

FirstSpirit™

Unlock Your Content

Dokumentation für Administratoren

FirstSpirit Version 5.0

Version	1.16
Status	RELEASED
Datum	2013-11-04
Abteilung	FS-Core
Copyright	2013 e-Spirit AG
Dateiname	ADMI50DE_FirstSpirit_AdminDocumentation

e-Spirit AG
Barcelonaweg 14
44269 Dortmund | Germany

T +49 231 . 477 77-0
F +49 231 . 477 77-499

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

e-Spirit

Inhaltsverzeichnis

1	Einführung	12
1.1	Thema dieser Dokumentation	17
1.2	Gliederung der Dokumentation	18
2	Systemvoraussetzungen und Installation	19
3	Steuerung des FirstSpirit-Servers	20
3.1	Unix.....	20
3.1.1	GNU/Linux und Solaris 9.....	20
3.1.2	Solaris	20
3.1.3	AIX.....	20
3.1.4	Unter normalem Benutzerkonto	21
3.1.5	Stackdump erzeugen	21
3.2	Windows.....	22
3.2.1	Registrieren / Deregistrieren als Systemdienst über Startmenü ...	22
3.2.2	Deinstallieren über das Startmenü.....	23
3.2.3	Starten / Stoppen als Systemdienst über Startmenü	23
3.2.4	Starten / Stoppen im Konsolenmodus.....	23
3.2.5	Konfigurationsdateien über Startmenü editieren	24
3.2.6	Weitere Startmenü-Funktionen	26
4	Konfiguration des FirstSpirit-Servers	27
4.1	Dateisystemorganisation.....	27
4.1.1	Dateinamen	27
4.1.2	Webanwendungen	28



- 4.1.3 Verzeichnisstrukturen..... 28
- 4.2 Allgemeines zur Konfiguration..... 29
- 4.3 Konfigurationsdateien (FirstSpirit-Server)..... 30
 - 4.3.1 Konfiguration des FirstSpirit-Servers (fs-server.conf) 31
 - 4.3.2 Konfiguration der Java-VM und des -Wrappers (fs-wrapper.conf)..... 66
 - 4.3.3 Konfiguration der Datenbankbindung (fs-database.conf)..... 76
 - 4.3.4 Konfiguration des Anmeldevorgangs (fs-jaas.conf) 77
 - 4.3.5 Konfiguration Lizenz (fs-license.conf) 94
 - 4.3.6 Konfiguration Logging (fs-logging.conf)..... 97
 - 4.3.7 Konfiguration des Webservers (fs-webapp.xml) 100
- 4.4 Anbindung an einen LDAP-Server 104
 - 4.4.1 Authentifizierung über LDAP 104
 - 4.4.2 LDAP-Attribute an einen FirstSpirit Benutzer binden 106
 - 4.4.3 Verwendung von TLS oder SSL 107
- 4.5 Einbinden in externen Webserver 108
 - 4.5.1 Apache HTTP-Server mit Servlet-Engine Jetty..... 108
 - 4.5.2 Apache HTTP-Server mit Servlet-Engine Tomcat..... 111
 - 4.5.3 Servlet-Engine Tomcat 114
 - 4.5.4 Servlet-Engine Tomcat auf dediziertem Host 121
 - 4.5.5 Externe Servlet-Engine und Lastverteilung auf mehrere Servlet-Engines..... 129
- 4.6 Einbinden in externen Application-Server..... 131
 - 4.6.1 Einbinden in Websphere Application Server..... 132
 - 4.6.2 Logging für FirstSpirit-Webanwendungen..... 134
 - 4.6.3 Anforderungen an externen Application-Server 134
- 4.7 Konfiguration des HTTPS-Servers..... 136
 - 4.7.1 Sicherheitszertifikat für einen Testserver installieren 136
 - 4.7.2 Vertrauenswürdiges Sicherheitszertifikat installieren 137



- 4.8 Datenbankanbindung 142
 - 4.8.1 Ablage der JDBC-Treiberdateien 142
 - 4.8.2 Erstellen eines JDBC-Treiber-Moduls 143
 - 4.8.3 Installation und Konfiguration des JDBC-Treiber-Moduls 150
 - 4.8.4 Konfiguration von Datenquellen 156
 - 4.8.5 Notwendige Rechte für Datenbank-Benutzerkonten 164
 - 4.8.6 Hinweise und Einschränkungen zu den einzelnen
Datenbanksystemen 166
 - 4.8.7 Beispiele für die Anbindung verschiedener Datenbanksysteme. 169
 - 4.8.8 Vorgehen bei der Anbindung externer Datenbanken 173
- 4.9 Ausrollprozess für native Applikationen 175
 - 4.9.1 Ausrollprozess (Server) 175
 - 4.9.2 Ausrollprozess (Arbeitsplatzrechner) 176
 - 4.9.3 Aktualisierung der nativen Systemkomponenten 178
 - 4.9.4 Überschreiben von Dateien beim Ausrollprozess unterbinden ... 179
 - 4.9.5 Anforderungen an das Ausrollverzeichnis 179
- 5 Konfiguration der FirstSpirit Webanwendungen 180**
 - 5.1 Konfiguration der FirstSpirit Startseite (fs5root) 180
 - 5.2 Konfiguration des WebClients 180
 - 5.2.1 Projektvoraussetzungen für den Einsatz des WebClients 181
 - 5.2.2 Der WebClient als projektlokale Anwendung 182
 - 5.2.3 Browserkonfiguration für den Einsatz des WebClients 184
- 6 FirstSpirit Startseite 185**
 - 6.1 Automatische Anmeldung über Single Sign-on 185
 - 6.2 Anmeldung mit Benutzername und Passwort 186
 - 6.3 FirstSpirit-Startseite 187
 - 6.3.1 Schnellstart 188



- 6.3.2 Clientstart 189
- 6.3.3 Administration 190
- 6.3.4 Benutzer 191
- 6.4 Starten der Anwendungen 198
 - 6.4.1 JavaClient 198
 - 6.4.2 WebClient 199
 - 6.4.3 Server-Monitoring 199
 - 6.4.4 Server- und Projektkonfiguration 199
- 6.5 FirstSpirit-Client als JAVA-Anwendung starten 200
 - 6.5.1 Socket-Modus 200
 - 6.5.2 HTTP (Internet)-Modus 202
- 7 FirstSpirit Server- und Projektkonfiguration 204**
 - 7.1 Server- und Projektadministratoren 205
 - 7.2 Elemente der Menüleiste 205
 - 7.2.1 Datei 205
 - 7.2.2 Server 206
 - 7.2.3 Projekt 209
 - 7.2.4 Benutzer 225
 - 7.2.5 Extras 229
 - 7.2.6 Hilfe 230
 - 7.3 Servereigenschaften 231
 - 7.3.1 Globale Server-Eigenschaften 231
 - 7.3.2 Präsentationskanäle 235
 - 7.3.3 Konvertierungs-Regeln 237
 - 7.3.4 Installierte Fonts 239
 - 7.3.5 Datenbanken 241
 - 7.3.6 Sprach-Vorlagen 246



7.3.7	Webstart	249
7.3.8	Startseite	250
7.3.9	Auftragsübersicht, Auftragsverwaltung und Aktionsvorlagen	252
7.3.10	JAAS-Konfiguration.....	253
7.3.11	Module	254
7.3.12	Webserver.....	261
7.3.13	Web-Applikationen	269
7.3.14	Clustering	275
7.4	Projekteigenschaften.....	283
7.4.1	Projekt.....	284
7.4.2	Optionen.....	285
7.4.3	Ersetzungen	293
7.4.4	Fonts.....	294
7.4.5	Sprachen	295
7.4.6	Auflösungen	298
7.4.7	Benutzer	301
7.4.8	Gruppen.....	303
7.4.9	Auftragsübersicht.....	310
7.4.10	Auftragsverwaltung	311
7.4.11	Aktionsvorlagen.....	311
7.4.12	Datenbanken.....	312
7.4.13	Vorlagensätze.....	313
7.4.14	WebEdit-Einstellungen.....	316
7.4.15	Quoten	323
7.4.16	Berechtigungen	323
7.4.17	Projekt-Komponenten.....	325
7.4.18	Web-Komponenten	327
7.4.19	Remote-Projekte	334



- 7.4.20 Medien-Restriktionen 338
- 7.4.21 Client-Applikationen 341
- 7.5 Auftragsplanung 349
 - 7.5.1 Auftragsübersicht 351
 - 7.5.2 Auftragsverwaltung 354
 - 7.5.3 Aktionsvorlagen 358
 - 7.5.4 Hinzufügen / Bearbeiten eines Auftrags (Register
Eigenschaften) 360
 - 7.5.5 Hinzufügen / Bearbeiten eines Auftrags (Register Aktionen) 364
 - 7.5.6 Hinzufügen von Aktionen zu einem Auftrag 365
 - 7.5.7 Kopieren von Aktionen aus einem anderen Auftrag 368
 - 7.5.8 Hinzufügen von Aktionen über Aktionsvorlagen 369
 - 7.5.9 Serverbezogene Aktionen 370
 - 7.5.10 Projektbezogene Aktionen 379
- 7.6 Clustering: Lastverteilung bei der Generierung 402
 - 7.6.1 Konzept 403
 - 7.6.2 Lizenzdatei prüfen 404
 - 7.6.3 Konfiguration der Clusterknoten 405
 - 7.6.4 Konfiguration des Generierungsauftrags 406
- 7.7 Konfiguration der Rechtschreibprüfung 408
 - 7.7.1 SpellService installieren / deinstallieren (Servereigenschaften) 408
 - 7.7.2 SpellService aktualisieren (Servereigenschaften) 410
 - 7.7.3 Globalen SpellService konfigurieren 411
 - 7.7.4 Globale Wörterbücher konfigurieren 413
 - 7.7.5 Dienst "SpellService" starten und konfigurieren 416
 - 7.7.6 SpellService als Projekt-Komponente hinzufügen 416
 - 7.7.7 Projektspezifische Spell-Service-Konfiguration 418
 - 7.7.8 Projektspezifische Wörterbücher hinzufügen 420



7.8	Unterstützung für Apache FOP	421
7.9	Projektarchivierung	423
7.9.1	Versionshistorie.....	423
7.9.2	Revisionen	423
7.9.3	Mindestanforderung an die Projektarchivierung	424
7.9.4	Versionshistorie nach Archivierung.....	426
8	FirstSpirit Server-Monitoring	429
8.1	Übersicht.....	430
8.1.1	Übersicht – Status	430
8.1.2	Übersicht – Aktivitäten	431
8.1.3	Übersicht – Sitzungen	432
8.2	Projekte	434
8.2.1	Projekte – Übersicht	434
8.2.2	Projekte – Statistik.....	435
8.3	Log-Dateien.....	438
8.3.1	Log-Dateien – kompletter Server.....	438
8.3.2	Log-Dateien – nach Projekt.....	443
8.3.3	Log-Datei – nach Veröffentlichung.....	444
8.4	Zeitplanung.....	446
8.4.1	Zeitplanung – Übersicht.....	446
8.4.2	Zeitplanung – geplante Aufträge.....	447
8.4.3	Zeitplanung – ausgeführte Aufträge	448
8.5	Benutzer	449
8.5.1	Benutzer – Suche	449
8.6	FirstSpirit	450
8.6.1	FirstSpirit – Konfiguration.....	450
8.6.2	FirstSpirit – Steuerung	457



8.6.3	FirstSpirit – Nachricht.....	466
8.6.4	FirstSpirit – Datenbanken.....	466
8.6.5	FirstSpirit – Monitoring.....	467
8.6.6	FirstSpirit – Clustering.....	475
9	FirstSpirit JMX-Konsole.....	476
9.1	Start der JMX-Konsole.....	477
9.2	MBeans	478
9.3	BerkeleyDB-Backend	481
9.3.1	Attribute	481
9.3.2	Operation	482
9.4	CacheSizeManager	483
9.4.1	Attribute (gesamt).....	483
9.4.2	Operation	484
9.4.3	Attribute (projektbezogen).....	484
9.5	Content Manager.....	485
9.6	DiscoveryManager	485
9.6.1	Attribute	485
9.6.2	Operations	485
9.6.3	Attribute (serverbezogen)	485
9.7	Event Manager.....	486
9.8	ExecutionManager.....	487
9.8.1	Klassifizierung der Thread Queues.....	487
9.8.2	Verarbeitung innerhalb des ExecutionManagers.....	489
9.8.3	Attribute	490
9.9	LockManager	492
9.10	MediaManager	493
9.11	ModuleManager.....	493



9.12 NIOSocketServer.....	494
9.13 PreviewManager.....	495
9.14 ProjectManager.....	496
9.14.1 Attribute	496
9.14.2 Operations	497
9.15 ReferenceManager	498
9.15.1 Attribute	498
9.15.2 Operations	499
9.16 RegistryManager	499
9.17 RepositoryManager	500
9.18 SearchManager	500
9.18.1 Attribute:.....	501
9.18.2 Operations	501
9.19 ServerActionManager	502
9.19.1 Attribute	502
9.19.2 Operations	502
9.20 ServerManager	503
9.20.1 Attribute	503
9.20.2 Operations	503
9.21 ServiceManager.....	504
9.21.1 Attribute	504
9.21.2 Operations	504
9.22 SessionManager.....	505
9.22.1 Attribute	505
9.22.2 Operations	505
10 Sichere Veröffentlichung über rsync und ssh.....	506
10.1 Webserver unter Unix	506



10.2 Webserver unter Windows	507
10.3 FirstSpirit-Server unter Unix.....	508
10.4 FirstSpirit-Server unter Windows.....	509
10.5 Konfiguration des FirstSpirit-Projekts.....	511
11 Konfiguration von Benutzerrechten	512
11.1 Einführung	512
11.1.1 Definition von Benutzerrechten	513
11.1.2 Prüfung von Benutzerrechten.....	514
11.1.3 Vergabe von Benutzerrechten.....	515
11.2 Architektur.....	517
11.2.1 Einleitung	517
11.2.2 Übersicht	518
11.2.3 Berechtigungsprüfung mit FirstSpirit	522
11.3 Schutz personalisierter Projektinhalte in FirstSpirit	523
11.3.1 Personalisierung	523
11.3.2 Prüfung der Zugriffsrechte über die Access-Control-Datenbank	525
11.3.3 Konzept Sichere Medien	527
11.3.4 Abgrenzung	528
11.3.5 Berechtigungsdefinition	529
11.3.6 Definitionen.....	532
11.4 Konfiguration.....	533
11.4.1 Einleitung	533
11.4.2 Aktivierung des Permission Service	533
11.4.3 Konfiguration der Server-Komponente	533
11.4.4 Konfiguration für die Veröffentlichung.....	539
11.4.5 Konfiguration der Berechtigungen über die Projekteigenschaften.....	539
11.5 Einsatz im Live-System	540



11.5.1 Konfiguration des Servlet-Servers.....	540
11.6 Einsatz im Projekt.....	541
11.6.1 Manuelle Gruppendifinition und ID-Vergabe	541
11.6.2 Konfigurationssemantik	543
12 Anhang.....	545
12.1 fs-wrapper.conf	545
12.2 fs-jaas.conf	549
12.3 fs-webapp.xml	550



1 Einführung

FirstSpirit™ ist eine klassische Client-Server-Anwendung, die komplett auf Java- und Webtechnologie basiert. Der Zugriff auf den FirstSpirit™-Server erfolgt entweder über den FirstSpirit™-JavaClient oder über den FirstSpirit™-WebClient. Die Kommunikation zwischen Client und Server basiert, je nach Einsatzszenario, entweder auf HTTP(S) unter Verwendung eines Application-Servers oder auf der direkten Verwendung von TCP.

Das Live-System übernimmt die eigentliche Auslieferung von Inhalten an den Endbenutzer. Der FirstSpirit™-Server muss dazu nicht zwangsläufig immer in Verbindung zum Live-System stehen. Besteht keine dauerhafte Verbindung zum Live-System, werden alle vom Live-System benötigten Daten im Rahmen der Veröffentlichung (Deployment) übertragen, so dass keine direkte Interaktion des Live-Systems mit dem FirstSpirit™-Server notwendig ist.



