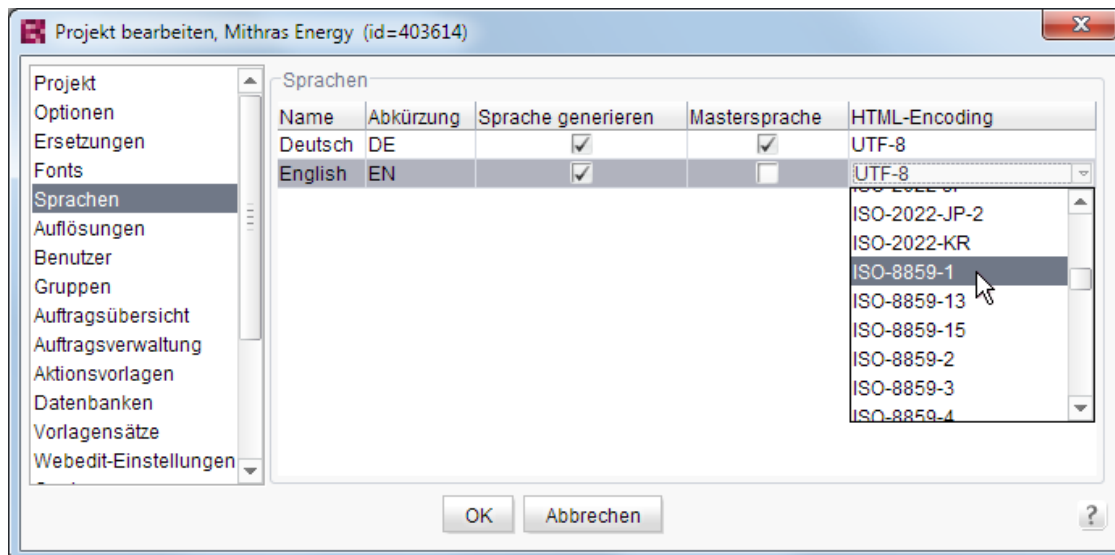


Abbildung 6-24: Projekteigenschaften – Sprachen – Encoding

Sollte bei einem Wechsel von einer JDK-Version zu einer anderen das Encoding nicht (mehr) unterstützt werden, wird mit roter Schrift visualisiert. Wird ein Projekt von einem FirstSpirit-Server auf einen anderen exportiert, der ein in dem Projekt verwendetes Encoding nicht unterstützt, wird beim Import eine Warnung in der Form

```
Warning: Language 'Deutsch' uses an unsupported HTML encoding (UTF-8)
```

ausgegeben.



6.12 Neue Funktionalitäten im Bereich Arbeitsabläufe und Aufgaben

Folgende Funktionen wurden neu implementiert, um speziell in Projekten mit vielen Aufgaben die Performance zu erhöhen:

6.12.1 Archivierung von geschlossenen Aufgaben

Über den Auftrag "Alte Projektstände archivieren" können nicht mehr benötigte Daten aus einem Projekt ausgelagert werden, um Ladezeiten zu reduzieren und die Performance des FirstSpirit-Servers zu erhöhen. Dazu werden Daten aus dem Repository eines Projekts (im Server-Verzeichnis `data\projects\`) in Archivdateien verschoben. Dabei kann festgelegt werden, welche Art von Daten archiviert werden sollen (siehe dazu auch *FirstSpirit Handbuch für Administratoren*, Kapitel "Alte Projektstände archivieren"):

- **Inhalte, Medien und Datenquellen:** mit dieser Option werden alle Inhalte der Inhalte-, Medien- und Datenquellen-Verwaltung eines Projekts archiviert.
- **Vorlagen:** mit dieser Option werden alle Inhalte der Vorlagen-Verwaltung archiviert.
- **Systemdaten:** Systemdaten sind Informationen, die zu jeder Aktion im JavaClient (z. B. Anlegen oder Löschen von Objekten, Freigaben etc.) vom System erzeugt werden. Ist diese Option aktiviert, werden nicht mehr verwendete Systemdaten archiviert. Auch Revisionen gehören zu den Systemdaten.

Ab FirstSpirit Version 4.2R4 werden jetzt auch geschlossene Aufgaben archiviert. Dazu müssen die Checkboxen "Inhalte, Medien und Datenquellen" und "Systemdaten" aktiviert sein:



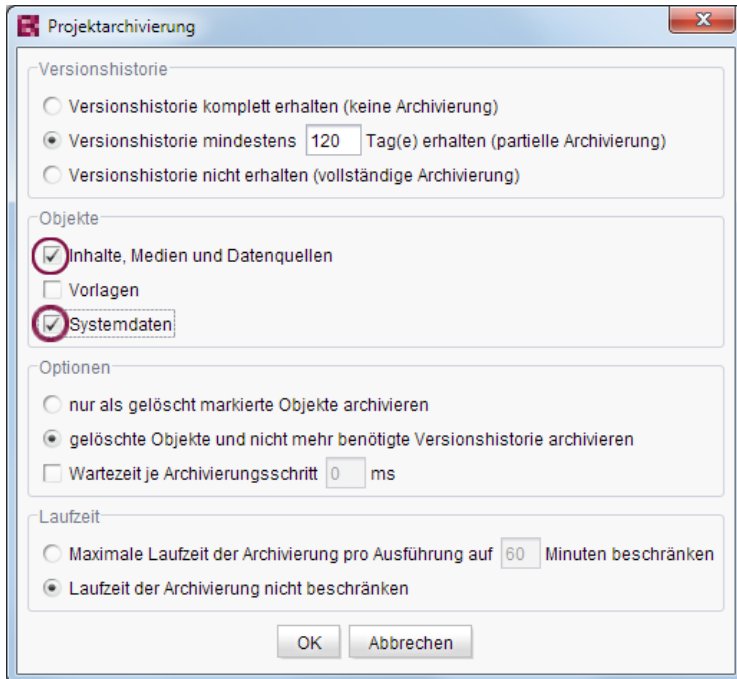


Abbildung 6-25: Archivieren von geschlossenen Aufgaben

In diesem Fall werden alle zu einer Aufgabe gehörenden Dateien archiviert, die zu einem Zeitpunkt geschlossen wurde, der innerhalb des zu archivierenden Zeitraum liegt.

6.12.2 Cache-Speicher für Arbeitsabläufe und Aufgaben konfigurieren

In der Konfigurationsdatei `fs-server.conf` kann im Bereich "Server" über die Parameter `workflow.model.cache` und `workflow.task.cache` Cache-Größe und -Typ für Arbeitsablauf-Modelle bzw. Aufgaben eingestellt werden. Der hier definierte Typ gibt Auskunft darüber, wie lange die Arbeitsablauf-Modelle bzw. Aufgaben im Cache erhalten bleiben. Dabei wird unterschieden zwischen `WEAK` und `SOFT`. Wird `WEAK` angegeben, werden die Objekte direkt aus dem Cache gelöscht, sobald sie nicht mehr verwendet werden. Bei der Angabe von `SOFT` verbleiben die Objekte, abhängig von der verwendeten VM, solange im Cache, wie genügend Speicher vorhanden ist. (Bei einem großen Speicher ist der Typ `WEAK` zumeist von Vorteil.) Die LRU-Größe wird in KB mit einem Unterstrich abgetrennt an den Typ angehängt.

Standardmäßig sind folgende Werte vorgegeben:

```
workflow.task.cache=SOFT_1024
workflow.model.cache=SOFT_128
```



6.12.3 Arbeitsablauf-E-Mails an Gruppen versenden

In den Projekteigenschaften gibt es nun die Möglichkeit, einen E-Mail-Verteiler für Gruppen anzugeben, an die Benachrichtigungs-E-Mails versendet werden, wenn eine Arbeitsablauf-Aktivität oder -Transition ausgeführt wird (vgl. dazu *FirstSpirit Handbuch für Entwickler (Teil 1: Grundlagen)*, Unterkapitel "Register Email" in den Kapiteln "Eigenschaften einer Aktivität" und "Eigenschaften einer Transition"). Dieser wird im Bereich "Gruppen" der Projekteigenschaften definiert (siehe auch Abbildung 6-29). Auf diese Weise ist es nun möglich, E-Mails, die im Rahmen eines Arbeitsablaufs versendet werden, an alle Mitglieder von externen Gruppen zu versenden.