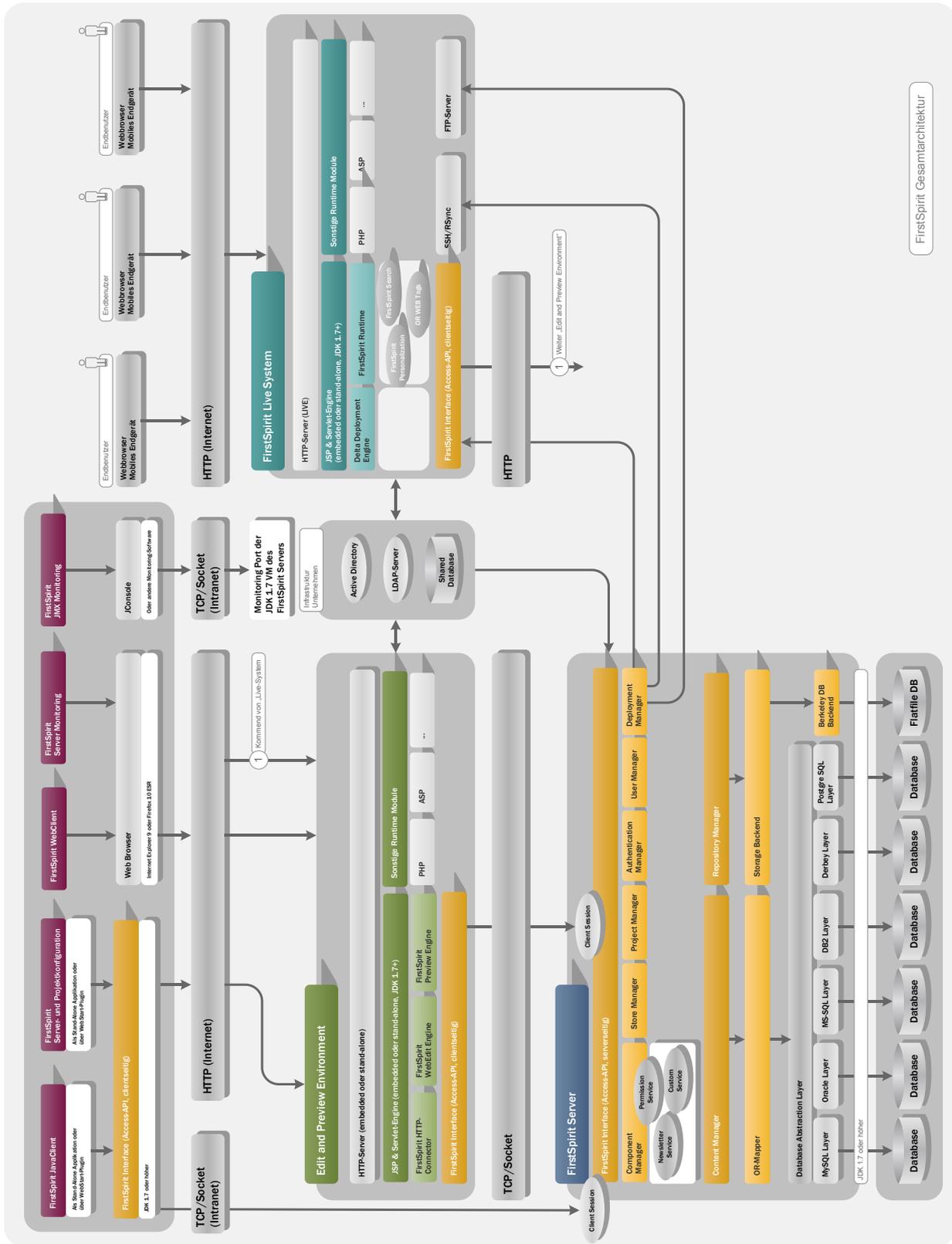


Abbildung 1-1: FirstSpirit Gesamtarchitektur



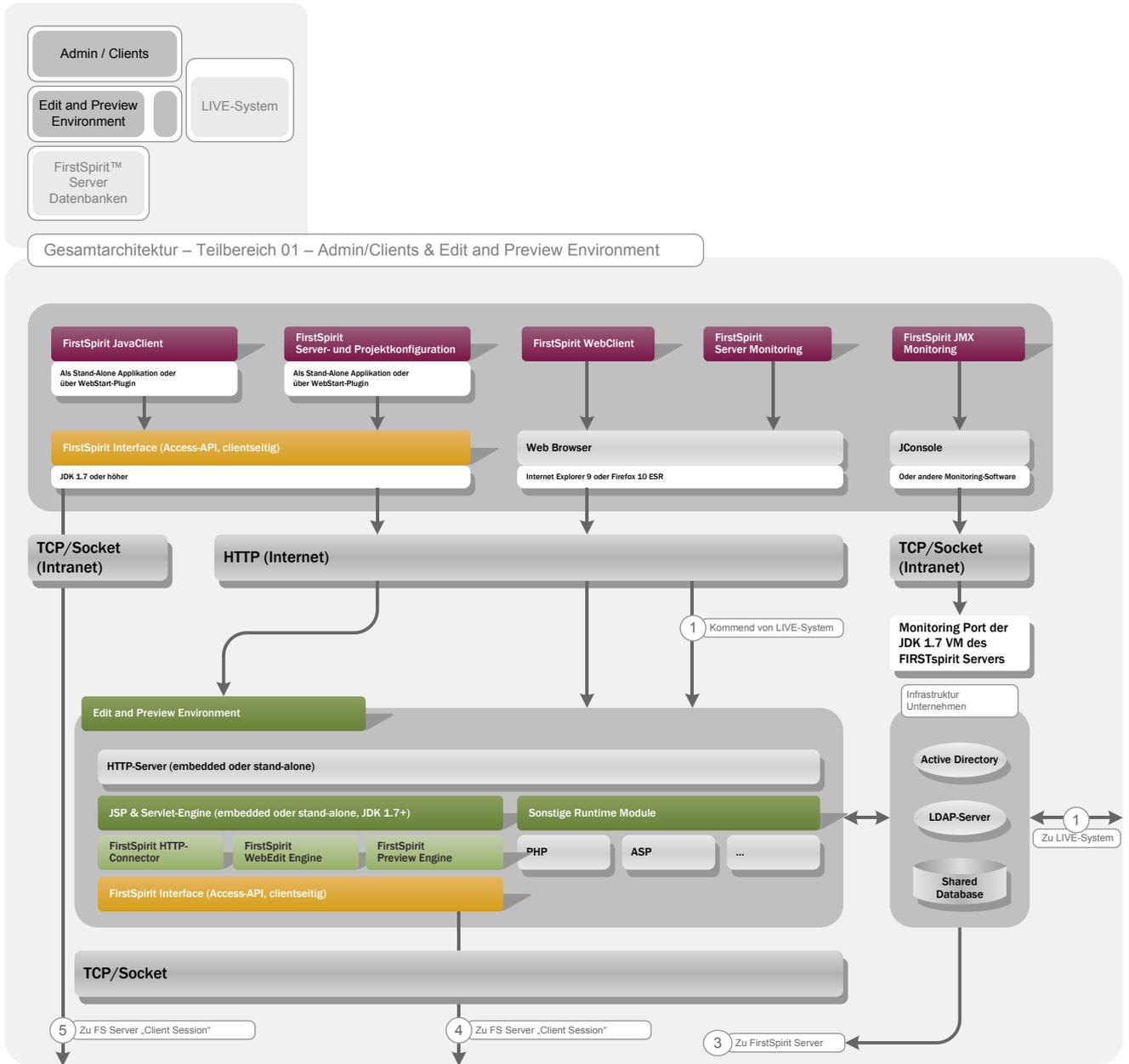


Abbildung 1-2: Gesamtarchitektur – Detail: Edit- and Preview Environment



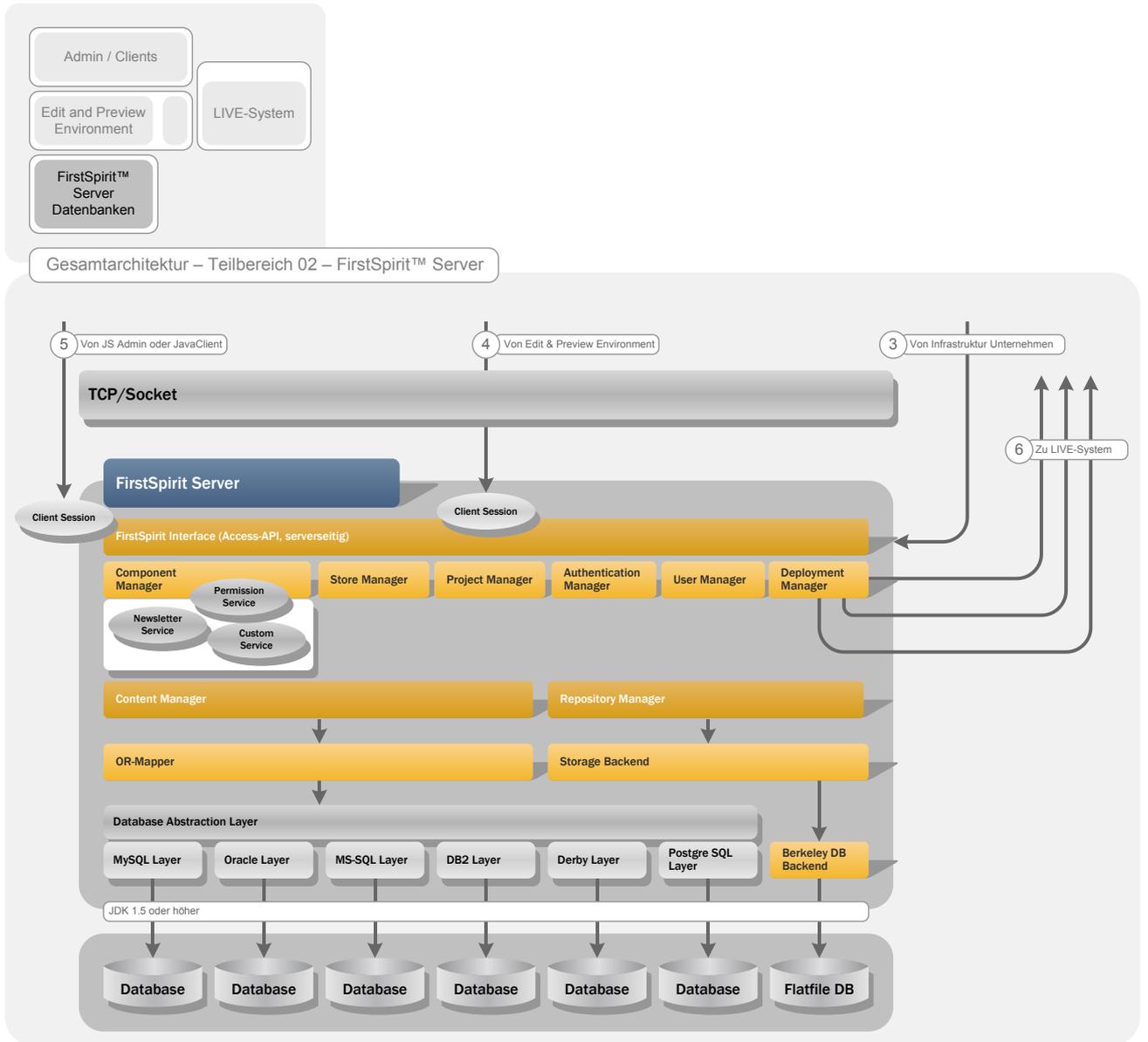


Abbildung 1-3: Gesamtarchitektur – Detail: FirstSpirit-Server



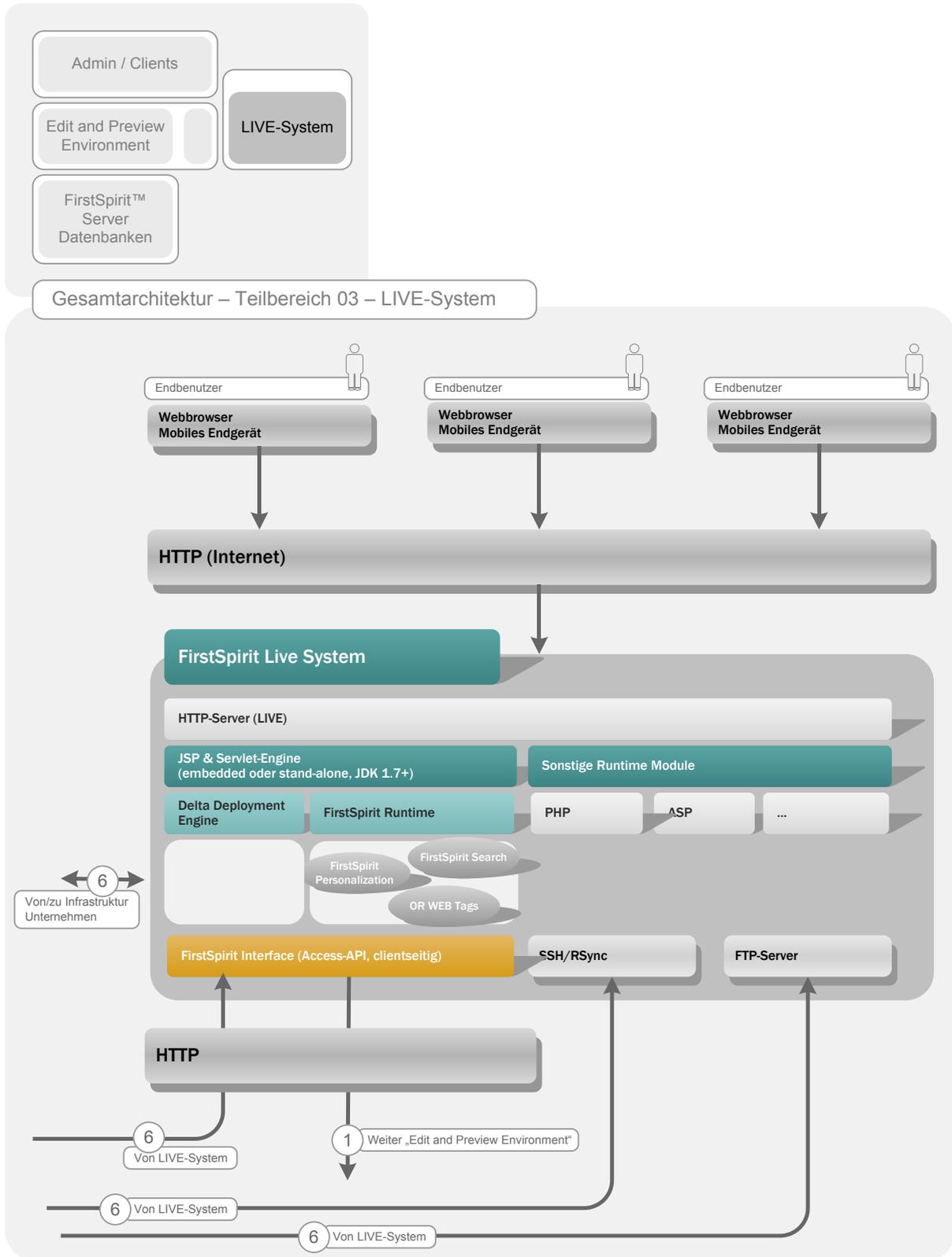


Abbildung 1-4: Gesamtarchitektur – Detail: Live-System



1.1 Thema dieser Dokumentation

FirstSpirit stellt allen Anwendern, abhängig von ihren Aufgaben, einen exakt auf die jeweiligen Bedürfnisse angepassten Client zur Verfügung. FirstSpirit bietet damit also mehrere Clients für unterschiedliche Aufgaben und Benutzer an. Grundsätzlich kann dabei zwischen Redaktionsumgebungen und Administrationsumgebungen unterschieden werden. Während die FirstSpirit-Redaktionsumgebungen die Arbeit der Redakteure und der Vorlagenentwickler unterstützen, sind die Administrationsumgebungen primär auf die Überwachung und Konfiguration von FirstSpirit zugeschnitten.

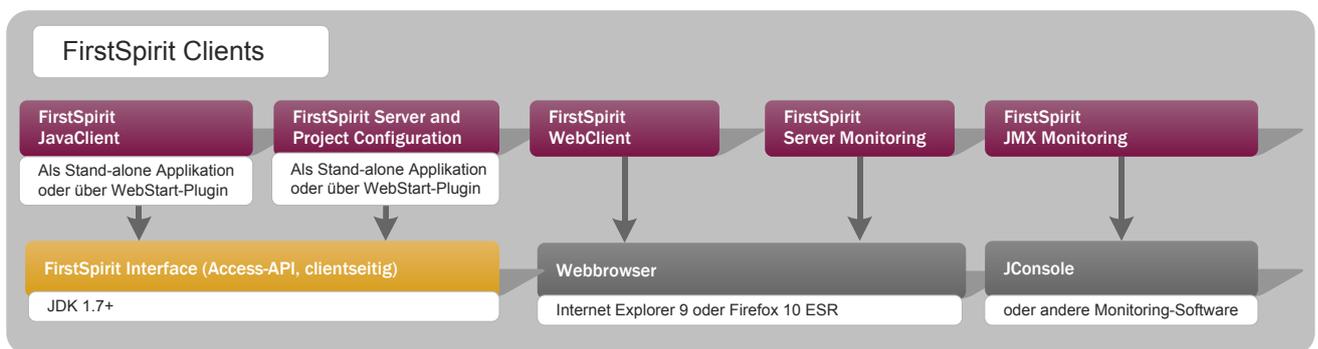


Abbildung 1-5: FirstSpirit-Redaktionsumgebungen

Die **Dokumentation für Administratoren** beschreibt alle Aspekte zur Administration von FirstSpirit V5.0 und die erforderlichen Administrationsumgebungen und richtet sich daher ausschließlich an Administratoren. Die FirstSpirit-Redaktionsumgebungen (JavaClient und WebClient) werden in der gesonderten Dokumentation für Redakteure im Einzelnen beschrieben.