Der Dialog beim Anlegen einer Gruppe wurde durch ein entsprechendes Feld erweitert:

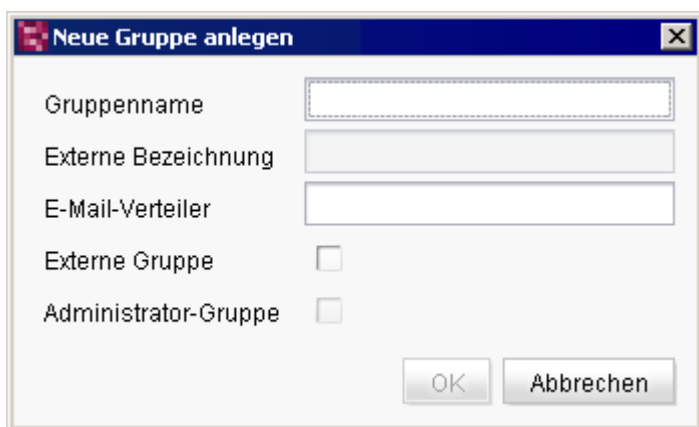


Abbildung 6-26: Neue Gruppe anlegen

Um einen E-Mail-Verteiler zu einer bestehenden Gruppe hinzuzufügen, wird auf der gewünschten Gruppe mit rechtem Mausklick das Kontextmenü aufgerufen. Hier wird der Eintrag "E-Mail-Verteiler bearbeiten" gewählt:

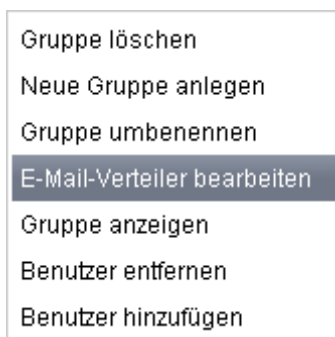


Abbildung 6-27: E-Mail-Verteiler bearbeiten



In den sich öffnenden Dialog wird die gewünschte E-Mail-Adresse eingetragen:

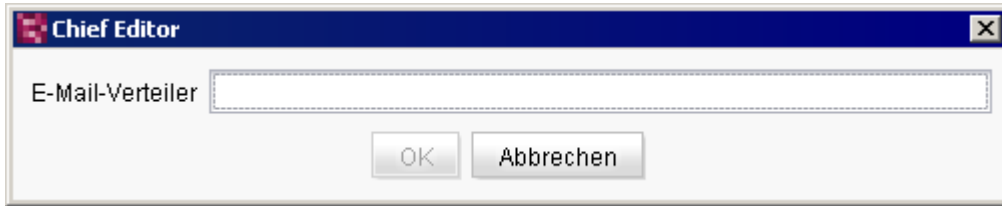


Abbildung 6-28: E-Mail-Verteiler angeben

Mehrere E-Mail-Adressen können durch Semikolon getrennt angegeben werden. Das Feld kann auch leer bleiben, bzw. bereits eingetragene E-Mail-Adressen können gelöscht werden.

Mit "OK" werden die Änderungen für die Gruppe übernommen, der E-Mail-Verteiler wird nun in der Gruppen-Übersicht angezeigt:

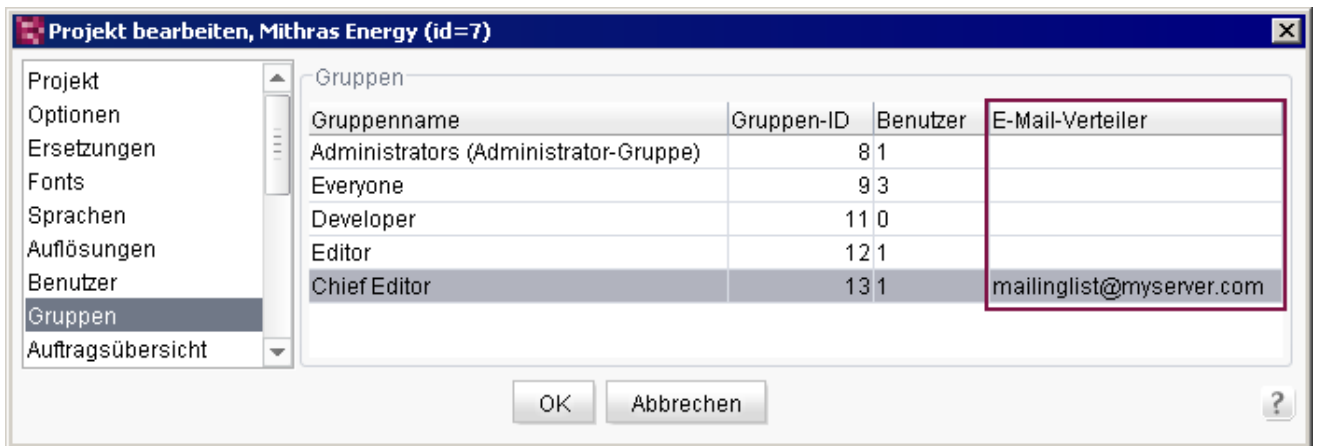


Abbildung 6-29: Projekteigenschaften – E-Mail-Verteiler für Gruppen

Die Arbeitsablauf-E-Mails werden dann zusätzlich an die hier definierten E-Mail-Adressen versendet. Wird für eine Gruppe keine E-Mail-Adresse angegeben, werden die E-Mails wie bisher an alle Benutzer der Gruppe versendet, Mitglieder von externen Gruppen erhalten potenziell jedoch keine E-Mails.



7 Neue/geänderte Funktionen in Modulen



Bei einem Update auf 4.2R4 müssen alle installierten Module aktualisiert werden, da in 4.2R4 ein neuer Signierungsschlüssel verwendet wird und es sonst zu Fehlermeldungen kommen kann. (Siehe dazu FirstSpirit Handbuch für Administratoren, Kapitel "Module".)



An dieser Stelle sei auch darauf hingewiesen, dass im Datenbank-Kontext ab 4.2R4 die Einbindung von JDBC-Treibern über Module empfohlen wird. Siehe dazu auch Kapitel 6.1 ab Seite 119.



Die Installation oder Aktualisierung von Modulen sollte nur innerhalb von "Wartungsfenstern" vorgenommen werden, da es sonst z. B. bei einer laufenden Generierung zu Fehlermeldungen (z. B. NullPointerException) kommen kann.

7.1 FirstSpirit Portal SAP EP BP



Für detaillierte Informationen zum Modul FirstSpirit Portal SAP EP BP siehe auch betreffende Modul-Dokumentation.

7.1.1 Auswertung von Gruppen in CMS_INPUT_PERMISSION

Mithilfe des Moduls "FirstSpirit Portal SAP EP BP" kann eine rollenabhängige Navigation realisiert werden. Diese Rollen werden über die Eingabekomponente CMS_INPUT_PERMISSION zugewiesen, die eine hierarchische Definition von Gruppen ermöglicht. Die hier vorgenommenen Einstellungen werden in die XML-Datei für die Navigation übernommen, die wiederum die Grundlage für die Navigation im Portal liefert. Jede Rolle in dieser XML-Datei wird im Projekt angelegt und erhält eine eigene Navigation in der Struktur-Verwaltung.

Bisher wurden dabei nur die berechtigten Gruppen auf unterster Ebene, also ohne



Untergruppen, berücksichtigt. Mit 4.2R4 werden jetzt die berechtigten Gruppen **aller** Ebenen berücksichtigt, auch solche, die vielleicht nur als Strukturierungshilfe verwendet wurden.



Daraus resultiert, dass in allen Projekten, die dieses Modul verwenden, zusätzliche Gruppen bzw. Rollen sowie zusätzliche Navigations-Knoten in der Struktur-Verwaltung hinzugefügt werden.

7.1.2 Änderung des Skripts "Portal_createLink"

Das Skript zur Link-Erzeugung "Portal_createLink" ist mit Version 4.2R4 verändert und die Klasse `CreatePortalLink` so erweitert worden, dass diese direkt das Interface implementiert, um das Aufrufen des Beanshell-Interpreters zu vermeiden.

Nach einem Update des Moduls "FirstSpirit Portal SAP EP BP" sollte in allen betreffenden Projekten der Aufruf des Skripts "Portal_createLink" folgendermaßen umgestellt werden:

```
#!executable-class  
de.espirit.firstspirit.opt.portal.script.CreatePortalLinkExecutor
```

7.2 FirstSpirit Personalisation

Um Open-Redirect-Angriffe zu unterbinden, sind in Projekten, die das Modul FirstSpirit Personalisation verwenden, jetzt externe Redirects verboten. D. h. es sind nur relative und absolute URLs möglich (z. B. `start.jsp`, `../bereich/index.html`, `../bereich/search.jsp`). Ausnahme: Redirects auf denselben Host oder einen Host in der gleichen Domain sind zulässig. Soll zu einer externen URL weitergeleitet werden, muss dafür eine Weiterleitungsseite erstellt werden.

7.2.1 Benutzer-Anmeldung über Kerberos

Ab FirstSpirit Version 4.2R4 ist die Anmeldung über Kerberos in Verbindung mit dem Modul *FirstSpirit Personalisation* möglich. Dazu wurde das neue Login-Modul "Kerberos Login" implementiert.





Detaillierte Informationen zur Konfiguration des Login-Moduls können der Modul-Dokumentation "FirstSpirit Personalisation", Kapitel „Login-Modul 'Kerberos Login'“ entnommen werden.

7.3 FirstSpirit Search

Um Open-Redirect-Angriffe zu unterbinden, sind in Projekten, die das Modul FirstSpirit Search verwenden, jetzt externe Redirects verboten. D. h. es sind nur relative und absolute URLs möglich (z. B. *start.jsp*, *../bereich/index.html*, *../bereich/search.jsp*). Ausnahme: Redirects auf denselben Host oder einen Host in der gleichen Domain sind zulässig. Soll zu einer externen URL weitergeleitet werden, muss dafür eine Weiterleitungsseite erstellt werden.



Für detaillierte Informationen zum Modul FirstSpirit Search siehe auch betreffende Modul-Dokumentation.

7.4 FirstSpirit Paket-Verwaltung

Über die FirstSpirit Paket-Verwaltung können Objekte aus einem (Master-)Projekt in ein oder mehrere Zielprojekt/e transportiert und damit wiederverwendet werden. Durch Paket-Aktualisierung können Änderungen, die an den Objekten vorgenommen werden, die Teil eines Paketes sind, einfach in die Zielprojekte übernommen werden.

In vielen Fällen spielt die Reihenfolge, in der die Objekte im Masterprojekt innerhalb einer Objektkette vorliegen, eine Rolle. Diese muss sowohl beim initialen Importieren ins Zielprojekt sowie beim Importieren von geänderten Objekten so gut wie möglich beibehalten werden; ebenso in Fällen, in denen Objektketten im Zielprojekt erweitert werden. Folgende Änderungen wurden in Version 4.2R4 vorgenommen, dank derer die Reihenfolge von importierten Objekten innerhalb einer Objektkette noch logischer und stringenter übernommen bzw. eingehalten werden kann:

Bei der Einsortierung von Objekten aus dem Master- in das Zielprojekt ist die Bindung an das vorangehende Objekt ("Vorgänger") vorrangig. D.h.,

- existiert sowohl ein Vorgänger als auch ein Nachfolger, so wird das neue Objekt nach dem Vorgänger eingefügt,
- existiert nur ein Vorgänger, wird nach dem Vorgänger eingefügt,



- existiert nur ein Nachfolger, wird vor dem Nachfolger eingefügt

Zuvor waren Objekte stärker an das nachfolgende Objekt ("Nachfolger") gebunden, soweit ein Nachfolger vorhanden war.

Beispiel 1:

Im Masterprojekt ist eine Seite mit 3 Absätzen in folgender Reihenfolge enthalten:

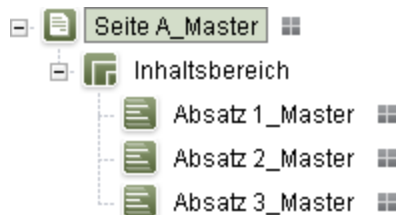


Abbildung 7-1: Paket-Verwaltung – Seite mit drei Absätzen

Wird ein Paket mit diesen Objekten in das Zielprojekt ausgerollt, bleibt die Reihenfolge der Absätze erhalten.

Der Seite werden nun 3 weitere Absätze hinzugefügt, und zwar **vor** den bereits vorhandenen:

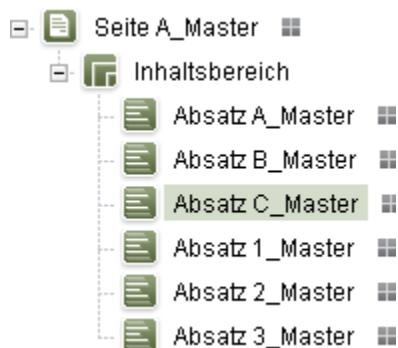


Abbildung 7-2: Paket-Verwaltung – Neue Absätze

Absatz C bindet sich dabei stärker an den Vorgänger, so dass er beim Ausrollen wie erwartet nach Absatz B eingefügt wird.



Beispiel 2: Im Zielprojekt wird in eine Seite mit 3 Absätzen, die aus dem Masterprojekt stammen, ein Absatz an letzter Position eingefügt:



Abbildung 7-3: Paket-Verwaltung – Kombinierte Absätze im Zielprojekt

Im Masterprojekt wird ebenfalls an letzter Position in die Seite ein Absatz eingefügt:

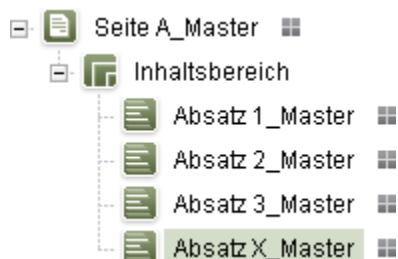


Abbildung 7-4: Paket-Verwaltung – Masterprojekt

Beim Ausrollen des aktualisierten Pakets in das Zielprojekt wird nun der neu hinzugefügte Absatz aus dem Masterprojekt **vor** dem neu hinzugefügten Absatz im Zielprojekt eingefügt.



Werden im Masterprojekt Absätze umsortiert und in das Zielprojekt ausgerollt, in dem diese Absätze bereits vorhanden sind, werden sie dort nicht umsortiert. Um die Reihenfolge der Absätze aus dem Masterprojekt zu übernehmen, sollte die betreffende Seite aus dem Zielprojekt gelöscht und das Paket mit den umsortierten Absätzen erneut ins Zielprojekt ausgerollt werden.



8 Anhang

8.1 Umbenennung von FirstSpirit-Modulen

Zum 01.01.2011 wurde das gesamte FirstSpirit-Modulangebot restrukturiert. Dabei wurden die Bezeichnungen der Module geändert und sie wurden sogenannten Suites zugeordnet. Durch die Vereinheitlichung der Modulnamen und die Kategorisierung soll das vorhandene Angebot einfacher und übersichtlicher dargestellt werden. Die Umstellung ist rein organisatorischer Art, am Funktionsumfang der Module ändert sich nichts.

Folgende Tabelle stellt die alten den neuen Namen gegenüber:

alt	neu	Modul-Dateiname
Multisite Management Suite		
Paket-Verwaltung / Package Pool	CorporateContent	
Database Pool	CorporateDatabase	
Remote Media	CorporateMedia	
Infrastructure Suite		
HighAvailability	HighAvailability	
EnterpriseBackup	EnterpriseBackup	
Preview Server	PreviewServer	
Generation Server	Generation Server	
API-Zugriff	API-Access	
Integration	Dynamic Database Access	<i>fs-integration.fsm</i>
Information Suite		
Backend Suite		
EnterpriseConnect	EnterpriseConnect	<i>fs-ep-connect.fm</i>

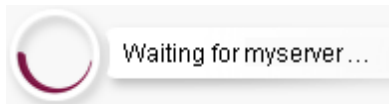


alt	neu	Modul-Dateiname
DMS-Connect	DMS-Connect	
PIM Integration	PIM-Connect	
Translation Connect	TranslationConnect	
Office	OfficeConnect	<i>fs-office.fsm</i>
Author Experience Suite		
	OfficeIntegration (ab 4.2R4)	
Workflow	WorkflowDesigner	
Newsletter	EmailMarketing	
	WebControlling (ab 4.2R4)	
	VideoManagement (ab 4.2R4)	
Portal Suite		
SAP Business Package / Portal SAP EP BP	SAP Business Package for FirstSpirit	<i>fs-portal.fsm</i>
IBM Websphere Portal	IBM Websphere Portal Integration	
	Liferay Portal Integration (ab 4.2R4)	
Customer Experience Suite		
Personalisation	DynamicPersonalization	<i>fs-perso.fsm</i>
	SocialWeb	
Search	BasicSearch	<i>fs-search.fsm</i>
Exalead	EnterpriseSearch	
Publishing Suite		
PDF Professional	PDF-Publisher	
Mobilisation	MobilePublisher	



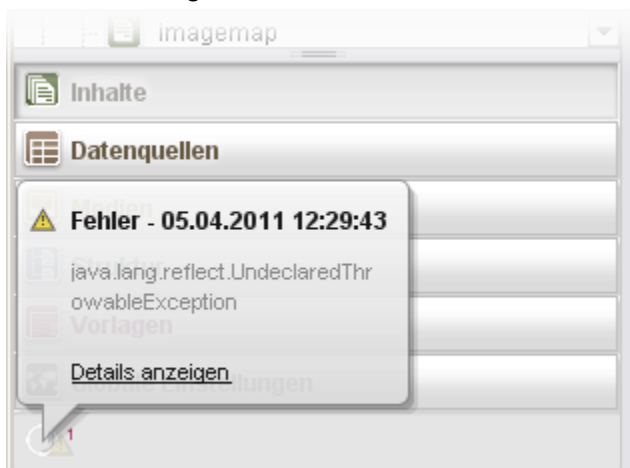
8.2 Änderungen im Softwareverhalten

- **Layout – Neues Lade-Icon und Fehlervisualisierung:** An vielen Stellen im FirstSpirit JavaClient kommt in 4.2R4 ein neues Lade-Icon zum Einsatz, z. B. in der Integrierten Vorschau:



Zusätzlich wird die Vorschau abgeschwächt dargestellt.