- **FirstSpirit Server- und Projektkonfiguration:** Die FirstSpirit Server- und Projektkonfiguration ist eine Java-Anwendung mit einer komfortablen Swing-Benutzeroberfläche, die den FirstSpirit-Administrator bei allgemeinen, administrativen Aufgaben rund um FirstSpirit unterstützt. Über die Oberfläche können beispielsweise neue FirstSpirit-Projekte angelegt und konfiguriert werden. Neben den allgemeinen Aufgaben sind jedoch auch weiterreichende Funktionen möglich. Beispielsweise kann die Server- und Projektkonfiguration zur Benutzerdefinition bzw. zur Anbindung bereits vorhandener Identity-Management-Systeme, wie z. B. LDAP oder Active-Directory, verwendet werden. Analog zum JavaClient wird die Server- und Projektkonfiguration über Java-Web-Start gestartet und aktualisiert (siehe Kapitel 7).
- **FirstSpirit Server-Monitoring:** Das browserbasierte FirstSpirit Server-Monitoring ist eine Webapplikation, die zur Überwachung des FirstSpirit-Servers



dient. Mit ihrer Hilfe können aktuelle Betriebsparameter, wie Speicherauslastung, Benutzeranzahl, u.v.m. angezeigt werden. Das FirstSpirit Server-Monitoring wird analog zum WebClient über einen Webbrowser bedient (siehe Kapitel 8).

- **FirstSpirit JMX-Konsole:** Mithilfe der Java Management Extensions (JMX) ist es möglich, Java-Anwendungen auf einheitliche Weise zu überwachen. Während das FirstSpirit Server-Monitoring primär auf die manuelle Überwachung eines FirstSpirit-Servers abzielt, dient die JMX-Konsole zur automatischen Überwachung und integriert sich optimal in ein ggf. bereits vorhandenes, unternehmensweites Monitoring. Grundsätzlich kann die JMX-Konsole aber auch interaktiv verwendet werden. Insgesamt werden im Vergleich zum Server-Monitoring feingranularere Informationen zur Verfügung gestellt (siehe Kapitel 9).

Hinweis: FirstSpirit ist kein universelles "out-of-the-box"-Produkt, sondern eine Software, die sich in ständiger Entwicklung befindet. Es werden kontinuierlich neue Funktionalitäten eingebunden und Kundenanregungen verwirklicht. Diesem ständigen Aktualisierungsprozess kann die Dokumentation nur in begrenztem Umfang gerecht werden. So ist es möglich, dass aufgrund geringer Zeitverzögerungen einige Abbildungen in dieser Dokumentation von der aktuellen Ansicht in FirstSpirit abweichen. Lassen Sie sich davon nicht verwirren und folgen Sie der Anleitung wie gewohnt.

1.2 Gliederung der Dokumentation

Die Gliederung dieser Dokumentation orientiert sich an den Funktionen, die durch die Administrationsumgebungen zur Verfügung gestellt werden.

Kapitel 2: Nach den einleitenden Erläuterungen im ersten Kapitel wird in diesem Kapitel ein kurzer Überblick über die Installation von FirstSpirit gegeben. Eine ausführliche, gesonderte Dokumentation zu diesem Thema befindet sich in der "FirstSpirit Installationsanleitung" (Überblick zur Installation ab Seite 19).

Kapitel 3: In diesem Kapitel wird die Steuerung des FirstSpirit-Servers unter Unix- und Windows-Betriebssystemen erläutert (ab Seite 20).

Kapitel 4: Die Konfiguration des FirstSpirit-Servers erfolgt über Konfigurationsdateien, die sich im Installationsverzeichnis des FirstSpirit-Servers befinden. Die Beschreibung der Konfigurationsdateien und ihrer Parameter befindet sich in diesem Kapitel (ab Seite 27).

Kapitel 5: Die Konfiguration der FirstSpirit-Webanwendungen – insbesondere die Projektvoraussetzungen, Konfiguration und Einschränkungen des WebClients – wird in diesem Kapitel beschrieben (ab Seite 180).



Kapitel 6: Die FirstSpirit Startseite ist eine Webanwendung zum Starten der Redaktions- und Administrationsumgebungen von FirstSpirit (ab Seite 185).

Kapitel 7: Das Kapitel beschreibt projektübergreifende Funktionen, die über die Menüleiste der Server- und Projektkonfiguration aufgerufen werden können. Die FirstSpirit Server- und Projektkonfiguration, unterstützt den FirstSpirit-Administrator bei allgemeinen, administrativen Aufgaben rund um FirstSpirit (ab Seite 204).

Kapitel 8: Das browserbasierte FirstSpirit Server-Monitoring dient zur Überwachung des FirstSpirit-Servers und zeigt aktuelle Betriebsparameter an (z.B. Benutzeranzahl, Speicherauslastung) (ab Seite 429).

Kapitel 9: Mithilfe der Java Management Extensions (JMX) ist es möglich, Java-Anwendungen automatisch zu überwachen. Die JMX-Konsole kann optimal in ein ggf. bereits vorhandenes, unternehmensweites Monitoring integriert werden (ab Seite 476).

Kapitel 10: Für das Deployment über ungesicherte Internetverbindungen oder Netze mit geringer Bandbreite empfiehlt sich die Verwendung einer Kombination aus den externen Dienstprogrammen `rsync` und `ssh` (ab Seite 506).

Kapitel 11: Im Rahmen des Kapitels werden die von FirstSpirit zur Verfügung gestellten Mechanismen zur Benutzer-Rechtevergabe und -Rechteprüfung beschrieben und ihr konkreter Einsatz skizziert (ab Seite 512).

2 Systemvoraussetzungen und Installation

FirstSpirit ist durch die Verwendung von Java ein weitgehend plattformunabhängiges Client-Server-System. Die Installation betrifft im Wesentlichen nur den FirstSpirit-Server, da der Client entweder über Java Web Start verwaltet wird (JavaClient) oder als Webanwendung über einen Webbrowser bedient wird (WebClient). Ein installierter FirstSpirit-Server ist, aufgrund der Plattformunabhängigkeit, in der Dateisystemstruktur auf allen Betriebssystemen einheitlich aufgebaut. Unterschiede treten nur während des Ablaufs der Installation auf. Hierbei werden alle Dateien des FirstSpirit-Servers, bis auf einige wenige zum Systemstart notwendige, in das Zielverzeichnis installiert.

Detaillierte Informationen zu den Systemvoraussetzungen für FirstSpirit 5.0 siehe "FirstSpirit Technisches Datenblatt 5", zu Installation und zur Aktualisierung von FirstSpirit-Servern siehe "FirstSpirit Installationsanleitung Version 5.0".



3 Steuerung des FirstSpirit-Servers

Zur Steuerung über das FirstSpirit Server-Monitoring siehe Kapitel 8.6.2.



Vor dem Starten und Stoppen des FirstSpirit-Servers sollte der Wartungsmodus aktiviert werden (siehe Kapitel 8.6.2.1 Seite 457).

3.1 Unix

3.1.1 GNU/Linux und Solaris 9

Starten als `root`:

```
/etc/init.d/fs5 start
```

Stoppen als `root`:

```
/etc/init.d/fs5 stop
```

3.1.2 Solaris

Unter Solaris verwendet FirstSpirit die Service Management Facility.

Starten als `root`:

```
svcadm enable fs5
```

Stoppen als `root`:

```
svcadm disable -s -t fs5
```

Prozesse auflisten als `root`:

```
svcs -p fs5
```

3.1.3 AIX

Unter AIX dient zum Starten des FirstSpirit-Servers ein Eintrag in der Datei `/etc/inittab`, der mit Kennung "fs5" während der Installation erfolgt.

Zum manuellen Starten und Stoppen als `root` dienen folgende Aufrufe.



Starten als `root`:

```
/opt/firstspirit5/bin/fs5.init start
```

Stoppen als `root`:

```
/opt/firstspirit5/bin/fs5.init stop
```

3.1.4 Unter normalem Benutzerkonto

Der FirstSpirit-Server kann auch von einem normalen Benutzerkonto aus gesteuert werden. Die Standardinstallation legt das Benutzerkonto `fs5` an, deaktiviert aber die Anmeldung für dieses Benutzerkonto. Zum Aktivieren der Möglichkeit als `fs5` über SSH oder Telnet anzumelden, ist es ausreichend, das Passwort für das Benutzerkonto `fs5` einzutragen. Dazu als `root` aufrufen:

```
passwd fs5
```

Anschließend können nach Anmeldung mit dem Benutzerkonto `fs5` zur Steuerung des FirstSpirit-Servers folgende Aufrufe verwendet werden:

Starten als Benutzer `fs5`:

```
firstspirit5/bin/fs5 start
```

Stoppen als Benutzer `fs5`:

```
firstspirit5/bin/fs5 stop
```

3.1.5 Stackdump erzeugen

Über den Befehl:

```
fs5 dump
```

wird ein aktueller Thread Dump erstellt und nach `log/fs-dump-DATUM-UHRZEIT.log` geschrieben.

Eine gute Analysemöglichkeit bietet das FirstSpirit Server-Monitoring. Die über die Funktionalität "Threads" erstellten Thread Dumps können hier analysiert und in einer aufbereiteten Ansicht angezeigt werden (siehe Kapitel 8.6.5.5 Seite 470).



3.2 Windows

Unter Windows Betriebssystemen kann die Steuerung des FirstSpirit-Servers über das Startmenü vorgenommen werden. Nach der erfolgreichen Installation von FirstSpirit können über das Windows-Startmenü

Start / Alle Programme / FirstSpirit 5.0

folgende Funktionen aufgerufen werden:

- Installation: siehe Kapitel 3.2.1 und 3.2.2
- Konfiguration: siehe Kapitel 3.2.5
- Starten & Beenden: siehe Kapitel 3.2.3 und 3.2.4
- Helpdesk: siehe Kapitel 3.2.6.1
- Logdatei anzeigen: siehe Kapitel 3.2.6.2
- FirstSpirit-Startseite: siehe Kapitel 3.2.6.3
- Zustandsprotokoll erzeugen: siehe Kapitel 3.2.6.4

3.2.1 Registrieren / Deregistrieren als Systemdienst über Startmenü

Über die Unterpunkte "Dienst registrieren" und "Dienst deregistrieren" im Menü "Installation" können Administratoren den FirstSpirit-Server als Systemdienst konfigurieren. Dazu muss bei der Installation die Komponente "Systemdienste" installiert werden (weiterführende Informationen zur Installation siehe *FirstSpirit Installationsanleitung*). Ohne Systemdienst muss der Server immer erst manuell gestartet werden (siehe Kapitel 3.2.4 Seite 23).

Die Funktion "Dienst registrieren" wird benötigt, wenn der FirstSpirit-Server über den Dienst gestartet bzw. beendet werden soll (siehe Kapitel 3.2.3).



Das Registrieren oder Deregistrieren als Systemdienst über das Startmenü steht nur Administratoren zur Verfügung.



3.2.2 Deinstallieren über das Startmenü

Über den Menüeintrag "Installation" im Startmenü ist auch die Deinstallation von FirstSpirit möglich (weiterführende Informationen zur Deinstallation von FirstSpirit siehe *FirstSpirit Installationsanleitung*).

3.2.3 Starten / Stoppen als Systemdienst über Startmenü

Es besteht die Möglichkeit, den FirstSpirit-Server über das Windows Startmenü (Menü "Starten & Beenden") zu starten.

Administratoren haben die Möglichkeit, den FirstSpirit-Server als Systemdienst zu konfigurieren (siehe Kapitel 3.2.1) und den "Server als Dienst (zu) starten" bzw. den "Server als Dienst (zu) beenden". Ohne registrierten Systemdienst kann der Server im Konsolenmodus gestartet werden (siehe Kapitel 3.2.4 Seite 23).



Das Starten oder Stoppen als Systemdienst über das Startmenü steht nur Administratoren zur Verfügung.



Vor dem Starten und Stoppen des FirstSpirit-Servers sollte der Wartungsmodus aktiviert werden (siehe Kapitel 8.6.2.1 Seite 457).

3.2.4 Starten / Stoppen im Konsolenmodus

Mit einem Klick auf den Eintrag "Server in Konsole starten" im Menü "Starten & Beenden" öffnet sich ein Konsolenfenster, in dem der FirstSpirit-Server gestartet wird.

Für den Start des FirstSpirit-Servers im Konsolenmodus sind keine Administrationsrechte erforderlich.

Der Server kann über das Konsolenfenster mit der Tastenkombination STRG + C beendet werden.





Vor dem Starten und Stoppen des FirstSpirit-Servers sollte der Wartungsmodus aktiviert werden (siehe Kapitel 8.6.2.1 Seite 457).

3.2.5 Konfigurationsdateien über Startmenü editieren

Über das Startmenü, Untermenü "Konfiguration" können einige Konfigurationsdateien des FirstSpirit-Servers angezeigt und bearbeitet werden.



Die Konfiguration dieser Dateien sollte IMMER über das FirstSpirit Server-Monitoring erfolgen. Eine manuelle Konfiguration sollte nur dann vorgenommen werden, wenn die Konfiguration über das Server-Monitoring nicht mehr möglich ist.

Mit einem Klick auf den gewünschten Eintrag

- Java-Wrapper konfigurieren (Kapitel 3.2.5.1 Seite 24)
- Lizenz konfigurieren (Kapitel 3.2.5.2 Seite 25)
- Server konfigurieren (Kapitel 3.2.5.3 Seite 25)

öffnet sich ein Texteditor zum Bearbeiten der Konfigurationsdatei. Bei manuellen Änderungen der Konfigurationsdateien ist ein Neustart des Servers notwendig.

3.2.5.1 Java-Wrapper konfigurieren

Ein Klick auf den Eintrag öffnet die Konfigurationsdatei `fs-wrapper.conf`, die wichtige Konfigurationseinstellungen für den Serverstart und das Java-System des FirstSpirit-Servers enthält. Die Konfigurationsdatei ist zuständig für Start und Stopp des Java-Prozesses und enthält Parameter zur optimalen Ausnutzung des Hauptspeichers des Host-Betriebssystems (siehe Kapitel 4.3.2 Seite 66). Die Datei sollte, wenn möglich, über das FirstSpirit Server-Monitoring konfiguriert werden (siehe Kapitel 8.6.1.5 Seite 454).



3.2.5.2 Lizenz konfigurieren

Mit einem Klick auf den Eintrag öffnet sich die Konfigurationsdatei `fs-license.conf`. Eine Beschreibung der Parameter befindet sich in Kapitel 4.3.5 (Seite 94). Beim Einfügen einer neuen Konfigurationsdatei `fs-license.conf` ist kein Neustart des Servers erforderlich. Die Datei wird automatisch auf dem Server aktualisiert. Die Datei sollte, wenn möglich, über das FirstSpirit Server-Monitoring konfiguriert werden (siehe auch Kapitel 8.6.1.2 Seite 451).



Bei ungültiger Lizenz wird der FirstSpirit-Server nach 30 Minuten heruntergefahren. Ist keine gültige Lizenz installiert, wird vor Ablauf der Zeitspanne eine Nachricht an alle eingeloggten FirstSpirit Benutzer geschickt.



Manipulationen an der `fs-license.conf` führen zu einer ungültigen Lizenz. Sollten Änderungen notwendig werden (z. B. Wechsel der IP-Adresse), wenden Sie sich bitte an <https://helpdesk.e-spirit.de>.

3.2.5.3 Server konfigurieren

Mit einem Klick auf den Eintrag öffnet sich die Konfigurationsdatei `fs-server.conf`, die wichtige Konfigurationseinstellungen für den FirstSpirit-Server enthält. Eine Beschreibung der Parameter befindet sich in Kapitel 4.3.1 (Seite 31). Die Datei sollte, wenn möglich, über das FirstSpirit Server-Monitoring konfiguriert werden (siehe Kapitel 8.6.1.1 Seite 450). Bei einigen Änderungen der Konfigurationsdatei `fs-server.conf` ist ein Neustart des Servers erforderlich.



3.2.6 Weitere Startmenü-Funktionen

Neben der Steuerung und Konfiguration des FirstSpirit-Servers stehen über das Startmenü noch weitere Funktionen zur Verfügung:

3.2.6.1 Helpdesk

Ein Klick auf den Eintrag öffnet ein Browserfenster mit dem Anmeldedialog zum FirstSpirit Trouble Ticket System¹. Die Anmeldung erfolgt (nur nach vorhergehender Registrierung) über die E-Mail-Adresse und ein gültiges Passwort.

3.2.6.2 Protokolldatei anzeigen

Mit einem Klick auf den Eintrag öffnen sich die Log-Dateien:

- fs-wrapper.log: Log-Datei für die Ausgabemeldungen des Java Wrappers (Konfiguration siehe Kapitel 4.3.2.4 Seite 73).
- fs-server.log: Log-Datei für die Ausgabemeldungen des FirstSpirit-Servers (Konfiguration siehe Kapitel 4.3.6 Seite 97).

3.2.6.3 Startseite anzeigen

Mit einem Klick auf den Eintrag öffnet sich die FirstSpirit-Startseite zum Starten der FirstSpirit-Clients (siehe Kapitel 6 Seite 185).

Treten Probleme beim Anzeigen des Anmeldefensters bzw. der FirstSpirit-Startseite auf, sollte zuerst überprüft werden, ob der http-Port auf Serverseite eventuell schon belegt ist. Der gleiche Hinweis gilt für den Start im Socket-Modus. Auch hier sollte zuerst die Portbelegung auf Serverseite geprüft werden (siehe 3.2.5.3 Seite 25).