Auch in der linken unteren Ecke, in der auch auftretende Fehlermeldung angezeigt werden, wird dieses Icon verwendet. Tritt eine Fehlermeldung auf, wird diese nun in einer Art Sprechblase dargestellt, das Lade-Icon erhält ein Ausrufungszeichen und eine Zahl, die die Anzahl der vorhandenen Fehlermeldungen darstellt:



Mit einem Klick auf "Details anzeigen" kann wie gewohnt der Dialog mit der bzw. den vollständigen Fehlermeldung/en geöffnet werden.

- **Vergrößern / Verkleinern von Fenstern:** Fenster, die sich im JavaClient öffnen (z. B. die Übersetzungshilfe) und nicht mehr über ein Maximieren-Icon verfügen, können mit dem Tastaturkürzel F11 für Windows und Cmd + Shift + M für Mac OS maximiert bzw. wieder verkleinert werden.
- **Eingabekomponenten – Änderungen bei Sprachangaben:** Über das Tag `<LANGINFO>` können für jede Eingabekomponente sprachabhängige Angaben definiert werden, z. B. wie die Eingabekomponente in den unterschiedlichen Projektsprachen betitelt sein soll (Parameter *label*), welcher Tooltip angezeigt werden soll (Parameter *description*) usw. Aus Gründen der Übersichtlichkeit werden dabei Definitionen, die in mehreren Sprachen identisch sind, zusammengefasst. So wird z. B. aus



```
<LANGINFOS>
  <LANGINFO lang="*" label="Datum" />
  <LANGINFO lang="DE" label="Datum" />
  <LANGINFO lang="EN" label="Datum" />
</LANGINFOS>
```

beim Speichern

```
<LANGINFOS>
  <LANGINFO lang="*" label="Datum" />
</LANGINFOS>
```

Zuvor wurden Sprachangaben nur dann entfernt, wenn die Werte, wie im Beispiel oben, in allen Sprachen gleich waren. Außerdem wurden die Parameter *format*, *expression*, *ignore* und *length* bei der Zusammenfassung nicht berücksichtigt. Seit 4.2R4 werden diese Parameter berücksichtigt und innerhalb von `<LANGINFOS>` werden jetzt auch Sprachangaben zusammengefasst, wenn sie in mindestens zwei Sprachen gleich sind, z. B. wird aus

```
<LANGINFOS>
  <LANGINFO lang="*" label="Date" format="dd.MM.yy" />
  <LANGINFO lang="DE" label="Date" format="dd.MM.yy" />
  <LANGINFO lang="EN" label="Date" format="MM/dd/yy" />
</LANGINFOS>
```

beim Speichern

```
<LANGINFOS>
  <LANGINFO lang="*" label="Date" format="dd.MM.yy" />
  <LANGINFO lang="EN" label="Date" format="MM/dd/yy" />
</LANGINFOS>
```

Die Definition für den Rückfallwert (*) bleibt dabei immer erhalten, von Definitionen für Sprachen mit gleichen Werten bleibt immer die oberste erhalten. Fehlt die Pflicht-Definition des Rückfallwerts, wird beim Speichern automatisch die zuerst definierte Sprache als Rückfallwert verwendet, z. B. wird aus

```
<LANGINFOS>
  <LANGINFO lang="EN" label="Date" />
  <LANGINFO lang="DE" label="Datum" />
</LANGINFOS>
```

beim Speichern

```
<LANGINFOS>
  <LANGINFO lang="*" label="Date" />
  <LANGINFO lang="DE" label="Datum" />
</LANGINFOS>
```

Eine weitere Änderung betrifft den Parameter *description*, über den ein Text, der per Tooltip zur jeweiligen Eingabekomponente angezeigt werden soll, definiert werden kann. Wird ab Version 4.2R4 *description* nicht angegeben, wird als



Tooltip der Text angezeigt, der für den Parameter *label* definiert ist. Zuvor wurde kein Tooltip angezeigt, wenn *description* nicht gesetzt war.

- **CMS_INPUT_COMBOBOX – Verändertes Verhalten beim Löschen:** Im Formular der Eingabekomponente CMS_INPUT_COMBOBOX können über das Tag *ENTRY* dem Redakteur Einträge zur Auswahl zur Verfügung gestellt werden. Wurde in einer Eingabekomponente in der Inhalte- oder Datenquellen-Verwaltung, die auf diesem Formular basiert, bereits ein Wert ausgewählt und gespeichert und die betreffende *ENTRY*-Definition aus der Vorlage gelöscht, wird bei Aktualisierung der Eingabekomponente in der Inhalte- oder Datenquellen-Verwaltung folgende Meldung angezeigt:



Abbildung 8-1: Combobox mit Fehlermeldung

Bei erneutem Bearbeiten (z. B. über Strg + E) wird eine Meldung in der Form "Die Daten für den Editor 'Combobox' sind korrupt und können nicht angezeigt werden. Grund: Datenfehler: Der Wert '3. Wert' ist keine gültige Wahl! Der Editorwert wird zur weiteren Bearbeitung zurückgesetzt!" angezeigt. Bei Klick auf "OK" wird die Fehlermeldung (Abbildung 8-1) aus der Eingabekomponente entfernt, ebenso der Wert, der zuvor gespeichert war. Nun kann ein anderer, zur Verfügung stehender Wert gewählt werden.

Zuvor wurde der gespeicherte Wert weiterhin gespeichert, bis ein neuer Wert ausgewählt wurde.

Dasselbe neue Verhalten wurde auch für Wertemengen implementiert, die über CMS_INCLUDE_CONTENT bzw. CMS_INCLUDE_OPTIONS in eine Combobox integriert werden: Wird ein Datensatz aus einer Datenquelle gelöscht, der in der Combobox ausgewählt und gespeichert war, wird ebenfalls die oben angegebene Fehlermeldung ausgegeben und der Wert beim Wechsel in den Bearbeitungsmodus zurückgesetzt. Allerdings ist das Löschen von Datensätzen, die in einer Combobox referenziert sind, Server- und Projekt-Administratoren vorbehalten und nur nach Akzeptieren einer Sicherheitsabfrage möglich.

- **CMS_INPUT_CONTENTLIST – Hinzufügen von Datensätzen:** Wird CMS_INPUT_CONTENTLIST in einer Tabellenvorlage verwendet, die über eine n:m-Beziehung mit einer anderen Tabelle verbunden ist, wurden Datensätze nicht als geändert dargestellt, wenn über diese Eingabekomponente Datensätze hinzugefügt wurden. Wurde ein Datensatz gelöscht, wurde dies hingegen entsprechend visualisiert. In Version 4.2R4 wurde das Verhalten dahin geändert, dass nun weder beim Hinzufügen noch beim Löschen der Datensatz als geändert dargestellt wird. Aktuell existiert allerdings die Möglichkeit, die



Änderung (Hinzufügen oder Löschen) doch über einen Umweg zu visualisieren: dazu muss in der Tabelle eine Spalte vom Typ "xs:date" angelegt werden, die in der Tabellenvorlage mit einer (versteckten) CMS_INPUT_DATE-Eingabekomponente verbunden ist. Dieses Vorgehen ist allerdings nicht sicher, da sich das Verhalten der CMS_INPUT_DATE-Eingabekomponente möglicherweise in dieser Hinsicht in zukünftigen FirstSpirit-Versionen auch ändern kann.

- **CMS_INPUT_NUMBER – Formatierung von Zahlen:** Zahlen in Eingabekomponenten vom Typ CMS_INPUT_NUMBER können je nach Konfiguration durch den Vorlagen-Entwickler nach der Eingabe formatiert dargestellt werden. Z. B. kann an Währungsangaben ein Komma und 2 Nullen angehängt (aus der Eingabe "12" wird "12,00") oder überflüssige Nullen in der Eingabe entfernt werden. Ab FirstSpirit-Version 4.2R4 wird diese Formatierung aufgrund der Refactoring-Maßnahmen (siehe Kapitel 5 Seite 78) erst bei Verlassen des Bearbeitungs-Modus' (z. B. über Strg + E) vorgenommen, zuvor wurde sie bereits beim (Zwischen)Speichern (z. B. über Strg + S) vorgenommen.
- **CMS_INPUT_TOGGLE – Geänderte Visualisierung von Zuständen:** Wird CMS_INPUT_TOGGLE mit Radiobuttons verwendet (*type="RADIO"*), wurde die Eingabekomponente bislang initial mit dem Zustand *OFF* dargestellt, bis der Zustand von einem Redakteur auf *ON* geändert wurde. Es wurde allerdings kein Wert ausgegeben. Ab Version 4.2R4 wird die Eingabekomponente initial ohne Zustand dargestellt. Während der Zustand *ON* den Wert *true* und *OFF* den Wert *false* zurückliefert, liefert der neu eingeführte Zustand jetzt *null* zurück:

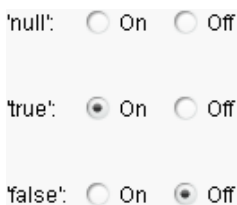


Abbildung 8-2: CMS_INPUT_TOGGLE

Der Ausgangszustand kann nach einer Änderung durch einen Redakteur nicht mehr wiederhergestellt werden.


- **FS_REFERENCE – Ordneranzeige in WebEdit:** Während für eine erfolgreiche Auswahlmöglichkeit über FS_REFERENCE in WebEdit zuvor übergeordnete Ordner nicht ausgeblendet werden durften (z. B. über *HIDE*), damit eine Navigation zu untergeordneten Elementen möglich war, werden in 4.2R4 alle Ordner, die zur Navigation zum gewünschten Element erforderlich sind, im Auswahldialog eingeblendet. So werden nun mit folgender Konfiguration beispielsweise alle Ordner oberhalb des Ordners mit dem Referenznamen "powerinverter" angezeigt:



```

<FILTER>
  <ALLOW type="media"/>
  <HIDE type="mediafolder"/>
</FILTER>
...
<SOURCES>
  <FOLDER name="powerinverter" store="mediastore"/>
</SOURCES>

```

- **Auflösungsberechnung von Bildern im JPG-Format:** Sind für ein Projekt Auflösungen für Bilder definiert (Projekteigenschaften, Bereich "Auflösungen"), werden alle Bilder in der Medien-Verwaltung bei Bedarf automatisch vom System auf die für diese Auflösungen angegebenen Größen skaliert und abgespeichert. Unabhängig vom Datei-Format des Original-Bildes (Auflösung ORIGINAL) werden dabei diese vom System erzeugten Bild-Dateien im Format PNG abgespeichert. Ausnahme: Original-Bilder im Format JPG werden in den Auflösungen auch als JPG abgespeichert. Dieses Verfahren wird ab sofort auch für Bilder, die manuell zugeschnitten werden (JavaClient, Medien-Verwaltung: Funktion "Bildausschnitt bearbeiten" bei Medien, ) , angewendet. Zuvor wurden manuell zugeschnittene Bilder im JPG-Format ebenfalls im PNG-Format abgespeichert. Darüber hinaus werden nun Bild-Dateien mit der Dateiendung JPEG nach automatischer Skalierung und manuellem Zuschneiden ebenfalls im JPG-Format gespeichert, in früheren Versionen wurden sie als PNG-Datei abgespeichert.
- **Funktion "Referenznamen ändern" projektweit deaktiviert:** Je nach Projektkonfiguration durch den Administrator kann die bisher auf den meisten FirstSpirit-Objekten zur Verfügung stehende Funktion des Kontextmenüs "Extras" / "Referenzname ändern" deaktiviert sein (außer für Server- und Projekt-Administratoren, siehe Kapitel 6.10 Seite 157):

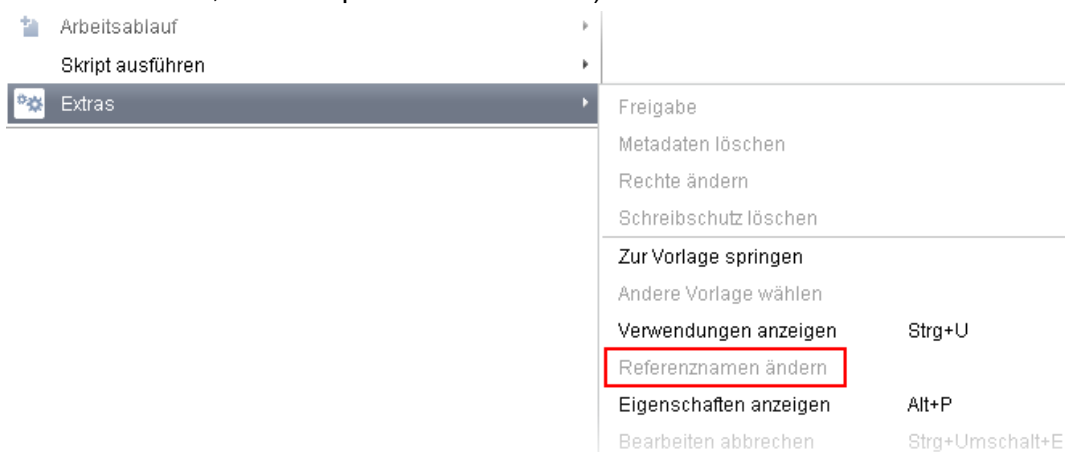


Abbildung 8-3: Projektweit deaktivierte Funktion "Referenznamen ändern"



- **Stilvorlagen: Neues Register "Eigenschaften":** Stilvorlagen (spezielle Formatvorlagen in der Vorlagen-Verwaltung), die zum Erstellen von Inline-Tabellen verwendet werden, verfügen nun über ein neues Register "Eigenschaften". Auf diesem wird der eindeutige Name (Referenzname) der Stilvorlage angezeigt. Außerdem können nun Vorgabewerte über den vorhandenen Button "Vorgabewerte" definiert werden.
- **"Verweisvorlage konvertieren":** Vor der Einführung von generischen Verweis-Editoren standen in FirstSpirit ausschließlich die drei Verweistypen „interner Verweis“, „externer Verweis“ und „Daten-Verweis“ zur Verfügung. In einer späteren FirstSpirit-Version werden die drei Verweistypen „interner Verweis“, „externer Verweis“ und „Daten-Verweis“ entfallen. Aus diesem Grund wird die Verwendung von generischen Verweis-Editoren empfohlen. Um den Umstieg auf generische Verweis-Editoren zu erleichtern, stellt FirstSpirit eine Konvertierungsfunktion für diese drei Verweistypen zur Verfügung. Die Konvertierungsfunktion ist im Kontextmenü des JavaClients unter dem Unterpunkt „Extras“ enthalten und lautet „Verweisvorlage konvertieren“. Bei der Pflege von Daten-Verweisen können die drei Eingabefelder „Datenquelle“, „Datensatz“ und „Ziel-Vorlage der Suche“ bearbeitet werden:

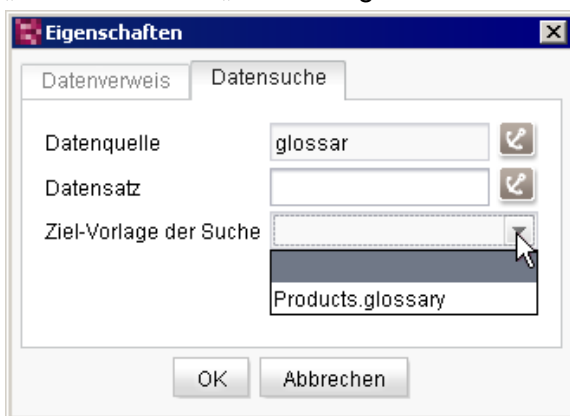


Abbildung 8-4: Daten-Verweis – Register "Datensuche"

Mit diesen Eingabefeldern ist es möglich, einen Verweis zu einem bestimmten Datensatz, der mit der ausgewählten Tabellenvorlage ausgegeben wird, zu erstellen. Nach der Konvertierung eines Daten-Verweises werden die drei Eingabefelder zu einem Eingabefeld mit der Bezeichnung „Datensatz“ (Variablenname „content“) zusammengefasst. Durch die Zusammenfassung ist es nicht mehr möglich, aus allen Tabellenvorlagen zu einer Tabelle im Schema auszuwählen (vgl. *FirstSpirit Online Dokumentation*, Kapitel „Vorlagenentwicklung / Verweisvorlagen / Konfiguration / für Daten-Verweise“, Beschreibung zu „sectiontemplate“). Bei generischen Verweisen wird nur die Tabellenvorlage zurückgeliefert, auf der die Datenquelle basiert, aus der der Datensatz ausgewählt wurde. Werden von dieser Tabellenvorlage abweichende



Tabellenvorlagen benötigt, so müssen zu diesen Datenquellen in der Datenquellen-Verwaltung angelegt und der Datensatz aus diesen Datenquellen ausgewählt werden. Alternativ kann auch in der Verweisvorlage die entsprechende Tabellenvorlage angegeben werden. In vorherigen FirstSpirit-4er-Versionen wurde bei der Ausgabe des Attributes „sectiontemplate“ mit dem Systemobjekt „#link“ dem Referenznamen der Tabellenvorlage das Präfix „template:“ vorangestellt. Dieses Präfix wird nun bei der Ausgabe nicht mehr verwendet, d. h. mit „#link.sectiontemplate“ wird nur noch der Referenzname der Tabellenvorlage ausgegeben. Die Verwendung des Systemobjektes „#link“ in generischen Verweisen wird nicht empfohlen. Stattdessen wird empfohlen, direkt den Variablennamen der einzelnen Eingabekomponente zu verwenden.