3.2.6.4 Zustandsprotokoll erzeugen

Mit einem Klick auf den Eintrag kann ein Zustandsprotokoll zur Überwachung des aktuellen Systemzustands des FirstSpirit-Servers erzeugt werden. Diese sogenannten Stackdumps können auch über das FirstSpirit Server-Monitoring erzeugt werden. Das Server-Monitoring bietet außerdem Funktionen zum Analysieren der Stackdumps (siehe Kapitel 8.6.5.5 Seite 470).

¹ <https://helpdesk.e-spirit.de/>



4 Konfiguration des FirstSpirit-Servers

4.1 Dateisystemorganisation

Im Folgenden werden die Verzeichnisstrukturen der Serverinstallation, die Benennung der Konfigurations-Dateien und auch die Struktur der Web-Applikationen erläutert.

4.1.1 Dateinamen

- Konfigurations-Dateien:
 - fs-server.conf
 - fs-database.conf
 - fs-logging.conf
 - fs-license.conf
 - fs-update.conf
 - fs-wrapper.conf
 - fs-webapp.xml
- Programmdateien:
 - fs-server.jar
 - fs-client.jar
 - fs-access.jar
 - fs-or.jar
 - fs-webrt.jar
- Log-Dateien:
 - fs-server.log
 - fs-clients.log
 - fs-wrapper.log
 - fs-gc.*.log
 - fs-webapp-*.log
 - fs-database.log
 - fs-schedule.*.log



4.1.2 Webanwendungen

- Startseite und JavaClient (Dynamische Vorschau): fs5root
- Staging (lokales Vorschau-System, Statische Vorschau): fs5staging
- WebClient und WebClient-Vorschau: fs5webedit
- Server-Monitoring: fs5webmon
- Vorschau (nicht WebClient): fs5preview

Generierte Projekthinhalte werden vom FirstSpirit-Server in den Standard-Generierungsverzeichnissen (fs5staging, fs5preview und fs5webedit) oder in projektlokalen Generierungsverzeichnissen abgelegt. Der Zugriff auf diese Generierungsverzeichnisse ist geschützt, das bedeutet beim Aufrufen dieser Inhalte ist eine Benutzerauthentifizierung erforderlich, sofern dies konfiguriert ist (zur Konfiguration des Anmeldevorgangs siehe Kapitel 4.3.4 Seite 77).

Diese Sicherheitsmechanismen wirken sich auf alle globalen und projektlokalen Staging-Web-Applikationen aus.

4.1.3 Verzeichnisstrukturen

- Systembereiche, die bei der Server-Aktualisierung komplett überschrieben werden:
 - ~fs5\bin Start-Umgebung inkl. benötigter System-Binaries
 - ~fs5\server Java-Umgebung inkl. aller Bibliotheken
 - ~fs5\web FirstSpirit Web-Applikationen
- Konfigurations-Bereiche, die vom Benutzer für die Systemkonfiguration verändert werden können:
 - ~fs5\conf FirstSpirit Konfigurations-Dateien (siehe Kapitel 4.3)
- Datenbereiche:
 - ~fs5\data alle Projektdaten inkl. Module usw.
- Temporäre Dateibereich & Exporte:
 - ~fs5\archive Archiv-Dateien (siehe Kapitel 7.5.10.1)
 - ~fs5\backup Projektexporte für Datensicherung
 - ~fs5\export Projektexporte für interaktive Export/Import-Vorgänge
 - ~fs5\log Log-Dateien
 - ~fs5\work Temporäre FirstSpirit Dateien
 - ~fs5\web\fs5staging\projects Staging mit Web-Apps
 - ~fs5\web\fs5preview\preview_cache Vorschau-Cache
 - ~fs5\web\fs5webedit\preview_cache Vorschau-Cache WebClient



4.2 Allgemeines zur Konfiguration

Die Konfiguration von FirstSpirit erfolgt über Konfigurationsdateien, die sich im Installationsverzeichnis des FirstSpirit-Servers befinden (siehe Kapitel 4.3 Seite 30).

In diesem Kapitel wird der Aufbau der FirstSpirit Konfigurationsdateien erläutert sowie die entsprechenden Parameter beschrieben.

Für das Bearbeiten der Konfigurationsdateien stehen unterschiedliche Möglichkeiten zur Verfügung:

- 1) **über die FirstSpirit Server- und Projektkonfiguration:** Eine komfortable Möglichkeit zum Bearbeiten der Konfigurationseinstellungen für die Konfiguration der Datenbankanbindung (`fs-database.conf`) und die Konfiguration des Logins (`fs-jaas.conf`) bietet die FirstSpirit Server- und Projektkonfiguration (siehe Kapitel 6 Seite 185). Alle Änderungen an diesen beiden Konfigurationsdateien, die über die Server- und Projektkonfiguration vorgenommen werden, werden automatisch in der entsprechenden Konfigurationsdatei gespeichert und auf dem Server aktualisiert.
- 2) **über das FirstSpirit Server-Monitoring:** Weitere Konfigurationseinstellungen können über das FirstSpirit Server-Monitoring oder die JMX-Konsole vorgenommen werden (siehe Kapitel 8 Seite 429 und Kapitel 9 Seite 444). Auch für diese Änderungen wird, analog zur Server- und Projektkonfiguration, automatisch die entsprechende Konfigurationsdatei neu geschrieben und geladen.
- 3) **direkte Änderung der Konfigurationsdateien über das Dateisystem:** Direkte Änderungen der Konfiguration über die Konfigurationsdateien sind nur möglich, wenn der Zugriff über das Dateisystem besteht. Ist ein Zugriff möglich, sollten Änderungen niemals im laufenden Betrieb erfolgen. Es wird empfohlen, Änderungen an den Konfigurationsdateien immer über die von FirstSpirit zur Verfügung gestellten Administrationsumgebungen auszuführen (siehe dazu die jeweiligen Unterkapitel). Zur Realisierung der Änderungen ist teilweise ein Neustart des Servers erforderlich (weitere Informationen siehe Kapitel 4.3.1 ff.).
- 4) **über die JMX-Konsole:** Die Konfigurationseinstellungen aus den Konfigurationsdateien können teilweise auch über die JMX-Konsole angezeigt und geändert werden (Beispiel siehe Kapitel 9.4 Seite 483).



Auf die jeweilige Konfigurationsmöglichkeit innerhalb der Server- und Projektkonfiguration, der JMX-Konsole oder des Server-Monitorings wird in den folgenden Unterkapiteln verwiesen.

Innerhalb der Konfigurationsbeispiele in den folgenden Kapiteln, werden Ausdrücke der Form `key_1 = ${key_2}` zu finden sein. Über den Ausdruck `${ }` wird ein Platzhalter definiert, der den Wert eines anderen Parameters übernehmen kann. Im Beispiel wird dem Parameter `key_1`, der Wert von `key_2` zugewiesen und eventuell durch weitere Angaben ergänzt. Auf diese Weise können beispielsweise längere Pfade aus einzelnen, bereits definierten Werten zusammengesetzt werden (siehe Beispiel in Kapitel 4.3.1.1 Seite 32).



Beim Kopieren von Konfigurationsbeispielen aus den Handbüchern (PDF-Dateien) muss darauf geachtet werden, dass alle Zeilenumbrüche korrekt kopiert werden. Werden die Zeichen beim Kopieren beispielsweise falsch kodiert, kann dies zu Problemen bei der Konfiguration führen.

Generell sollten immer nur die Parameter in den Konfigurationsdateien ergänzt werden, die wirklich geändert werden sollen. Ein einfaches Kopieren von Konfigurationsbeispielen aus den Handbüchern kann zum Überschreiben der Standardwerte führen, was nicht in allen Fällen erwünscht ist.

4.3 Konfigurationsdateien (FirstSpirit-Server)

Im Installationsverzeichnis des FirstSpirit-Servers befinden sich unterschiedliche Konfigurationsdateien. Alle Konfigurationsdateien beginnen mit dem Präfix `fs-` und befinden sich im Konfigurations-Unterverzeichnis `conf`.

Konfigurationsdateien für den FirstSpirit-Server:

- `fs-server.conf` Konfiguration des FirstSpirit-Servers (Kapitel 4.3.1)
- `fs-wrapper.conf` Konfiguration für den Start des FirstSpirit-Servers (Kapitel 4.3.2)
- `fs-database.conf` Konfiguration der Datenbankbindung (Kapitel 4.3.3)
- `fs-jaas.conf` Konfiguration der Authentifizierung (Kapitel 4.3.4)
- `fs-license.conf` Konfiguration der FirstSpirit Lizenz (Kapitel 4.3.5)
- `fs-logging.conf` Konfiguration des Loggings (Kapitel 4.3.6)
- `fs-webapp.xml` Konfigurationseinstellungen Jetty (Kapitel 4.3.7)
- `fs-update.conf` Erforderlich für die Server-Aktualisierung (Kapitel 8.6.2.3),



diese Datei braucht in der Regel nicht bearbeitet werden und darf nicht gelöscht werden.

4.3.1 Konfiguration des FirstSpirit-Servers (fs-server.conf)

Im Unterverzeichnis `conf` des FirstSpirit-Servers befindet sich die Datei `fs-server.conf`, die wichtige Konfigurationseinstellungen für den Server enthält und bei Bedarf angepasst werden muss.

Änderungen an der Konfigurationsdatei `fs-server.conf` können über das FirstSpirit Server-Monitoring vorgenommen werden (siehe Kapitel 8.6.1.1 Seite 450). Die Änderungen werden dann in die Konfigurationsdatei geschrieben und auf dem Server aktualisiert.



Bei einigen Änderungen der Konfigurationsdatei `fs-server.conf` ist ein Neustart des Servers erforderlich (z. B. Änderung des Ports). Direkte Änderungen der Konfigurationsdatei `fs-server.conf` (über die Konfigurationsdatei, nicht über das FirstSpirit Server-Monitoring) erfordern immer einen Server-Neustart!

Die Datei besitzt unter Windows und UNIX denselben Aufbau. Bei den Pfadnamen sollte auf die systemunabhängige Notation mit "/" geachtet werden. Kommentarzeilen können durch # am Zeilenanfang eingeleitet werden, # hinter einem Parameterwert wird nicht als Kommentarzeichen angesehen. Parameterwerte mit Leerzeichen können ohne Änderung, also ohne " oder \, direkt eingegeben werden. Die Konfigurationsdatei `fs-server.conf` ist in funktionsbezogene Bereiche unterteilt. Die einzelnen Bereiche und die zugehörigen Parameter werden nachfolgend erläutert, wobei die Reihenfolge der Einträge variieren kann:

- Communication (Kapitel 4.3.1.1 Seite 32)
- Server (Kapitel 4.3.1.2 Seite 35)
- Server-Monitoring (Kapitel 4.3.1.3 Seite 38)
- Thread Pool (Kapitel 4.3.1.4 Seite 39)
- Thread Queues (Kapitel 4.3.1.5 Seite 40)
- JAAS (Kapitel 4.3.1.6 Seite 42)
- Web Applications (Kapitel 4.3.1.7 Seite 43)
- Preview (Kapitel 4.3.1.8 Seite 47)
- Mail (Kapitel 4.3.1.9 Seite 49)
- LDAP (Kapitel 4.3.1.10 Seite 50)
- Storage Engine Properties (Kapitel 4.3.1.11 Seite 54)
- CacheManager (Kapitel 4.3.1.12 Seite 55)



- Interne Datenbank (Kapitel 4.3.1.13 Seite 56)
- WEBedit configuration (Kapitel 4.3.1.14 Seite 57)
- Miscellaneous (Kapitel 4.3.1.15 Seite 58)
- Webstart configuration (Kapitel 4.3.1.16 Seite 60)
- JumpToServlet (Kapitel 4.3.1.17 Seite 62)
- JMX (Kapitel 4.3.1.18 Seite 63)
- Authentifizierungscookies (Kapitel 4.3.1.19 Seite 65)

4.3.1.1 Bereich: Communication

```
#####  
# communication  
#####  
HTTP_PORT=8000  
SOCKET_PORT=1088  
INTERNAL_SERVLET_ENGINE=1
```

HOST (optional): Falls ein externer Application-Server an Stelle des in FirstSpirit integrierten Web-Servers eingesetzt wird, wird hier der Hostname oder IP-Adresse des FirstSpirit-Servers eingetragen. Servlets auf dem externen Application-Server verbinden zum `SOCKET_PORT` des FirstSpirit-Servers auf dieser Adresse. Dieser Parameter wird automatisch in die Datei `web.xml` der FirstSpirit-Servlets beim Start eingetragen.

HTTP_PORT: Http-Port des FirstSpirit-Servers (wird für die Standardkommunikation zwischen FirstSpirit-Client und -Server benötigt).

URL (optional): Die hier angegebene URL zur Startseite des FirstSpirit-Servers wird bei automatisch gesendeten E-Mails eingetragen, die eine URL zum Start des Clients enthalten. Solche E-Mails werden zum Beispiel bei Zustandsänderungen innerhalb definierter Arbeitsabläufe gesendet. Die hier angegebene URL wird ebenso bei Verwendung der Funktion "FirstSpirit Adresse kopieren" im JavaClient (Menü "Extras") berücksichtigt.

Normalerweise wird die URL automatisch ermittelt, was aber nicht funktioniert, falls der Server unter mehreren Hostnamen bekannt ist.

```
URL=http://fs5server.domain.net
```

Dieser Parameter muss immer gesetzt werden, wenn Arbeitsablauf-E-Mails und ein externer Web-Applikations-Server (z. B. Tomcat) verwendet werden sollen.



Folgende Parameter müssen in diesem Fall ebenfalls gesetzt werden. Die Werte werden an die automatisch ermittelten bzw. durch den Parameter `URL` definierte URL angehängt und für den Verbindungsaufbau berücksichtigt.

`fs.url.hostname` (optional): Über diesen Parameter kann ein Hostname für die Client-Verbindung angegeben werden.

`fs.url.socketport` (optional): Über diesen Parameter kann der Port angegeben werden, wenn die Client-Verbindung im Socket-Modus hergestellt werden soll (Platzhalter `%FIRSTspiritSOCKETURL%` in Arbeitsablauf-E-Mails).

`fs.url.httpport` (optional): Über diesen Parameter kann der Port angegeben werden, wenn die Client-Verbindung im HTTP-Modus hergestellt werden soll (Platzhalter `%FIRSTspiritURL%` in Arbeitsablauf-E-Mails).

`fs.url.usehttps` (optional): Soll beim Aufruf der Url HTTPS berücksichtigt werden, muss dieser Parameter auf `true` gesetzt werden. Der Standardwert ist `false`

Beispiel:

Aus folgender Konfiguration in der `fs-server.conf`

```
HTTP_PORT=5000
SOCKET_PORT=5100
URL=http://myServer:8000
fs.url.hostname=aliashost
fs.url.socketport=8300
fs.url.httpport=8200
fs.url.usehttps=true
```

würde für den Platzhalter `%FIRSTspiritURL%`:

```
http://myServer:8000/start/FIRSTspirit.jnlp?app=client&project=myProject&name=null&type=Page&id=443977&host=aliashost&port=8200&mode=HTTP&usehttps=true
```

für den Platzhalter `%FIRSTspiritSOCKETURL%`:

```
http://myServer:8000/start/FIRSTspirit.jnlp?app=client&project=myProject&name=null&type=Page&id=443977&host=aliashost&port=8300&mode=SOCKET&usehttps=true
```

Bei Verwendung in einem bereits gestarteten Projekt z. B. über die



Funktion "Zu FirstSpirit Adresse wechseln" im JavaClient haben diese Parameter in der Regel keine Auswirkung.



*Sind die Verbindungseinstellungen auf der Startseite aktiviert (siehe Kapitel 6.3.4.1 Seite 192), werden gemäß der in Kapitel 7.3.8 Seite 250 beschriebenen Auswertungsreihenfolge (Punkt **Auswertungsreihenfolge**) die `fs.url`-Parameter nicht berücksichtigt. Entgegen dieser Auswertungsreihenfolge werden allerdings die entsprechenden Parameter, die in den Servereigenschaften in den Bereichen "Webstart" und "Startseite" definiert sind, bei deaktivierten Verbindungseinstellungen nicht berücksichtigt, sondern es kommen die `fs.url`-Parameter zum Tragen.*

`SOCKET_PORT`: TCP-Port, auf dem der FirstSpirit-Server auf Verbindungen für das FirstSpirit-eigene Socket-Protokoll wartet. Wird für die interne Kommunikation zwischen Servlets und FirstSpirit-Server verwendet und, falls konfiguriert, für die Kommunikation mit dem FirstSpirit-JavaClient. Dieser Parameter wird außerdem automatisch in die Datei `web.xml` der FirstSpirit-Servlets beim Start eingetragen.

`INTERNAL_SERVLET_ENGINE`: Wert 1= interner Webserver und Servlet Engine (Jetty) wird gestartet (Standardwert ist 1)
Wert 0 = interner Webserver und Servlet Engine (Jetty) wird nicht gestartet, falls z. B. ein externer Application-Server eingesetzt werden soll.

`SOCKET_HOST` (optional): Hostname für die Bindeadresse des `SOCKET_PORT`, um den Server, wenn notwendig, auf eine IP-Adresse zu beschränken. Wird kein Wert übergeben, bindet der Server an alle IP-Adresse des Hosts.