- **Rechtevergabe-Dialog auf Wurzelknoten:** Auf jedem Knoten der Baumstruktur können Benutzern bzw. Benutzergruppen durch den Projekt-Administrator Rechte zugewiesen werden. Dazu wird der Rechte-Dialog über das Kontextmenü "Extras" / "Rechte ändern" aufgerufen. Hier werden u. a. die vom höher liegenden Knoten ererbten Rechte angezeigt:

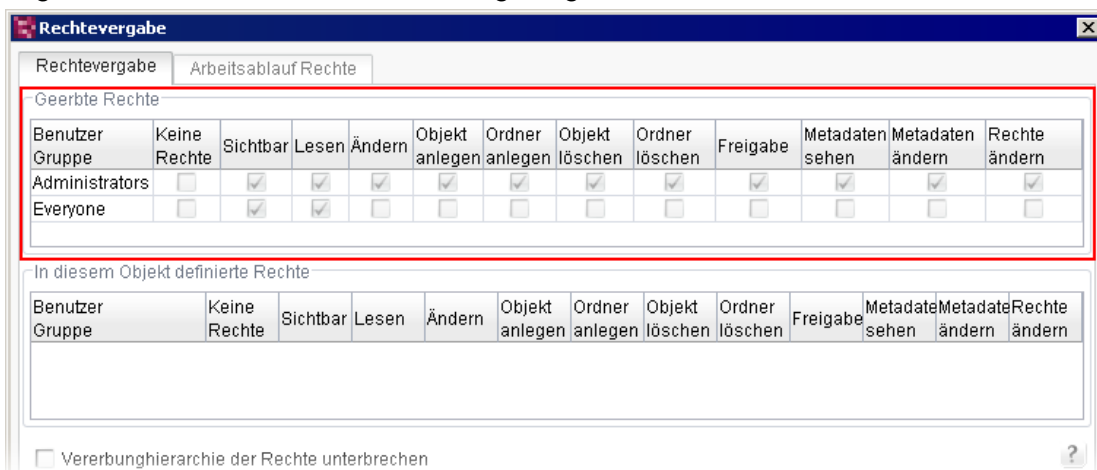


Abbildung 8-5: Dialog Rechtevergabe – Geerbte Rechte

Zuvor wurde auch auf Wurzelknotenebene der Bereich "Geerbte Rechte" angezeigt, allerdings "leer". Nun wird auf Wurzelknoten nur noch der Bereich "In diesem Objekt definierte Rechte" angezeigt:



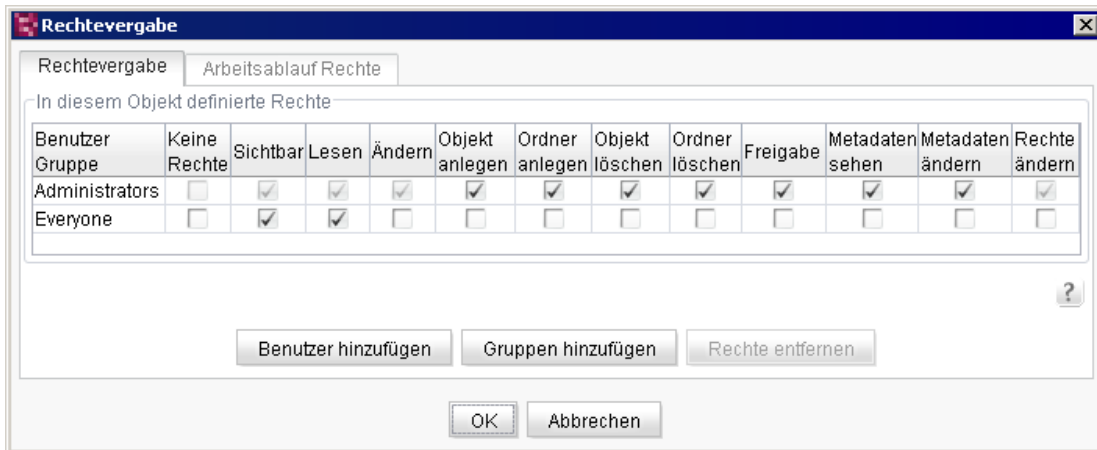


Abbildung 8-6: Dialog Rechtevergabe auf Wurzelknotenebene

Hier können die Rechte für die Wurzelknoten wie gewohnt definiert werden.

- Rechtevergabe-Dialog – Gruppe "Everyone" hinzufügen:** Wird auf einem Knoten im Rechtevergabe-Dialog die Gruppe "Everyone" über die Schaltfläche "Gruppen hinzufügen" hinzugefügt, wird die Vererbungshierarchie unterbrochen und alle Rechte, bis auf die Rechte für die Gruppe "Everyone" sowie bereits für den betreffenden Knoten im Bereich "In diesem Objekt definierte Rechte" vergebene Rechten, entzogen. Aus folgender Konfiguration

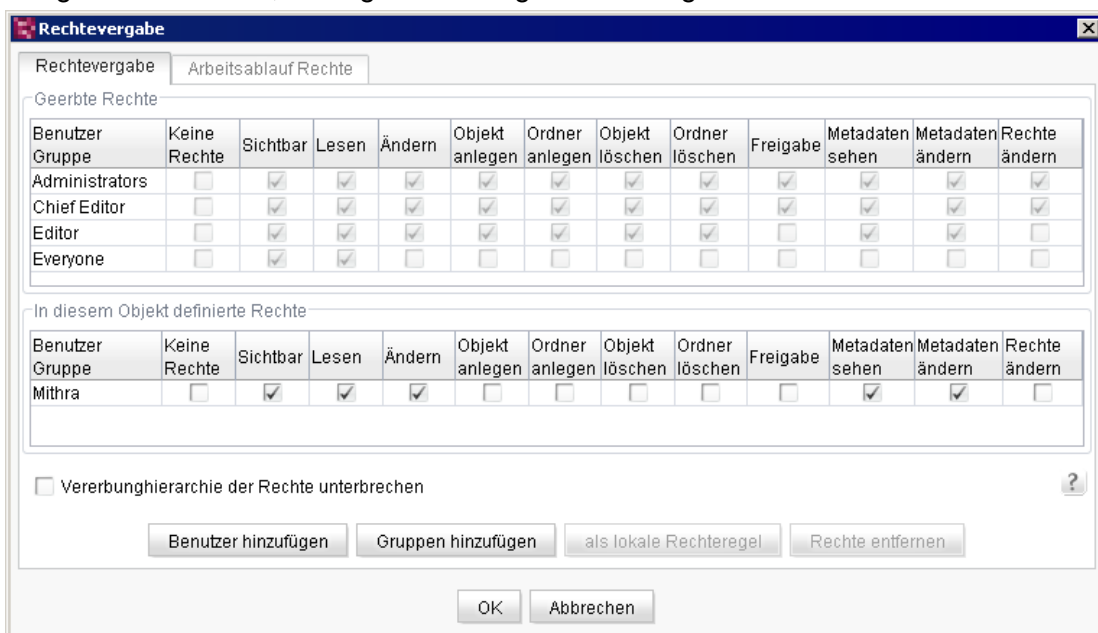


Abbildung 8-7: Gruppe "Everyone" hinzufügen – Ausgangskonfiguration

wird nach dem Hinzufügen der Gruppe "Everyone"



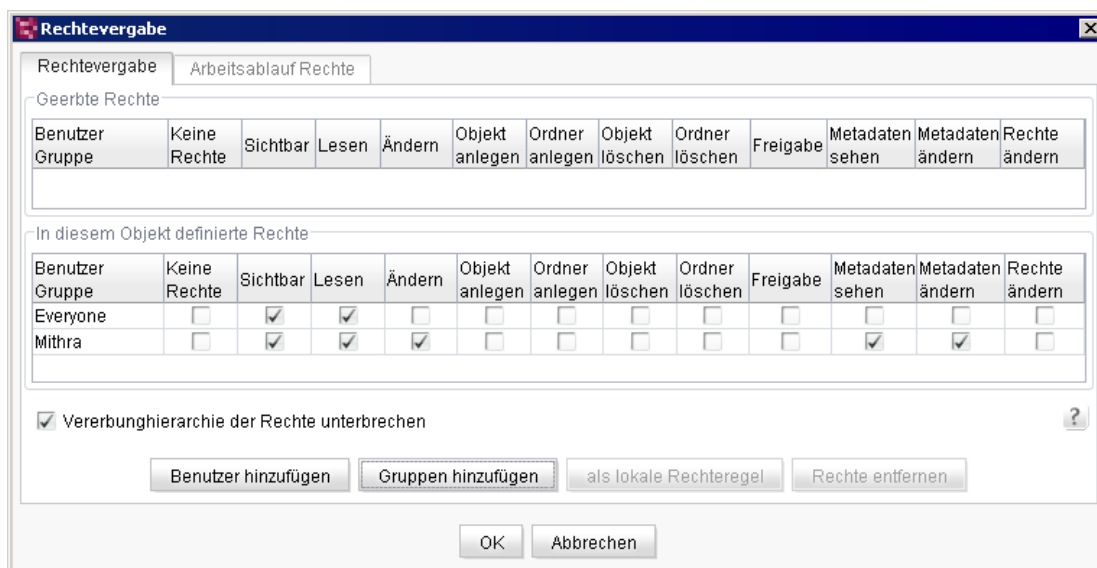


Abbildung 8-8: Gruppe "Everyone" hinzugefügt

Ab FirstSpirit Version 4.2R4 wird in diesem Fall ein entsprechender Hinweis angezeigt. Wird er mit "Ja" bestätigt, ist das Verhalten wie oben beschrieben, wird "Nein" oder "Abbrechen" gewählt, werden alle ausgewählten Gruppen bis auf "Everyone" hinzugefügt. Wurde nur "Everyone" ausgewählt, wird daher bei "Nein" oder "Abbrechen" keine Änderung vorgenommen.