Der Parameter `SOCKET_HOST` (= das Interface, auf den sich der Socket-Listener binden soll) kann nur verwendet werden, wenn auch der Parameter `HOST` (= Hostname, über den der Server von außen erreichbar ist) korrekt konfiguriert wurde, d.h. auf dasselbe Netzwerk-Interface des Servers abgebildet wird.

`SYMBOLIC_HOSTNAME` (optional): Symbolischer Hostname des FirstSpirit-Servers. Dieser Hostname dient nur zur Anzeige auf der Startseite und hat ansonsten keine weitere Funktion.



`ALLOWED_ENCRYPTIONS` (optional): Über diesen Parameter kann dem Benutzer vorgegeben werden, welche Verschlüsselung er in seinen Verbindungseinstellungen (siehe Kapitel 6.3.4.1 Seite 192) verwenden muss. Für die Verwendung von TLS wird der Wert 1 gesetzt, für die Verwendung von DH_ARC4 der Wert 2. Soll keine Verschlüsselung verwendet werden, muss der Wert 0 gesetzt werden. Auch beliebige Kombinationen der Parameter 0, 1 und 2 sind möglich. Beispiel:

```
ALLOWED_ENCRYPTIONS=1,2
```

Stimmt die Verschlüsselung, die der Benutzer eingestellt hat, nicht mit der hier vorgegebenen Verschlüsselung überein, erhält der Benutzer bei der Anmeldung am JavaClient oder zur Anwendung der Server- und Projektkonfiguration eine Fehlermeldung, dass die Verbindungsparameter falsch gesetzt wurden; es ist keine Kommunikation zwischen Client und Server möglich.



Für den Parameter `ALLOWED_ENCRYPTIONS` kann keine TLS-Verschlüsselung (Parameterwert 1) verwendet werden, wenn Websphere als Application-Server für die Webanwendung `fs5root` eingesetzt wird. In diesem Fall bietet sich an, RC4-Verschlüsselung (Parameterwert 2) für SOCKET oder HTTP einzusetzen oder HTTPS zu verwenden. Bei HTTPS ist keine zusätzliche Verschlüsselung notwendig, so dass dann der Parameterwert 0 einzusetzen ist.

4.3.1.2 Bereich: Server

```
#####  
# server  
#####  
backup_files=50  
cyclicReferenceSaveTime=60
```

`backup_files`: Anzahl der Backup-Versionen, die zu den internen Strukturdateien gespeichert werden. Dies betrifft ausschließlich die Verwaltungs- und Strukturdaten und nicht die Inhaltsdaten. Alle Inhaltsdaten werden separat versioniert. Eine Erhöhung des Wertes verbessert die Möglichkeiten zur Fehlerkorrektur, führt aber auch zu einer erhöhten Belegung des Plattenspeichers auf dem Rechner (Standardwert ist 50).

`cyclicSaveTime`: Gibt die Zeit in Sekunden an, nach der Änderungen, beispielsweise an der Historie, der Statistik, den Aufgaben oder an



den Benutzereinstellungen gespeichert werden. Das zyklische Speichern wird nur dann ausgeführt, wenn die entsprechenden Daten geändert wurden (Standardwert ist 60 sec.).

JNLP_SERVLET_URL: Der hier definierte Wert wird für die Erzeugung der FirstSpirit-URLs benötigt, die als Verweise zum Starten des Clients innerhalb von E-Mails generiert werden. Beispiel:

```
JNLP_SERVLET_URL=${URL}/start/FIRSTspirit.jsp
```

JNLP_REDIRECT: Dieser Parameter kann auf den Wert 1 gesetzt werden, wenn eine Authentifizierung über Kerberos (siehe Kapitel 4.3.4.5 Seite 83) in Verbindung mit Java Web Start (JNLP) und Microsoft Internet Explorer nicht funktioniert. (Standardwert ist 0.)

DTO_LRU_SIZE: Der Parameter bestimmt die Größe des DTO-Caches für ein Projekt auf dem Server. Hier werden die zuletzt genutzten Baumobjekte eines Projekts gehalten. Der Wert definiert die Anzahl der Objekte die im Cache verwaltet werden (Standardwert 512 Store-Elemente).

CLIENTAPP_PATH: Der Parameter wird ausschließlich für den Ausrollprozess nativer Clientapplikationen benötigt (siehe Kapitel 4.9.1 Seite 175). Standardwert: `FirstSpirit5\data/clientapp`.

CLIENT_HOME_DIR: Pfadangabe zum Verzeichnis im Dateisystem, in das Client-Applikationen abgelegt werden sollen (siehe Kapitel 4.9.2 Seite 176). Dabei können absolute (z. B. `CLIENT_HOME_DIR=C:/test`) oder relative Pfadangaben verwendet werden. Für relative Pfadangaben kann das Zeichen `~` zu Beginn der Pfadangabe als Platzhalter verwendet werden, z. B. `~/myclientapps`. `~` wird dann durch das betriebssystem-spezifische, aktuelle User-Home-Verzeichnis ersetzt. Wird der Parameter nicht angegeben, werden Client-Applikationen standardmäßig in das Verzeichnis `\.firstspirit` im betriebssystemspezifischen User-Home-Verzeichnis ausgerollt. Dabei wird für jede FirstSpirit-Version ein eigenes Verzeichnis verwendet. Der Verzeichnisname enthält dabei die jeweilige FirstSpirit-Major-, -Minor- und -Release-Versionsnummer, z. B. `\.firstspirit_5.0R0`





Diese Verzeichnisse werden bei einem Versionswechsel nicht vom System gelöscht und müssen bei Bedarf (z. B. um Speicherplatz freizugeben) manuell entfernt werden.

Standardwert:

```
CLIENT_HOME_DIR=~/.firstspirit_${FS_MAJOR}.${FS_MINOR}${FS_RELEASE}
```

CLIENT_HOME_DIR_WINDOWS: Pfadangabe zum Verzeichnis im Dateisystem, in das Client-Applikationen auf Arbeitsplatzrechnern mit Windows-Betriebssystem ausgerollt werden sollen. Beispiel:

```
CLIENT_HOME_DIR_WINDOWS=C:/test
```

Wird nur **CLIENT_HOME_DIR_WINDOWS** angegeben, wird die Client-Applikation in das betriebssystemspezifische User-Home-Verzeichnis ausgerollt.

CLIENT_HOME_DIR_LINUX: Pfadangabe zum Verzeichnis im Dateisystem, in das Client-Applikationen auf Arbeitsplatzrechnern mit Linux-Betriebssystem ausgerollt werden sollen (vgl. Parameter **CLIENT_HOME_DIR_WINDOWS**).

CLIENT_HOME_DIR_MAC: Pfadangabe zum Verzeichnis im Dateisystem, in das Client-Applikationen auf Arbeitsplatzrechnern mit Macintosh-Betriebssystem ausgerollt werden sollen (vgl. Parameter **CLIENT_HOME_DIR_WINDOWS**).

CLIENT_HOME_DIR_AIX: Pfadangabe zum Verzeichnis im Dateisystem, in das Client-Applikationen auf Arbeitsplatzrechnern mit AIX-Betriebssystem ausgerollt werden soll (vgl. Parameter **CLIENT_HOME_DIR_WINDOWS**).

CLIENT_HOME_DIR_SOLARIS: Pfadangabe zum Verzeichnis im Dateisystem, in das Client-Applikationen auf Arbeitsplatzrechnern mit Solaris-Betriebssystem ausgerollt werden sollen (vgl. Parameter **CLIENT_HOME_DIR_WINDOWS**).

workflow.task.cache:

workflow.model.cache: Der hier definierte Typ gibt an, wie lange eine Aufgabe bzw. ein Arbeitsablaufmodell im Cache erhalten bleiben soll. 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. (Standardwerte: `workflow.task.cache=SOFT_1024` und `workflow.model.cache=SOFT_128`).

`webserver.conf-migration`: Dieser Parameter wird automatisch gesetzt. Er wird vom FirstSpirit-Server verwaltet und darf nicht geändert oder gelöscht werden.

4.3.1.3 Bereich Server Monitoring

```
#####  
# Server Monitoring - Ajax  
#####  
AJAX_DATA_SYNC_TIMEOUT=20  
AJAX_IMAGE_RELOAD_TIMEOUT=60  
  
# ;-seperated mail-addresses for mail-system-monitoring-site  
SYSTEM_MONITORING_MAIL_RECIPIENTS=admin@yourdomain.net
```

`AJAX_DATA_SYNC_TIMEOUT`: Definiert das Zeitintervall in Sekunden, in der die dynamisch aktualisierten Daten (per Ajax) innerhalb des Server-Monitorings erneuert werden (Standardwert ist 20).

`AJAX_IMAGE_RELOAD_TIMEOUT`: Definiert das Zeitintervall in Sekunden, in der die dynamisch aktualisierten Statistiken (Grafik) (per Ajax) innerhalb des Server-Monitorings erneuert werden (Standardwert ist 60).

`SYSTEM_MONITORING_MAIL_RECIPIENTS`: Mail-Adressen von Benutzern, die über den Systemstatus des FirstSpirit-Servers informiert werden sollen. Als Trennzeichen zwischen den Adressen wird ";" verwendet.



4.3.1.4 Bereich Thread Pool

```
#####  
# 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=
```

Der FirstSpirit-Server wird 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.

`ThreadPool.minSize`: Definiert die Minimalgröße des limitierten ThreadPools (vgl. Abbildung 9-6 Seite 489).

`ThreadPool.maxSize`: Definiert die Maximalgröße des limitierten ThreadPools und damit die größtmögliche Anzahl parallel auszuführender Tasks (vgl. Abbildung 9-6 Seite 489). Je mehr Prozessoren ein Server besitzt, desto höher kann auch der Wert für `ThreadPool.maxSize` konfiguriert werden.

Der Wert der tatsächlich ausgeführten Threads kann höher liegen, da für die hochpriorisierten Tasks diese Einschränkung nicht gültig ist (unlimitierter Thread-Pool).

Werden für die Parameter

- `ThreadPool.minSize`
- `ThreadPool.maxSize`

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



4.3.1.5 Bereich Thread Queues

```
#####
# 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).
#####
ThreadQueue.LOW.maxRunning=2
ThreadQueue.LOW.queueCapacity=128

# 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=

# 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=
```

Die Ausführung von Tasks erfolgt über den ExecutionManager, der eine Vielzahl von unterschiedlich klassifizierten Queues verwaltet (siehe Kapitel 9.8 Seite 487). Einige der Queues können über Parameter die Anzahl der aktiven Tasks einschränken. Für Queues mit der Klassifizierung `BOUNDED` kann darüber hinaus die Aufnahmekapazität der Queue eingeschränkt werden.

`ThreadQueue.LOW.maxRunning`: Von den Ressource-intensiven Tasks innerhalb dieser Queue sollte jeweils nur eine geringe Anzahl parallel ausgeführt werden. Die Anzahl der Tasks, die gleichzeitig ausgeführt werden dürfen, kann über den Parameter `maxRunning` konfiguriert werden (Standardwert ist 2 Threads).

`ThreadQueue.DEFAULT.maxRunning`: Die Anzahl der parallel ausführbaren Tasks dieser Queue können ebenfalls über den Parameter `maxRunning` eingeschränkt werden. Da diese Tasks nicht so ressourcenintensiv sind, kann der Wert hier höher liegen, als der in `ThreadQueue.LOW.maxRunning` definierte (Standardwert ist 25 Threads).

`ThreadQueue.DEFAULT.queueCapacity`: Die Aufnahmekapazität einer mit `DEFAULT` klassifizierten Queue kann eingeschränkt werden.

`ThreadQueue.BOUNDED.maxRunning`: Die mit `BOUNDED` klassifizierte Queue kann



über zwei Parameter konfiguriert werden. Über den Parameter `maxRunning` kann die Anzahl der aktiven Tasks begrenzt werden (Standardwert ist 25 Threads).

`ThreadQueue.BOUNDED.queueCapacity`: Außerdem kann die Aufnahmekapazität einer mit `BOUNDED` klassifizierte Queue über den Parameter `queueCapacity` eingeschränkt werden. Ist der unter `queueCapacity` konfigurierte Wert erreicht, werden weitere Tasks vom Server zurückgewiesen. Das bedeutet, bei hoher Belastung des Servers wird nur eine bestimmte Anzahl von Tasks in die Queue gestellt (Standardwert ist 50 Tasks) (wovon die unter `maxRunning` definierte Anzahl direkt abgearbeitet wird), alle weiteren Tasks werden vorerst zurückgewiesen und an eine interne Rejection-Strategie überstellt.

Werden für die Parameter

- `ThreadQueue.DEFAULT.maxRunning`
- `ThreadQueue.DEFAULT.queueCapacity`
- `ThreadQueue.BOUNDED.maxRunning`
- `ThreadQueue.BOUNDED.queueCapacity`

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` (siehe Kapitel 4.3.1.4 Seite 39).



4.3.1.6 Bereich: JAAS – Konfiguration des Anmeldevorgangs

```
#####  
# JAAS  
#####  
JAAS=${cmsroot}/conf/fs-jaas.conf  
JAAS.default=plain  
JAAS.client=sso  
JAAS.admin=plain  
JAAS.preview=system  
JAAS.system=system  
JAAS.webedit=websso  
JAAS.webmonitor=websso
```

Über diesen Bereich können wichtige Konfigurationseinstellungen für den Anmeldevorgang für unterschiedliche FirstSpirit Anwendungen (z. B. Anmeldung am JavaClient oder Anforderung einer Vorschau) definiert werden. Dafür werden eine Reihe unterschiedlicher Authentifizierungs-Module zur Verfügung gestellt (z. B. SSO). Im Bereich JAAS² wird ein Mapping der Anwendungsmöglichkeit zu den Verbindungsmodulen hergestellt, die in der Konfigurationsdatei fs-jaas.conf konfiguriert wurden (siehe Kapitel 4.3.4 Seite 77). Eine komfortable Möglichkeit zur Konfiguration dieses Bereichs befindet sich in der Server- und Projektkonfiguration (siehe Kapitel 7.3.10 Seite 253) und über das Server-Monitoring (siehe Kapitel 8.6.1.8 Seite 456).

² Java Authentication and Authorization Service
(Informationen: <http://www.oracle.com/technetwork/java/javase/jaas/index.html>)



4.3.1.7 Bereich: Web Applications

```
#####
# web applications
# - ROOT: start page with login page, etc.
# - WEBEDIT: WEBedit
# - WEBMON: WEBmonitor
# - STAGING: project generation web applications
#####

# WEBAPP_xyz_URL: application url (must be start with a slash
# '/'); used for jumps between different applications

WEBAPP_ROOT_URL=/
WEBAPP_WEBEDIT5_URL=/${WEBAPP_WEBEDIT5_NAME}
WEBAPP_WEBMON_URL=/${WEBAPP_WEBMON_NAME}
WEBAPP_STAGING_URL=/${WEBAPP_STAGING_NAME}
WEBAPP_PREVIEW_URL=/${WEBAPP_PREVIEW_NAME}

# root directory of all web applications
WEB_DIR=${cmsroot}/web

# WEBAPP_xyz_PATH: directory of this web application
WEBAPP_ROOT_PATH=${WEB_DIR}/${WEBAPP_ROOT_NAME}
WEBAPP_WEBEDIT5_PATH=${WEB_DIR}/${WEBAPP_WEBEDIT5_NAME}
WEBAPP_WEBMON_PATH=${WEB_DIR}/${WEBAPP_WEBMON_NAME}
WEBAPP_STAGING_PATH=${WEB_DIR}/${WEBAPP_STAGING_NAME}
WEBAPP_PREVIEW_PATH=${WEB_DIR}/${WEBAPP_PREVIEW_NAME}

# title parameter for fs5root site
WEBAPP_ROOT_TITLE=FirstSpirit ${HOSTNAME}:${HTTP_PORT}

# where is the clientjar?
CLIENTJAR_URL=${WEBAPP_ROOT_PATH}/clientjar/fs-client.jar
```

GLOBAL_WEBAPPS: Dieser Parameter wird automatisch gesetzt und enthält die IDs der auf dem Server verfügbaren globalen Web-Applikationen (siehe Kapitel 7.3.13.1 Seite 270). Der Parameter wird vom FirstSpirit-Server verwaltet und darf nicht geändert oder gelöscht werden. Weitere Konfigurationseinstellungen zu den globalen Web-Applikationen werden mithilfe der Parameter

```
WEBAPP_*_NAME,
WEBAPP_*_CONTEXT_NAME,
WEBAPP_*_SELECTED_WEBSERVER,
WEBAPP_*_CONFIG_CHANGED          und
WEBAPP_*_ACTIVE_WEBSERVER
```



gespeichert, wobei * die ID der jeweiligen globalen Web-Applikation ist. Auch diese Parameter werden von FirstSpirit verwaltet und dürfen nicht geändert oder gelöscht werden.

WEBAPP_LOG_LEVEL: Mit diesem Parameter kann das Level für das Logging von Webanwendungen bestimmt werden. Dazu können folgende Werte angegeben werden: TRACE, DEBUG, INFO, WARN, ERROR. Nur Einträge mit einem Level gleich oder höher dem angegebenen werden in die Logdatei geschrieben. (Siehe auch Kapitel 4.6.2 Seite 134; Standardwert: WARN.)

WEBAPP_ROOT_URL: URL zur Webanwendung Startseite (fs5root). Der hier hinterlegte Wert wird für das Mapping innerhalb der internen Servlet Engine benötigt (siehe auch Kapitel 4.3.7.2 Seite 102). Der Wert muss immer mit einem "/" beginnen, dann folgt der symbolische Name der jeweiligen Webanwendung. Der URL wird für Verweise zu unterschiedlichen FirstSpirit-Anwendungen benötigt. (Standardwert: /).

Beim Betrieb unter Websphere ist folgender Eintrag notwendig:

```
WEBAPP_ROOT_URL=/fs5root
```

(siehe auch Kapitel 4.6.1 Seite 132.)

WEBAPP_WEBEDIT5_URL: URL zur Webanwendung WebClient (fs5webedit) innerhalb der Startseite. Der hier hinterlegte Wert wird für das Mapping innerhalb der internen Servlet Engine benötigt (siehe auch Kapitel 4.3.7.2 Seite 102). Der Wert muss immer mit einem "/" beginnen, dann folgt der symbolische Name der jeweiligen Webanwendung. Der URL wird für Verweise zu unterschiedlichen FirstSpirit-Anwendungen benötigt (Standardwert: /\${WEBAPP_WEBEDIT5_NAME}).

WEBAPP_WEBMON_URL: URL zur Webanwendung "Server-Monitoring" (fs5webmon) innerhalb der Startseite. Der hier hinterlegte Wert wird für das Mapping innerhalb der internen Servlet Engine benötigt (siehe auch Kapitel 4.3.7.2 Seite 102). Der Wert muss immer mit einem "/" beginnen, dann folgt der symbolische Name der jeweiligen Webanwendung. Der URL wird für Verweise zu unterschiedlichen FirstSpirit-Anwendungen benötigt (Standardwert:

```
/${WEBAPP_WEBMON_NAME}).
```

WEBAPP_STAGING_URL: URL zur Webanwendung FirstSpirit Staging (fs5staging). Der hier hinterlegte Wert wird für das Mapping innerhalb der internen Servlet Engine benötigt (siehe auch Kapitel 4.3.7.2 Seite 102). Der



Wert muss immer mit einem "/" beginnen, dann folgt der symbolische Name der jeweiligen Webanwendung. Der URL wird für Verweise zu unterschiedlichen FirstSpirit-Anwendungen benötigt (Standardwert: `/${WEBAPP_STAGING_NAME}`).

WEBAPP_PREVIEW_URL: URL zur Webanwendung FirstSpirit Preview (fs5preview). Der hier hinterlegte Wert wird für das Mapping innerhalb der internen Servlet Engine benötigt (siehe auch Kapitel 4.3.7.2 Seite 102). Der Wert muss immer mit einem "/" beginnen, dann folgt der symbolische Name der jeweiligen Webanwendung. Der URL wird für Verweise zu unterschiedlichen FirstSpirit-Anwendungen benötigt (Standardwert: `/${WEBAPP_PREVIEW_NAME}`).

WEB_DIR: Pfadangabe zum FirstSpirit Web-Verzeichnis für die Nutzung aller FirstSpirit-Webanwendungen (fs5root, fs5preview, fs5staging, fs5webedit, fs5webmon) (zur Notation `${}` siehe Kapitel 4.2 Seite 29).

WEBAPP_ROOT_PATH: Pfadangabe zum Verzeichnis im Dateisystem in dem der FirstSpirit-Server die Webanwendung FirstSpirit Startseite (fs5root) ablegt (zur Notation `${}` siehe Kapitel 4.2 Seite 29).

WEBAPP_WEBEDIT5_PATH: Pfadangabe zum Verzeichnis im Dateisystem in dem der FirstSpirit-Server die Webanwendung WebClient (fs5webedit) ablegt (zur Notation `${}` siehe Kapitel 4.2 Seite 29).

WEBAPP_WEBMON_PATH: Pfadangabe zum Verzeichnis im Dateisystem in dem der FirstSpirit-Server die Webanwendung Server Monitoring (fs5webmon) ablegt (zur Notation `${}` siehe Kapitel 4.2 Seite 29).

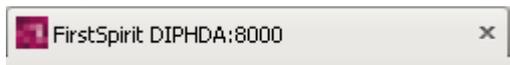
WEBAPP_STAGING_PATH: Pfadangabe zum Verzeichnis im Dateisystem in dem der FirstSpirit-Server die Webanwendung FirstSpirit Staging (fs5staging) ablegt. In diesem Verzeichnis legt der FirstSpirit-Server auch alle generierten Projektdateien ab (zur Notation `${}` siehe Kapitel 4.2 Seite 29).

WEBAPP_PREVIEW_PATH: Pfadangabe zum Verzeichnis im Dateisystem in dem der FirstSpirit-Server die Webanwendung Preview (fs5preview) ablegt. In diesem Verzeichnis legt der der FirstSpirit-Server auch die generierten Vorschauseiten ab.

WEBAPP_ROOT_TITLE: Über diesen Parameter kann der Titel der FirstSpirit-Startseite geändert werden. Standardwert ist die Anzeige `FirstSpirit ${HOSTNAME}:${HTTP_PORT}`. Damit stellt sich die Startseite des



FirstSpirit-Servers im Browser beispielsweise folgendermaßen dar:



Soll der über den Parameter `SYMBOLIC_HOSTNAME` definierte symbolische Hostname des FirstSpirit-Servers verwendet werden (siehe Kapitel 4.3.1.1 Seite 32), wenn `WEBAPP_ROOT_TITLE` nicht angegeben ist, kann z. B.

```
WEBAPP_ROOT_TITLE=FirstSpirit ${SYMBOLIC_HOSTNAME}
```

gesetzt werden. Es findet keine automatische Ersetzung statt, da es sich bei `SYMBOLIC_HOSTNAME` um einen optionalen Parameter handelt, der auch leer sein kann.

`CLIENTJAR_URL`: URL des FirstSpirit-Client Jar-Files.



4.3.1.8 Bereich: Preview

```
#####  
# preview  
#####  
preview.cacheMediaFiles=  
preview.internalDelivery=html,htm,txt,xml,pdf,jsp,shtml,ini  
preview.cacheTimeout=1200
```

`preview.cacheMediaFiles`: Standardmäßig ist der Wert `false` gesetzt. Der Standardwert muss nur geändert werden, wenn eine externe Servlet-Engine eingesetzt wird. Ist der Wert auf `true` gesetzt, werden zusätzlich zu den Vorschauseiten auch Medien-Dateien in der internen Servlet-Engine zwischengespeichert.

`preview.internalDelivery`: Hier kann eine kommaseparierte Liste von Datei-Endungen angegeben werden, die direkt von der Servlet-Engine ausgeliefert werden sollen. Dateien, die nicht in der Liste auftauchen, werden über einen internen Redirect an den Webserver ausgeliefert. (Dies kann z. B. ein Apache-Webserver sein, der dann das Processing der Datei übernimmt, z. B. PHP.)

Standardwert: `*`. Dieser wird verwendet, um alle Dateien mit einer Endung, die nicht in der Liste des Parameters `preview.externalDelivery` enthalten ist, von der Servlet-Engine ausliefern zu lassen.

`preview.externalDelivery`: Über diesen Parameter kann eine kommaseparierte Liste von Dateien-Endungen angegeben werden, die von einem externen Webserver und **nicht** von der Servlet-Engine ausgeliefert werden sollen. Dieser Parameter wird nur berücksichtigt, wenn der Parameter `preview.internalDelivery` auf `*` gesetzt ist. Standardmäßig ist der Parameter leer und muss bei Bedarf gesetzt werden.

`preview.externalDeliveryURL`: Über diesen Parameter kann die URL zum externen Webserver angegeben werden, der für Dateitypen verwendet wird, die nicht im Parameter `preview.internalDelivery` definiert wurden (z. B. PHP oder ASP). Die URL setzt sich aus dem Server-Namen und dem Port zusammen, der in Kapitel 4.5.1 im Abschnitt "Virtueller Webserver" (Seite 110) konfiguriert wird. Beispiel:

```
preview.externalDeliveryURL=http://fs5.yourdomain.net:80.
```

Dieser Parameter ist standardmäßig leer und muss bei Bedarf gesetzt



werden.

`preview.cacheTimeout`: Die generierten Seiten, die im `cacheDir` zwischengespeichert werden, sind nur für das hier definierte Zeitintervall (in Sekunden) gültig. Ist das Intervall abgelaufen, werden die Vorschauseiten (beim Anfordern) neu erzeugt und erneut zwischengespeichert. Dabei werden alle Dateien, die älter sind als das eingestellte Zeitintervall, aus dem Cache entfernt. Der Standardwert ist 1200 Sekunden, der minimale Wert 60.

`preview.cacheFileWithTimestamp`: Über diesen Parameter können Dateien einen Zeitstempel im Dateinamen erhalten. Dies ist z. B. wichtig bei der Verwendung von IBM Websphere, um JSP-Dateien korrekt neu kompilieren zu können, wenn der Inhalt der Datei nach der vorherigen Kompilation geändert wurde. Dazu wird für `preview.cacheFileWithTimestamp` eine kommaseparierte Liste mit Datei-Endungen angegeben, die den Zeitstempel im Dateinamen erhalten sollen, z. B.

`preview.cacheFileWithTimestamp=jsp,jsf`. Um alle Dateinamen mit dem Zeitstempel zu ergänzen, kann der Parameter auf `*` gesetzt werden. Standardmäßig ist der Parameter leer. Bei Verwendung von IBM WebSphere muss der Parameter gesetzt werden.

`preview.enforceDelivery`: In der Regel werden Daten, die sich noch im Cache eines Webbrowsers befinden und sich seit der letzten Abfrage nicht geändert haben, nicht erneut vom Server gesendet. Stattdessen übermittelt er den HTTP-Statuscode 304 ("Not Modified"). Über den Parameter `preview.enforceDelivery` werden Dateitypen angegeben, für die nie mit dem Statuscode 304 geantwortet werden soll. Auf diese Weise kann für diese Dateitypen eine Auslieferung erzwungen werden. Für den Parameter können auch mehrere Dateierweiterungen, durch Komma separiert, angegeben werden. Standardeinstellung:

```
preview.enforceDelivery=asp,aspx,dhtml,jsp,jsf,php
```



4.3.1.9 Bereich: Mail

```
#####  
# mail  
#####  
#mail.default-recipient=  
#mail.sender=  
#mail.smtp=  
  
LICENSE_EXPIRATION_WARNING_DAYS=30  
LICENSE_EXPIRATION_MAIL_ADDRESS=info@e-spirit.de
```

`mail.smtp`: Spezifiziert einen SMTP-Mailserver. Dieser wird benötigt, da der Server nach der Auslieferung oder Generierung einer Website auf Wunsch eine E-Mail versendet, die über das Resultat berichtet.

`mail.smtp.port`: Mithilfe dieses Parameters kann angegeben werden, über welchen Port der SMTP-Server E-Mails versenden soll. Standardmäßig wird Port 25 verwendet.

`mail.default-recipient`: Angabe einer Standard-E-Mail-Adresse.

`mail.sender`: Angabe einer E-Mail-Adresse, die als Absender für alle Mails des FirstSpirit-Servers verwendet wird. Ist hier keine E-Mail-Adresse angegeben, wird eine E-Mail-Adresse mit folgendem Schema verwendet:

```
cmsserver@[Hostname]
```

wobei der Hostname über die Methode

```
InetAddress.getLocalHost().getHostName();
```

ermittelt wird.

`mail.subject.charset`: Mithilfe dieses Parameters kann das Encoding für die Betreffzeile in von FirstSpirit versendeten E-Mails eingestellt werden. Standardmäßig wird ISO-8859-1 verwendet.

`mail.mime.charset`: Mithilfe dieses Parameters kann das Encoding für den Body in von FirstSpirit versendeten E-Mails eingestellt werden. Standardmäßig wird das Encoding des Betriebssystems verwendet, das auf dem FirstSpirit-Server läuft.



4.3.1.10 Bereich: LDAP

Im FirstSpirit-Server lassen sich verschiedene LDAP-Konfigurationen (sog. Sections) anlegen. Der Name einer Section wird definiert über den Eintrag in der Konfigurationsdatei `fs-jaas.conf` (siehe Kapitel 4.3.4 Seite 77). Die Konfiguration einer Section erfolgt in der Konfigurationsdatei `fs-server.conf` (Bereich LDAP). Vor jedem Konfigurations-Parameter wird der Section-Name (aus der `fs-jaas.conf`) angegeben, der der Form "LDAP_n" entsprechen muss, wobei n die Section-Nummer ist, die fortlaufend bei 1 beginnend durchnummeriert wird. Falls nur 1 LDAP-Section verwendet wird, kann die Section auch "LDAP" genannt werden.

```
LDAP_n.parameter=wert
```

Die Namen der LDAP-Attribute müssen in derselben Groß-/Kleinschreibung eingetragen werden, wie sie im LDAP-Verzeichnis vorhanden sind!



Zum Auskommentieren der folgenden Parameter muss ein Leerstring als Wert angegeben. Ein Auskommentieren dieser Parameter durch # hat keine Auswirkung, da in diesem Fall der Standardwert verwendet wird.

```
LDAP.IMPORT_USER.LOGIN_ATTRIBUTE=  
LDAP.IMPORT_USER.NAME_ATTRIBUTE=  
LDAP.IMPORT_USER.EMAIL_ATTRIBUTE=  
LDAP.IMPORT_USER.PHONE_ATTRIBUTE=  
LDAP.IMPORT_USER.ABBREVIATION_ATTRIBUTE=
```

Im nachfolgenden Konfigurationsbeispiel zur Anbindung an den LDAP-Server des Microsoft Active Directory (unter Verwendung von `LDAP.AUTHENTICATION=SEARCH_BIND`) wurde als Section-Name "LDAP" gewählt. Abhängig von der Konfiguration der `fs-jaas.conf` (vgl. Kapitel 4.3.4) können auch andere Section-Namen gewählt werden, und es können auch mehrere unterschiedliche LDAP-Sections gleichzeitig definiert sein.

```
LDAP.NAME=e-spirit.de  
LDAP.HOST_URL=ldap://server1 ldap://server2 ldap://server3  
LDAP.SSL=FALSE  
LDAP.AUTHENTICATION=SEARCH_BIND  
LDAP.SEARCH.BIND_DN=cn=ldapuser,cn=users,dc=e-spirit,dc=de  
LDAP.SEARCH.BIND_PASSWORD=ldappassword  
LDAP.SEARCH.BASE_DN=ou=mitarbeiter,ou=Dortmund,dc=e-spirit,dc=de  
LDAP.SEARCH.FILTER=(sAMAccountName=$USER_LOGIN$)  
LDAP.IMPORT_USER=TRUE  
LDAP.IMPORT_USER.LOGIN_ATTRIBUTE=sAMAccountName  
LDAP.IMPORT_USER.NAME_ATTRIBUTE=givenName,sn  
LDAP.IMPORT_USER.EMAIL_ATTRIBUTE=mail  
LDAP.IMPORT_USER.GROUP_ATTRIBUTE=memberof
```



```
LDAP.IMPORT_USER.PHONE_ATTRIBUTE=telephoneNumber
LDAP.IMPORT_USER.ABBREVIATION_ATTRIBUTE=initials
```

`LDAP.NAME`: Bezeichnung der entsprechenden LDAP-Section, z. B. der Domain-Name. Die Beschreibung erscheint im Dialog "Benutzer editieren" (siehe Kapitel 7.2.4.2 Seite 226).

`LDAP.HOST_URL`: LDAP-URL der LDAP-Section in der Form `ldap://hostname` (falls `LDAP.SSL=false`) oder `ldaps://hostname` (falls `LDAP.SSL=true`). Zur Erhöhung der Ausfallsicherheit können mehrere LDAP-Server eingetragen werden, die jeweils dieselben LDAP-Daten bereitstellen müssen.

`LDAP.SSL`: Hier kann die verschlüsselte SSL-Übertragung aktiviert (Wert: `true`) oder deaktiviert (Wert: `false`) werden (siehe Kapitel 4.4.3).

`LDAP.AUTHENTICATION`: Es gibt unterschiedliche Möglichkeit für die Anmeldung am Server. Mögliche Werte sind:

- `BIND`: Name und Passwort werden an den LDAP-Server geschickt. Dazu muss der "Distinguished Name" (DN), d.h. der eindeutige Schlüssel zur Identifizierung des Users innerhalb des LDAP-Servers bekannt sein. Falls der DN existiert, wird das übergebene Passwort mithilfe der "Bind"-Operation geprüft. `BIND` kann nur verwendet werden, wenn die LDAP-DNs der Benutzerkonten alle innerhalb desselben LDAP-Ordners liegen. Im Beispiel zum Parameter `LDAP.BIND.DN` (s.u.) ist das der Ordner mit dem DN

```
ou=Benutzer,ou=Dortmund,dc=e-spirit,dc=de.
```

Falls die Benutzerkonten über verschiedene Ordner verteilt sind, muss entweder `SEARCH_BIND` verwendet oder für jeden der Benutzer-Ordner eine eigene LDAP-Section in `fs-server.conf` eingetragen werden.