- **Darstellung von Objekten ohne Rechte:** Hat ein Benutzer auf einem Objekt keine Rechte, wird dieses in der Breadcrumb-Navigation des JavaClient komplett ausgeblendet bzw. *nicht verfügbar* angezeigt. Ebenso werden Formatvorlagen, für die der Benutzer keine Rechte hat, aus der Combobox im DOM-Editor oder in der DOM-Tabelle ausgeblendet und sind somit nicht auswählbar. Zuvor wurde in beiden Fällen ****Kein Zugriff**** angezeigt.
- **Arbeitsablauf-E-Mails / FirstSpirit Adresse kopieren – Verweise auf Datensätze:** Wird ein Arbeitsablauf "weitergeschaltet", kann eine Benachrichtigungs-E-Mail an Benutzer versendet werden. Diese Benachrichtigungs-E-Mail kann einen Verweis auf den entsprechenden Knoten im JavaClient enthalten, mit dem der Benutzer direkt zum betreffenden Objekt springen kann (Platzhalter `%FIRSTspiritURL%` bzw. `%FIRSTspiritSOCKETURL%`, siehe auch *FirstSpirit Handbuch für Entwickler (Teil 1: Grundlagen)*, Kapitel "Register Email"). Im Falle von Datensätzen führte dieser Verweis bisher nur zur Datenquelle, ab 4.2R4 führt er nun zum Datensatz selbst. Auch die Funktion "FirstSpirit Adresse kopieren" (Menü "Extras") berücksichtigt nun einzelne Datensätze.
- **WebEdit-Hilfe:** Die WebEdit-Hilfe wird ab 4.2R4 nicht mehr HTML-basiert sondern als PDF bereitgestellt. Je nach Kontext, in dem die Hilfe aufgerufen wird, wird direkt das passende Kapitel aus der aktuellsten Version des *FirstSpirit*



Handbuchs für Redakteure (WebEdit) dargestellt. In welcher Sprache das PDF-Dokument angezeigt wird, richtet sich nach der über die FirstSpirit-Startseite eingestellten Client-Sprache.

- **Aktualisieren von Modulen:** Mit Version 4.2R4 wurde der Button "Aktualisieren" in den Server-Eigenschaften, Bereich "Module" entfernt. Ab 4.2R4 werden Module über den Button "Installieren" aktualisiert.
- **Auftragsverwaltung:** Für mehr Schutz vor versehentlichem Starten von Systemaufträgen aus dem JavaClient heraus (z. B. Menü "Projekt" / "Projekt generieren" bzw. "Teilprojekt generieren") können die zugehörigen Menüeinträge im JavaClient nun auch deaktiviert werden. Dazu wird in den Projekteigenschaften (Anwendung zur "Server- und Projektkonfiguration") die Option "Aktiv" deaktiviert. Denn anders als zuvor ist diese Option nun auch für Systemaufträge (gelb hinterlegt) deaktivierbar:



Abbildung 8-9: Projekteigenschaften – Auftragsverwaltung

Die Aufträge können dann nicht im JavaClient gestartet werden, da die Menüeinträge ausgegraut sind, und zwar auch für Projektadministratoren und den Serveradministrator. Unabhängig von der Auswahl bei "Interaktive Ausführung" können Aufträge von Server- und Projektadministrator im JavaClient gestartet werden.

Bei neu angelegten Aufträgen ist die Option "Aktiv" standardmäßig aktiviert, ebenso die Systemaufträge in neu angelegten Projekten, so dass sie standardmäßig von Server- und Projektadministrator im JavaClient gestartet werden können.

Wird ein Projekt importiert (Anwendung zur "Server- und Projektkonfiguration" / "Projekt" / "Importieren"), war die Option "Aktiv" der Systemaufträge nach dem Import bisher automatisch aktiv, ab 4.2R4 muss im Import-Dialog die Option "Aufträge aktivieren" aktiviert werden, damit die Option "Aktiv" aktiv wird:



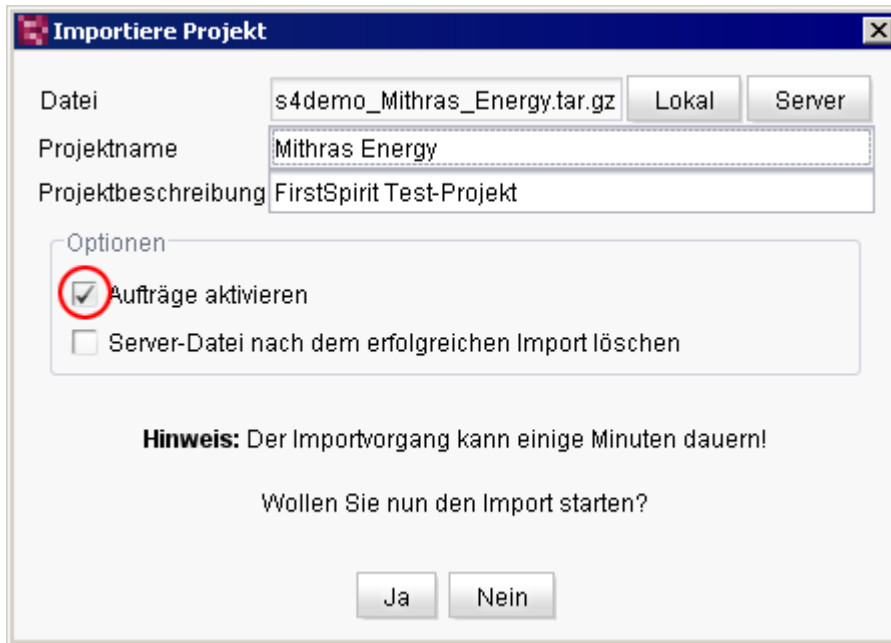


Abbildung 8-10: Projekt-Import

- **Rollout-Verzeichnis für Client-Applikationen:** Wird über den Parameter `CLIENT_HOME_DIR` in der Konfigurationsdatei `fs-server.conf` kein Pfad zu einem Verzeichnis angegeben, in das Client-Applikationen (z. B. die Mozilla-Engine zur Verwendung der Integrierten Vorschau) auf dem Arbeitsplatzrechner abgelegt werden sollen, werden Client-Applikationen ab 4.2R4 abhängig von der Version standardmäßig in ein eigenes Verzeichnis ausgerollt. Der Verzeichnisname enthält dabei die jeweilige FirstSpirit-Major-, -Minor- und -Release-Versionsnummer, z. B. `\.firstspirit_4.2R4`. Das bis 4.2R2 einschließlich verwendete Verzeichnis `\.firstspirit` wird dabei nicht vom System gelöscht, sondern muss bei Bedarf (z. B. um Speicherplatz freizugeben) manuell entfernt werden.
- **Reduzierter Standardwert für `CACHE_PERCENT`:** Der Standardwert für die prozentuale Größe des Caches (Parameter `CACHE_PERCENT` in der Datei `fs_server.conf`) wurde von 40 auf 25 gesetzt. Es ist damit zu rechnen, dass dieser Wert individuell angepasst werden muss.
- **LDAP-Konfigurationsparameter `LDAP.IMPORT_USER.LOGIN_ATTRIBUTE`:** Wird für diesen Parameter ein ungültiger Wert angegeben (z. B. LDAP-Attribut liefert einen „leeren“ Wert zurück, ungültiges LDAP-Attribut usw.), wird dies in der Server-Logdatei protokolliert:

```
INFO 17.05.2010 14:50:24.102
(de.espirit.firstspirit.server.usermanagement.LDAPAuthentication):
[LDAP] ignoring empty LOGIN_ATTRIBUTE value!
```



8.3 Auslaufende Funktionen in FirstSpirit Version 4.2R4

Mit der FirstSpirit Version 4.2R4 entfallen die folgenden Funktionen:

- **NTLM-Anmeldung und Windows 7:** Wird Microsoft Windows 7 oder Vista eingesetzt, kann das NTLM-Loginmodul, über das die NTLM-Authentifizierung am FirstSpirit-Server vorgenommen wird, nur noch verwendet werden, wenn die Windows-7-Sicherheitseinstellungen herabgesetzt werden. Eine Anmeldung über Kerberos-Ticket (Integrierte Windows-Anmeldung) ist in Kombination mit Mozilla Firefox und Microsoft Internet Explorer ab V4.2R2 möglich und in diesem Kontext die zu bevorzugende Variante. Siehe dazu auch Kapitel 6.6 Seite 147. Hinweise zur Anmeldung über Kerberos-Ticket in Verbindung mit dem FirstSpirit-Modul "Personalisation" siehe Kapitel 7.2.1 Seite 165.
- **Datensätze freigeben:** Die Funktion "Alle Zeilen freigeben" in der Datenquellen-Verwaltung im JavaClient im Kontextmenü unter "Extras" auf Datenquellen ist mit V4.2R4 umbenannt worden in "Angezeigte Zeilen freigeben" (siehe auch Kapitel 4.2.4 Seite 54).
- **Diff-Programm:** Die Möglichkeit, im JavaClient unter "Globale Einstellungen" / "Benutzereinstellungen" / Register "Browser" / "Diff-Programm" ein Programm zum Vergleich von Textdateien auszuwählen, ist entfallen, das Feld steht nun nicht mehr zur Verfügung.
- **Übersetzungshilfe / Versionsvergleich:** Im Übersetzungsvergleichs- / Versionshistorien-Dialog im JavaClient ist die Combobox "Bearbeitungsmodus" entfallen, so dass auf einem Knoten nur noch die Übersetzungshilfe oder der Versionsvergleich aufgerufen werden kann. Der Menüpunkt "Übersetzungshilfe" im Menü "Extras" kann nur noch auf Seiten und Absätzen in der Inhalte-Verwaltung oder auf Datensätzen der Datenquellen-Verwaltung aufgerufen werden, auf anderen Knoten ist der Menüpunkt ausgegraut. Der Modus "Nachübersetzen" ist entfallen, da sich bislang dahinter dieselbe Funktion verbarg wie hinter "Übersetzen".
- **WebEdit-Themes:** Die Combobox "WEBedit Theme" in den Projekteigenschaften (Anwendung zur Server- und Projektkonfiguration / "Projekt" / Bereich "WebEdit-Einstellungen") ist ab 4.2R4 nicht mehr vorhanden.



8.4 Ankündigungen für FirstSpirit Version 5.0

Während die R4-Linie als Minor-Version nur wenig Änderungen an zentralen Stellen der Software beinhaltet, die manuelle Anpassungen in Projekten nach sich ziehen, handelt es sich bei der kommenden FirstSpirit Version 5.0 um eine Major-Version, in der umfangreiche Erweiterungen und Optimierungen hinsichtlich der Software-Oberfläche (GUI), speziell für den WebClient, aber auch hinsichtlich des Server-Backends geplant sind. Daher wird es bei einem Update auf Version 5.0 projektseitige Migrationsaufwände geben. Speziell bei Verwendung von Easy-Edit und Content Highlighting werden Änderungen an den Projektvorlagen notwendig sein.

Darüber hinaus werden sich einige interne Datenformate inkompatibel ändern. Es wird einen Upgrade-Pfad von der letzten Release-Version von 4.2 auf Version 5 geben, allerdings kann dies nicht für alle Export-Datenformate garantiert werden. Es ist also möglich, dass z. B. ein Übersetzungs-Export, der mit Version 4.2 erfolgt, nach der Projektmigration auf Version 5.0 nicht mehr importiert werden kann (dies gilt analog für partielle Exporte und Pakete). Ein vollständiger Projektexport aus Version 4.2 wird hingegen auch in Version 5.0 eingelesen werden können, um so bei Bedarf eine projektweise Migration nach Version 5.0 zu ermöglichen. Aufgrund dieser Datenformat-Änderungen werden automatische Konvertierungen durchgeführt werden, die unter Umständen – je nach Projektgröße – eine signifikante Laufzeit haben.

Mit Version 5 geht die letzte Release-Version von 4.2 in die Langfristwartung, die Langfristwartung für Version 3.1 läuft aus.

Geplante Änderungen im Detail:

- **WebEdit:** In Version 5.0 wird es einen vollständig neuen WebClient geben, der sich optisch und funktional von den vorherigen Versionen deutlich unterscheidet. Maximal dabei ist eine möglichst einfache Bedienung durch den Redakteur. Dies wird sich z. B. auch auf die Gestaltung der Eingabekomponenten auswirken. Das Ziel ist dabei nicht, die bisher bekannten Eingabekomponenten komplett funktionsäquivalent in WebEdit 5.0 abzubilden, sondern eine möglichst intuitiv bedienbare, und dadurch ggf. funktionsreduzierte Variante zu realisieren. Für die FS_LIST-Variante im WebClient ist beispielsweise vorgesehen, dass sie nicht zwangsläufig alle Funktionen wie im JavaClient (siehe Kapitel 4.1.1 Seite 28 bzw. Kapitel 5.3 Seite 93) zur Verfügung stellt, sondern in erster Linie solche, die für Gelegenheitsbenutzer erforderlich und beherrschbar sind, um die Komplexität zu reduzieren und für mehr Usability und Einfachheit zu sorgen.



Darüber hinaus wird die bisher verwendete horizontale Quick-Edit-Leiste nicht mehr unterstützt und durch Easy-Edit sowie die in 4.2R4 neu eingeführte vertikale Quick-Edit-Leiste (siehe Kapitel 4.3.2 Seite 62) ersetzt.

- **Ersatz von CMS_INPUT_- durch FS_-Eingabekomponenten:** Mit FirstSpirit-Version 5.0 ist die Freigabe der seit FirstSpirit-Version 4.2 neu eingeführten Eingabekomponenten mit dem Namenspräfix "FS_" geplant. In diesem Fall werden in Version 5.0 folgende Eingabekomponenten entfallen:
 - CMS_INPUT_FILE
 - CMS_INPUT_PICTURE
 - CMS_INPUT_PAGEREF
 - CMS_INPUT_OBJECTCHOOSER
 - CMS_INPUT_CONTENTAREALIST
 - CMS_INPUT_CONTENTLIST
 - CMS_INPUT_LINKLIST
 - CMS_INPUT_SECTIONLIST
 - CMS_INPUT_TABLIST

Diese werden durch die Eingabekomponenten FS_REFERENCE, FS_DATASET bzw. FS_LIST ersetzt (siehe dazu auch Kapitel 5.1 Seite 80 und Kapitel 5.3 Seite 93).

- **CMS_INCLUDE_CONTENT:** Das Datenelement CMS_INCLUDE_CONTENT wird in FirstSpirit Version 5.0 nicht mehr unterstützt. CMS_INCLUDE_CONTENT wird durch das erweiterte Datenelement CMS_INCLUDE_OPTIONS ersetzt (siehe auch *FirstSpirit Release Notes Version 4.2* und Kapitel 5.5 Seite 107).
- **Verweise:** Mit FirstSpirit Version 4.2 sind die Konfigurationsmöglichkeiten von Verweisen durch die Einführung generischer Link-Editoren (Verweistyp "genericLink") erheblich erweitert worden.

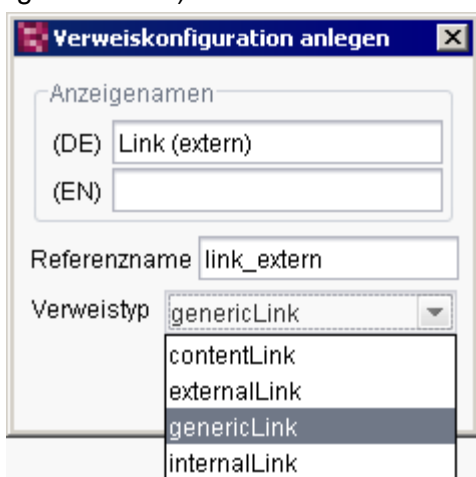


Abbildung 8-11: Verweistypen in FirstSpirit Version 4.2



Die Verweistypen "contentLink" (Daten-Verweis), "externalLink" (externer Verweis) und "internalLink" (interner Verweis) werden ab FirstSpirit Version 5.0 nicht mehr zur Verfügung stehen. Aus diesem Grund wird die Verwendung von generischen Verweis-Editoren empfohlen. Um den Umstieg auf generische Verweis-Editoren zu erleichtern, stellt FirstSpirit im Kontextmenü des JavaClients unter dem Unterpunkt "Extras" eine Konvertierungsfunktion zur Verfügung ("Verweissvorlage konvertieren").

- **FirstSpirit API:** Alle Funktionalitäten der FirstSpirit **Access-API**, die in FirstSpirit Version 4.2R4 oder davor abgekündigt wurden („deprecations“, vgl. API-Dokumentation), werden mit FirstSpirit Version 5.0 entfernt. Darüber hinaus wird es eine überarbeitete, den Kriterien der Langzeitstabilität genügende Version 2.0 der **Application-API** geben, die ab Version 5.0 nach den bekannten Strategien der Pflege stabiler APIs in FirstSpirit gewartet und weiterentwickelt wird. Sie kann inkompatibel zur Application-API in 4.2R4 sein (siehe dazu auch Kapitel 5.13 Seite 115).
- **Module:** Mit FirstSpirit Version 5.0 müssen alle Module aktualisiert werden. Für Modul-Entwickler bedeutet dies eine Neukompilation ihrer Module gegen die für FirstSpirit Version 5.0 gültige API.
- **Systemvoraussetzungen:** Ab FirstSpirit Version 5.0 wird das Oracle (Sun) JDK 7 unterstützt.