- `SEARCH_BIND`: Falls der "Distinguished Name" (DN) eines Users nicht bekannt ist, oder Benutzerkonten in verschiedenen Zweigen des LDAP-Baums eingetragen sind, kann man ihn innerhalb eines Teilbaums des LDAP-Servers suchen. Dazu muss ein Suchfilter definiert werden. Beispiel:

```
SEARCH.FILTER=(uid=$USER_LOGIN$)
SEARCH.BASE_DN=ou=department,dc=mycompany,dc=com
```

Dieser Filter sucht nach allen Einträgen im LDAP-Baum, bei



denen das Attribut "uid" gleich dem eingegebenen Login-Namen ist. Startknoten ist der Knoten mit dem DN "ou=department,dc=mycompany,dc=com". Die Anmeldung ist erfolgreich, sobald ein auf den SEARCH.FILTER passendes Benutzerkonto gefunden wird und das übergebene Benutzer-Passwort korrekt ist.

- `SEARCH_COMPARE`: Diese Option funktioniert äquivalent zum `SEARCH_BIND`, wobei hier aber nicht das Passwort-Attribut zur Authentifizierung verwendet wird, sondern ein beliebiges anderes LDAP-Attribut. Beispiel:

```
SEARCH.COMPARE.PASSWORD_ATTRIBUTE_NAME=mail
```

In diesem Fall muss das eingegebene Passwort mit dem Inhalt des "mail"-Attributes übereinstimmen.



Bei Verwendung von `LDAP.AUTHENTICATION=SEARCH_BIND` oder `LDAP.AUTHENTICATION=SEARCH_COMPARE` ist in der Regel die Angabe von `LDAP.SEARCH.BIND_DN` und `LDAP.SEARCH.BIND_PASSWORD` (s.u.) notwendig. Die Angabe ist nur dann nicht notwendig, wenn der LDAP-Server ohne Authentifizierung abgefragt werden kann. Dies kommt aber in Produktivumgebungen eigentlich nicht vor.

`LDAP.BIND.DN`: DN der Benutzerkonten, die zur Anmeldung an FirstSpirit verwendet werden können. Dieser Parameter ist nur sinnvoll in Verbindung mit `LDAP.AUTHENTICATION=BIND`. Als Platzhalter für den FirstSpirit-Benutzernamen wird `$USER_LOGIN$` eingetragen. Beispiel:

```
LDAP_1.BIND.DN=cn=$USER_LOGIN$,ou=Dortmund,dc=e-spirit,dc=de
```

Komplettes Konfigurationsbeispiel unter Verwendung von

`LDAP.AUTHENTICATION=BIND` für Active Directory:

```
LDAP_1.NAME=e-Spirit
LDAP_1.HOST_URL=ldap://ldapsver1 ldap://ldapsver2
LDAP_1.SSL=FALSE
LDAP_1.AUTHENTICATION=BIND
LDAP_1.BIND.DN=cn=$USER_LOGIN$,ou=Benutzer,ou=Dortmund,dc=e-spirit,dc=de
LDAP_1.IMPORT_USER=TRUE
LDAP_1.IMPORT_USER.LOGIN_ATTRIBUTE=sAMAccountName
LDAP_1.IMPORT_USER.NAME_ATTRIBUTE=displayName
LDAP_1.IMPORT_USER.EMAIL_ATTRIBUTE=mail
LDAP_1.IMPORT_USER.GROUP_ATTRIBUTE=memberof
LDAP_1.IMPORT_USER.PHONE_ATTRIBUTE=telephoneNumber
LDAP_1.IMPORT_USER.ABBREVIATION_ATTRIBUTE=initials
```



`LDAP.SEARCH.BIND_DN`: LDAP-DN eines technischen Benutzerkontos, das zum Durchsuchen des LDAP-Servers verwendet wird, um ein DN eines anzumeldenden FirstSpirit-Benutzer zu finden.

`LDAP.SEARCH.BIND_PASSWORD`: Passwort zum LDAP-DB des für `LDAP.SEARCH.BIND_DN` verwendeten technischen Benutzerkontos.

`LDAP.SEARCH.BASE_DN`: Der Parameter definiert den Startknoten der Suche nach LDAP-DN der anzumeldenden FirstSpirit-Benutzer.

`LDAP.SEARCH.FILTER`: Über den Parameter wird ein Suchfilter definiert. Der Filter: `SEARCH.FILTER=(cn=$USER_LOGIN$)` sucht beispielsweise nach allen Einträgen im LDAP-Baum, bei denen das Attribut "cn" gleich dem unter FirstSpirit eingegebenen Login-Namen ist. Startknoten ist der bei `LDAP.SEARCH.BASE_DN` angegebene DN.

`LDAP.IMPORT_USER`: Neben der reinen Authentifizierung ist es möglich, beliebige LDAP-Attribute in die Benutzerattribute eines CMS-Benutzers zu übernehmen. Dazu muss der Parameter `LDAP.IMPORT_USER` auf den Wert `TRUE` gesetzt werden.

`LDAP.IMPORT_USER.LOGIN_ATTRIBUTE`: Diese Zuordnung übernimmt den Login-Namen eines LDAP-Benutzers für einen FirstSpirit-Benutzer. Der hier zugeordnete LDAP-Name wird nach dem ersten Login des jeweiligen Benutzers automatisch importiert. Falls mehrere LDAP-Sections (`LDAP_1`, `LDAP_2`, ...) vorhanden sind, also die Benutzer-Anmeldungen aus verschiedenen Benutzer-Domains erfolgt, ist z. B. unter Active Directory meistens `sAMAccountName` als Attribut nicht eindeutig über alle Domains, so dass hier beispielsweise der vollständige User-Principal `userPrincipalName` eingetragen werden sollte, der auch den Domain-Namen enthält. Wird für diesen Parameter ein ungültiger Wert angegeben (z. B. LDAP-Attribut liefert einen "leeren" Wert zurück, ungültiges LDAP-Attribut usw.), wird dies folgendermaßen 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!
```

`LDAP.IMPORT_USER.NAME_ATTRIBUTE`: Diese Zuordnung übernimmt den Benutzernamen eines LDAP-Benutzers für einen CMS-Benutzer. Der hier zugeordnete LDAP-Name wird nach dem ersten Login des



jeweiligen Benutzers automatisch importiert.

`LDAP.IMPORT_USER.EMAIL_ATTRIBUTE`: Diese Zuordnung übernimmt die E-Mail-Adresse eines LDAP-Benutzers für den CMS-Benutzer. Die hier zugeordnete LDAP-E-Mail-Adresse wird beim ersten Login des jeweiligen Benutzers automatisch importiert.

`LDAP.IMPORT_USER.GROUP_ATTRIBUTE`: Über das LDAP-Gruppenattribut kann ein Benutzer automatisch einer bestimmten FirstSpirit-Gruppe eines FirstSpirit-Projekts anhand seiner Gruppenzugehörigkeit im LDAP zugewiesen werden.

Angegeben wird der Name des LDAP-Attributs, das LDAP-DNs der LDAP-Gruppen enthält, in denen der jeweilige Benutzer Mitglied ist. Das Attribut wird bei jeder Anmeldung des Benutzers erneut ausgelesen, um die Gruppenzugehörigkeit in FirstSpirit zu übernehmen. Dem Benutzerkonto werden alle als extern markierten FirstSpirit-Gruppen zugewiesen, deren in FirstSpirit zugewiesener "externer Name" mit der LDAP-DN der LDAP-Gruppe übereinstimmt. Statt eines LDAP-DNs kann auch jede beliebige Zeichenkette verwendet werden, wobei LDAP-Server aber üblicherweise Gruppenzugehörigkeit über DNs abbilden.

`LDAP.IMPORT_USER.PHONE_ATTRIBUTE`: Diese Zuordnung bindet die Telefonnummer eines LDAP-Benutzers an einen CMS-Benutzer. Die hier zugeordnete LDAP-Telefonnummer wird beim ersten Login des jeweiligen Benutzers automatisch importiert.

`LDAP.IMPORT_USER.ABBREVIATION_ATTRIBUTE`: Diese Zuordnung bindet das Kürzel eines LDAP-Benutzers an einen CMS-Benutzer. Das hier zugeordnete LDAP-Kürzel wird beim ersten Login des jeweiligen Benutzers automatisch importiert.

Weiterführende Informationen zur Anbindung eines LDAP-Servers siehe Kapitel 4.4 Seite 104.

4.3.1.11 Bereich: Storage Engine Properties

```
#####  
# storage engine properties  
#####  
# use one shared cache for all repositories  
repository.sharedCache=1
```

Alle Properties der Berkeley DB können auch in der Konfigurationsdatei



`fs-server.conf` verwendet werden. Dazu muss jeweils das Präfix "repository." vor dem entsprechende Property angegeben werden.

Die entsprechenden Properties befinden sich in der Dokumentation zur Berkeley DB³.

`repository.sharedCache`: Dieser Parameter ist standardmäßig auf den Wert 1 gestellt. Damit teilen sich die Berkeley DBs aller Projekte auf einem Server einen Cache. Auf diese Weise wird gerade die Verwaltung des Berkeley-Caches für Server mit vielen Projekten vereinfacht und ist insgesamt effektiver. Soll für jedes Projekt ein eigener Cache für die Berkeley DB verwendet werden, kann der Wert auf 0 gestellt werden. Dann kann für jedes Projekt der Cache-Speicher in den Projekteigenschaften eingestellt werden. Siehe dazu Kapitel 7.4.2 Seite 285, "Cache-Speicherbedarf".



Bei Änderungen an der Datei `fs-server.conf` ist ein Neustart des Servers erforderlich (vgl. Kapitel 4.2 Seite 29).

4.3.1.12 Bereich: CacheManager

```
#####  
# cache manager  
#####  
CACHE_PERCENT=25
```

In diesem Bereich kann entweder die absolute oder eine relative (prozentuale) Größe für den Speicherbedarf des Caches eingestellt werden. Dabei sollte der Wert nicht auf 0 gesetzt werden.

`CACHE_SIZE`: Absolute Größe des Caches in Bytes. Gültige Werte sind z. B. 456458345, 128m oder 4096k.

`CACHE_PERCENT`: Prozentuale Größe des Caches vom `-Xmx` Wert. Gültige Werte sind z. B. 30 oder 0.5. Standardmäßig wird der Wert 25 gesetzt.

³ <http://www.oracle.com/database/berkeley-db/index.html>



4.3.1.13 Bereich: internal Database

```
#####  
# internal database  
#####  
internalDB.port=1527  
internalDB.host=
```

`internalDB.port`: TCP-Port, auf dem der JDBC-Konnektor des internen Datenbanksystems gestartet wird. Standardwert: 0

`internalDB.host`: IP-Adresse oder Hostname, an den der JDBC-Konnektor gebunden werden soll. Standardwert: "", damit werden alle Adressen verwendet.

Der FirstSpirit-Server bringt bereits ein einfaches relationales Datenbanksystem mit (Apache Derby), das direkt nach der Serverinstallation zur Verfügung steht. Diese Datenbank ist allerdings nicht für den Produktivbetrieb geeignet und sollte daher lediglich für Tests verwendet werden.

Über die Bindeadresse können externe Clients oder Webanwendungen von externen Servern auf die interne Datenbank (Derby) zugreifen. Ist der Wert für `internalDB.port` auf 0 gesetzt, wird diese Funktion deaktiviert. Dies ist standardmäßig konfiguriert, um eine potentielle Sicherheitslücke zu schließen.

Die Zugangsdaten für die Datenbank befinden sich in der `fs-database.conf` (siehe Kapitel 4.3.3 Seite 76).

Ein Beispiel zur Konfiguration der Derby-Datenbank für eine Verwendung von externen Prozessen aus (z. B. Webanwendung mit dem FirstSpirit-Modul `DynamicDatabaseAccess` im externen Application-Server) findet sich in Kapitel 4.8.7.6 Seite 172.



4.3.1.14 Bereich: WEBedit configuration

```
#####  
# WEBedit configuration  
#####  
webedit.sessionCache.lru_size=0  
webedit.globalCache.lru_size=5120  
web.sessionCache.lru_size=0
```

`webedit.sessionCache.lru_size`: Der Parameter bestimmt die Größe des LRU⁴-Caches für jede einzelne WebClient-Sitzung. Der LRU-Cache ist der Cache für alle FirstSpirit-Objekte, die innerhalb des Projekts referenziert werden. Der hier definierte Wert, gibt Auskunft darüber, wie lange die referenzierten Objekte im Cache erhalten bleiben. Es wird unterschieden zwischen "Weak References" (Wert < 0), die direkt aus dem Cache gelöscht werden, sobald sie nicht mehr referenziert werden, und "Soft References" (Wert >= 0), die abhängig von der verwendeten VM, solange im Cache verbleiben, wie genügend Speicher vorhanden ist. (Bei einem großen Speicher sind Weak-References zumeist von Vorteil.)

`webedit.globalCache.lru_size`: Der Parameter bestimmt die Größe des DTO⁵-Caches aller WebClient-Sessions. Der hier definierte Wert bestimmt die Anzahl der Store-Elemente, die im DTO-Cache gespeichert werden können. Im DTO-Cache werden die reinen Datenobjekte eines Projekts abgelegt. Dabei arbeiten alle Benutzer auf den gleichen Datenobjekten (im Gegensatz zu den Projektobjekten innerhalb der Verwaltungen, die nur lokal innerhalb einer Sitzung gültig sind). Der Cache wird also über alle WebClient-Sitzungen geteilt (globaler Cache).

`web.sessionCache.lru_size`: Der Parameter bestimmt die Größe des Caches für die Projektobjekte einer Web-Session, die beispielsweise für die Vorschau-Erstellung benötigt werden. Der hier definierte Wert bestimmt die Anzahl der Store-Elemente, die im Cache gespeichert werden können.

⁴ LRU: Least recently used

⁵ DTO: Data Transfer Object



4.3.1.15 Bereich: Misc

```
#####  
  
# misc  
  
#####  
  
# comma seperated list of directories to check, e.g.  
# hdd.directories=${cmsroot}/data/projects,${cmsroot}/web/  
hdd.directories=${cmsroot}  
  
# hdd warning limit, use hdd.limit.active=false to turn off  
warning  
hdd.limit=90  
hdd.limit.active=true  
  
# hdd server shutdown limit, use hdd.shutdown.active=false to turn  
off shutdown  
hdd.shutdown=95  
hdd.shutdown.active=true
```

BATCHPATH: Der Parameter definiert den Pfad zu den Skriptdateien, die z. B. beim Deployment ausgeführt werden können.

LICENSE_EXPIRATION_WARNING_DAYS: Der Parameter definiert, wie viele Tage vor Ablauf einer FirstSpirit Lizenz, eine tägliche Lizenzauflaufs-Warnung erfolgen soll. Beträgt der angegebene Wert "0", wird keine Mail verschickt (Standardwert ist 30 Tage).

LICENSE_EXPIRATION_MAIL_ADDRESS: Der Parameter definiert, an welche E-Mail-Adresse die Lizenzauflaufs-Warnung verschickt werden soll. Wird keine E-Mail-Adresse definiert, wird die E-Mail an den Administrator (siehe Parameter `ADMIN_MAIL_ADRESS` in Kap. 4.3.1.9 Seite 49) geschickt.

BACKUP_PATH: Pfadangabe zum Backup-Verzeichnis, das Verzeichnis im Dateisystem, in dem der FirstSpirit-Server die Backups ablegen soll. Backups werden über den Auftrag "Projektsicherung durchführen" (siehe Kapitel 7.5.10.3 Seite 391) oder die lizenzabhängige Funktionalität "FirstSpirit EnterpriseBackup" (siehe entsprechende Modul-Dokumentation) angelegt. Standardmäßig werden Backups im Unterverzeichnis `backup` des FirstSpirit-Servers abgelegt.

mail.backup-recipient: Über diesen Parameter kann eine E-Mail-Adresse angegeben werden, die verwendet wird, wenn die Parameter `backup.size.limit`, `backup.min.age` und / oder



`backup.max.age` (siehe unten) aktiviert sind.

`backup.size.limit`: Über diesen Parameter kann eingestellt werden, dass eine E-Mail an die über den Parameter `mail.backup-recipient` definierte E-Mail-Adresse gesendet wird, wenn die Größe der Backup-Dateien die angegebene Größe überschreitet. Beispielwerte sind `500m` oder `3g`. Standardmäßig ist kein Wert angegeben, was bedeutet, dass die Funktion deaktiviert ist.

`backup.min.age`: Über diesen Parameter kann eingestellt werden, dass Backup-Dateien nach dem angegebenen Zeitraum (in Tagen) automatisch gelöscht werden, wenn der über den Parameter `backup.size.limit` definierte Wert erreicht ist. Standardmäßig ist kein Wert angegeben, was bedeutet, dass die Funktion deaktiviert ist. Um die Funktionalität zu aktivieren, muss ein Wert angegeben werden. Werden Backup-Dateien gelöscht, wird eine E-Mail an die über den Parameter `mail.backup-recipient` definierte E-Mail-Adresse gesendet.

`backup.max.age`: Über diesen Parameter kann eingestellt werden, dass Backup-Dateien nach dem angegebenen Zeitraum (in Tagen) automatisch gelöscht werden. Standardmäßig ist kein Wert angegeben, was bedeutet, dass die Funktion deaktiviert ist. Um die Funktionalität zu aktivieren, muss ein Wert angegeben werden. Werden Backup-Dateien gelöscht, wird eine E-Mail an die über den Parameter `mail.backup-recipient` definierte E-Mail-Adresse gesendet.

`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. Siehe auch Kapitel 7.3.1 Seite 231, Option "Verzeichnisse für Plattenplatz-Überprüfung (kommasepariert)" und folgende.

`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.limit.active`: Über diesen Parameter kann der Versand von Warnungs-E-Mails unterbunden werden. Dazu muss `hdd.limit.active=false` gesetzt werden. Standardmäßig ist der Versand bei der angegebenen Prozentzahl jedoch aktiviert.

`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.

`hdd.shutdown.active`: Über diesen Parameter kann das Herunterfahren des FirstSpirit-Servers unterbunden werden. Dazu muss `hdd.shutdown.active=false` gesetzt werden. Standardmäßig ist das Herunterfahren bei der angegebenen Prozentzahl jedoch aktiviert.

`SERVICES`: Über diesen Parameter können die Systemdienste definiert werden. (Diese Dienste können beim Start des Servers mitgestartet werden.)

4.3.1.16 Bereich: Webstart configuration

Zum Starten der FirstSpirit Server- und Projektkonfiguration und des FirstSpirit-JavaClients wird das Oracle Java Runtime Environment (JRE) benötigt, darin ist Java Web Start enthalten. (Das JRE wird i.d.R. automatisch bei der Installation von JDK installiert. Zu unterstützten Versionen siehe *FirstSpirit Technisches Datenblatt*.) Die Konfiguration erfolgt über JNLP-Dateien:

- global (serverweit) über die FirstSpirit Server- und Projektkonfiguration (siehe Kapitel 7.3.7 Seite 249) oder über Parameter in der Konfigurationsdatei `fs-server.conf` (siehe unten)
- benutzerspezifisch über die Verbindungseinstellungen auf der FirstSpirit-Startseite (siehe Kapitel 6.3.4.1 Seite 192).

Webstart-Konfiguration für den Start des FirstSpirit-JavaClients: Die hier konfigurierten Parameter wirken sich auf alle Schnellstart-Einträge der Startseite vom Typ JavaClient **JAVA** aus, sofern nicht explizit andere Parameter für den Eintrag



im Bereich "Startseite" definiert wurden (siehe Kapitel 7.3.8 Seite 250).

```
webstart.client.connection=  
webstart.client.server=  
webstart.client.port=  
webstart.client.memory=  
webstart.client.compression=  
webstart.client.encryption=  
webstart.client.servletZone=  
webstart.client.parameters=
```

Die Konfigurationsmöglichkeiten entsprechen der benutzerspezifischen Webstart-Konfiguration (siehe Kapitel 6.3.4.1 Seite 192).



Für den Parameter `webstart.client.encryption` kann keine TLS-Verschlüsselung (Parameterwert 1) verwendet werden, wenn Websphere als Application-Server für die Webanwendung `fs5root` eingesetzt wird. In diesem Fall bietet sich an, RC4-Verschlüsselung (Parameterwert 2) für SOCKET oder HTTP einzusetzen oder HTTPS zu verwenden. Bei HTTPS ist keine zusätzliche Verschlüsselung notwendig, so dass dann der Parameterwert 0 einzusetzen ist.

Register Server- und Projektkonfiguration: Konfiguration für die Anwendung zur Server- und Projektkonfiguration.

```
webstart.admin.connection=  
webstart.admin.server=  
webstart.admin.port=  
webstart.admin.memory=  
webstart.admin.compression=  
webstart.admin.encryption=  
webstart.admin.servletZone=  
webstart.admin.parameters=
```

Die Konfigurationsmöglichkeiten entsprechen der benutzerspezifischen Webstart-Konfiguration (siehe Kapitel 6.3.4.1 Seite 192).



Für den Parameter `webstart.admin.encryption` kann keine TLS-Verschlüsselung (Parameterwert 1) verwendet werden, wenn Websphere als Application-Server für die Webanwendung `fs5root` eingesetzt wird. In diesem Fall bietet sich an, RC4-Verschlüsselung (Parameterwert 2) für SOCKET oder HTTP einzusetzen oder HTTPS zu verwenden. Bei HTTPS ist keine zusätzliche Verschlüsselung notwendig, so dass dann der Parameterwert 0 einzusetzen ist.



4.3.1.17 Bereich: JumpToServlet and Webedit ForwardAction

`allowedRedirectHosts`: An einigen Stellen in FirstSpirit werden Redirect-URLs erzeugt (z. B. Verweise aus Remote-Projekte in der Vorschau oder für die WebClient-Vorschau), die potenziell auch auf externe URLs verweisen können, z. B.

```
http://localhost:5000/jump?url=http://www.heise.de
```

oder

```
http://localhost:5000/fs5webedit/Dispatcher?project=1183078&language=DE&weAction=Forward&forward=http://www.heise.de
```

Über den optionalen Parameter `allowedRedirectHosts` können URLs definiert werden, zu denen eine Weiterleitung erlaubt werden soll. Folgende Modi sind dabei möglich:

```
allowedRedirectHosts=ALLOW_ALL
```

Es sind Weiterleitungen an alle URLs ohne Einschränkungen erlaubt.

```
allowedRedirectHosts=fs.mywebsite.de,heise.de,intranet.mywebsite.de
```

Auf diese Weise kann eine Whitelist der erlaubten Ziele erstellt werden. Die erlaubten URLs werden dabei mit Komma separiert angegeben.

```
allowedRedirectHosts=FS_SERVER
```

Es wird eine Whitelist der erlaubten URLs aus folgenden Quellen erstellt:

1) `fs-server.conf`, Parameter

<code>URL</code>	(Kapitel 4.3.1.1 Seite 32)
<code>JNLP_SERVLET_URL</code>	(Kapitel 4.3.1.2 Seite 35)
<code>WEBAPP_ROOT_URL</code>	(Kapitel 4.3.1.7 Seite 43)
<code>WEBAPP_WEBEDIT5_URL</code>	(Kapitel 4.3.1.7 Seite 43)
<code>WEBAPP_WEBMON_URL</code>	(Kapitel 4.3.1.7 Seite 43)
<code>WEBAPP_STAGING_URL</code>	(Kapitel 4.3.1.7 Seite 43)
<code>WEBAPP_PREVIEW_URL</code>	(Kapitel 4.3.1.7 Seite 43)
<code>preview.externalDeliveryURL</code>	(Kapitel 4.3.1.8 Seite 47)

2) in den Server-Eigenschaften konfigurierte Web-Server (Anwendung zur "Server- und Projektkonfiguration" / "Server" / "Eigenschaften" / "Webserver", siehe Kapitel 7.3.12 Seite 261).

Dies ist die Standard-Einstellung.

Wird versucht, eine nicht erlaubte URL aufzurufen, wird der HTTP-Statuscode 403 (Forbidden) ausgegeben.



4.3.1.18 Bereich: JMX

```
#####
# JMX
#####
# Listen host and port. If port is empty, JMX is disabled
jmx.host=${HOST}
jmx.port=

# SSL and keystore
jmx.ssl=false
jmx.ssl.needClientAuth=false
javax.net.ssl.keyStore=
javax.net.ssl.keyStorePassword=

# User authentication and access level. If password file is empty, authentication
is disabled
jmx.password.file=
jmx.access.file=
```

Konfiguration des JMX-Connectors zur (Remote-)Überwachung der JVM des FirstSpirit-Servers (siehe Kapitel 9 Seite 476). JMX dient zur Abfrage des Systemstatus und liefert aktuelle Systeminformationen des FirstSpirit-Servers und Java-Systems. Als Client kann z. B. `jconsole` (enthalten im JDK) verwendet werden oder andere System-Monitore die das JMX-Protokoll unterstützen.

`jmx.host`: Angabe eines Hostnamens für die JMX- / RMI-Verbindung (JMX-Connector).

`jmx.port`: Angabe einer freien Portnummer für die JMX- / RMI-Verbindung (JMX-Connector). Wird kein Port angegeben, ist kein JMX-Zugriff möglich.

`jmx.ssl`: Die Verwendung von Secure Sockets Layer (SSL) ist standardmäßig deaktiviert (Standardwert: `false`).

`jmx.ssl.needClientAuth`: Die Verwendung einer Client-SSL-Authentifizierung ist standardmäßig deaktiviert (Standardwert: `false`). Um die Client-SSL-Authentifizierung für das Remote-Monitoring per JMX zu aktivieren, muss dieser Parameter auf `true` gesetzt werden.

`javax.net.ssl.keyStore`: Über den Parameter wird der Dateisystem-Pfad zum Zertifikatsspeicher (Keystore) angegeben.

`javax.net.ssl.keyStorePassword`: Angabe eines Passworts für die Keystore-Datei.

`jmx.password.file`: Aktiviert die Benutzerauthentifizierung für den JMX-Zugriff auf dem FirstSpirit-Server. Über den Parameter wird der Dateisystem-Pfad zur JMX-Passwort-Datei angegeben. Die JMX-Passwort-Datei verwaltet verschiedene Rollen/Benutzer und deren Passwörter. Hinweis: Da die in dieser Datei hinterlegten



Passwörter im Klartext gespeichert werden, sollten hier nicht die Standard-Authentifizierungsinformationen hinterlegt werden, sondern speziell für den JMX-Zugriff definierte Rollen und Passwörter (siehe `jmx.access.file`). Das JRE enthält eine Vorlage für eine Passwort-Datei mit dem Namen `jmxremote.password.template`. Diese Vorlage kann nach `JRE_HOME/lib/Management/jmxremote.password` kopiert und um Passwörter für die Rollen ergänzt werden, die in der JMX-Access-Datei definiert sind. Ist kein Wert angegeben (Standardfall), ist die JMX-Authentifizierung deaktiviert.

`jmx.access.file`: Über den Parameter wird der Dateisystem-Pfad zur JMX-Access-Datei (`jmxremote.access`) angegeben. Die Access-Datei verwaltet verschiedene Rollen/Benutzer und deren Zugriffsrechte. Die hier verwalteten Rollen müssen den Rollen in der Passwort-Datei entsprechen. Der zugehörige Wert muss entweder "readonly" oder "readwrite" sein. So kann beispielsweise eine "monitorRole" definiert werden, die ausschließlich den Lesezugriff für die Überwachung erlaubt und eine "controlRole", die den Lese- und Schreibzugriff für die Überwachung und Verwaltung gewährt.



Für den Betrieb der JMX-Konsole in einer Produktionsumgebung sollten immer die Benutzerauthentifizierung und ggf. der verschlüsselte SSL-Zugriff aktiviert werden. Sind diese Parameter deaktiviert (Standardeinstellung), ist der Zugriff auf den JMX-Port nicht geschützt und unberechtigte Benutzer könnten über den JMX-Port den Server beenden.

Weitere Informationen zur Konfiguration von JMX siehe:

<http://docs.oracle.com/javase/6/docs/technotes/guides/management/agent.html#gdemv>



4.3.1.19 Weiterreichen von Authentifizierungscookies

Einige HTTP-Server können für den Verbindungsaufbau bzw. für die Kommunikation eine Benutzerauthentifizierung voraussetzen. In diesem Fall werden nach einer erfolgreichen Authentifizierung im HTTP-Client (z. B. einem Webbrowser) mithilfe eines Cookies Informationen hinterlegt (z. B. eine Ticketnummer), um eine erneute Authentifizierung bei weiteren Aufrufen zu vermeiden.

Diese Authentifizierungscookies können für die HTTP-Kommunikation zwischen dem FirstSpirit JavaClient und HTTP-Server verwendet werden, d. h. der Cookie wird vom Webbrowser über Java Web Start an einen aufgerufenen FirstSpirit JavaClient weitergereicht.

Welche Cookies für die Weiterreichung berücksichtigt werden sollen, kann mithilfe des Parameters `clientCookieNames` definiert werden.

Als Wert erwartet der Parameter eine kommaseparierte Liste von Cookie-Namen, die für die Weiterreichung genutzt werden sollen:

```
clientCookieNames=cookieA,cookieB...
```

Aktualisierungen der Cookiewerte durch den HTTP-Server werden für die HTTP-Verbindungen entsprechend berücksichtigt.

Ab FirstSpirit 5.0R4 werden diese Cookies auch an die, für das Projekt konfigurierte Browser-Engine weitergereicht und sind damit in der integrierten Vorschau des JavaClients verfügbar. Das gleiche Cookie wird dann in drei unterschiedlichen Sitzungskontexten (Webbrowser: HTTP-Client, JavaClient: Client/Server-Kommunikation, JavaClient: Integrierte Browser-Engine) verwendet.



Eine Re-Injektion des/der Cookies z. B. beim Wechsel der Browser-Engine ist NICHT möglich!



Die Client-/Serverkommunikation erfolgt in vielen Fällen parallel in mehreren Threads, daher kann die Reihenfolge der Aktualisierung der Cookies nicht vorherbestimmt werden. D.h. wenn auf der Ebene der Security-Infrastruktur z. B. eine One-Time-Cookie-Verwaltung implementiert ist, die erwartet, dass neue Requests immer das Cookie des vorhergehenden Requests mitschicken, schlägt dies durch die Nebenläufigkeit der HTTP-Requests des Clients unter Umständen fehl.



4.3.2 Konfiguration der Java-VM und des -Wrappers (fs-wrapper.conf)

Im Unterverzeichnis `conf` des FirstSpirit-Servers befindet sich die Datei `fs-wrapper.conf`, die wichtige Konfigurationseinstellungen für den Serverstart und das Java-System des FirstSpirit-Servers enthält.

Die Konfigurationsdatei ist zuständig für Start und Stop des Java-Prozesses und enthält Parameter zur optimalen Ausnutzung des Hauptspeichers des Host-Betriebssystems.

Änderungen an der Konfigurationsdatei `fs-wrapper.conf` können über das FirstSpirit Server-Monitoring vorgenommen werden (siehe Kapitel 8.6.1.5 Seite 454). Die Änderungen werden dann in die Konfigurationsdatei geschrieben und auf dem Server aktualisiert. Werden ungültige Konfigurationen vorgenommen, wird beim Speichern der Konfiguration im Server-Monitoring auf den Fehler hingewiesen, und der Speichervorgang wird nicht ausgeführt:

```
unexpected configuration property key 'wrapper.startUp.timeout' in line 76
```

Wenn Zugriff auf das Dateisystem besteht, können Änderungen an der Konfigurationsdatei auch direkt dort erfolgen.

Eine vollständige Beschreibung aller Parameter des Java-Wrappers und weiterführende Informationen finden Sie unter:

<http://wrapper.tanukisoftware.org/doc/english/properties.html>.

Die Datei besitzt unter Windows und UNIX denselben Aufbau. Bei den Pfadnamen sollte auf die systemunabhängige Notation mit "/" geachtet werden. Die einzelnen Bereiche und die zugehörigen änderbaren Parameter werden nachfolgend erläutert. Die Reihenfolge der Parameter in der Datei ist beliebig.

- Konfiguration der Java-VM (Kapitel 4.3.2.1)
- Allgemeine Parameter (Kapitel 4.3.2.3)
- Protokollierung (Log-Files) (Kapitel 4.3.2.4)
- Systemdienst unter Windows (Kapitel 4.3.2.5)

Im Kapitel 12.1 finden Sie die bei der Installation mit Standardwerten angelegte Datei.





Bei allen Änderungen der Konfigurationsdatei `fs-wrapper.conf` ist ein Neustart des Servers erforderlich.

Innerhalb der Konfigurationsdatei `fs-wrapper.conf` dürfen Kommentare nicht in einer Zeile mit Parameter-Werten stehen, z. B.:

`wrapper.startup.timeout=30 # Kommentar`

4.3.2.1 Konfiguration der Java-VM

Da FirstSpirit als Anwendung innerhalb einer virtuellen Java-Maschine (Java-VM) läuft und die zurzeit verfügbaren Java-VMs über keine dynamische Speicherverwaltung gegenüber dem Betriebssystem verfügen, sind zur optimalen Skalierung an den Anwendungsfall einige Parameter bezüglich der Speicherverwaltung der Java-VM manuell zu konfigurieren.

Der für FirstSpirit nutzbare Speicherbereich ist der sogenannte "Heap" der Java-VM. Dieser Heap sollte so groß wie möglich eingestellt werden, aber nicht größer als der im Betriebssystem freie Hauptspeicher sein. Als Standardeinstellung sollten 75% des RAM eingestellt werden, sofern keine anderen Dienste parallel zu FirstSpirit auf dem Server betrieben werden.

Ein Heap wesentlich größer als 10 GByte kann mit den hier genannten Parametern nicht problemlos verwaltet werden. Falls diese Größenordnung notwendig ist, muss das Verhalten der Garbage-Collection über JMX mittels `jconsole` oder `VisualVM` analysiert und die Java-VM-Parameter an den Anwendungsfall im Detail angepasst werden.



Sofern ein Heap größer als 10 GByte eingestellt werden soll, muss zunächst beim Hersteller (e-Spirit AG) angefragt werden, da in solchen Fällen meist weitere, spezielle Parameter zum Konfigurieren des Garbage Collectors notwendig sind.

Die Größe des Heap wird über die Parameter `wrapper.java.initmemory` bzw. `wrapper.java.initmemory.percent` und `wrapper.java.maxmemory` bzw. `wrapper.java.maxmemory.percent` eingestellt (Kapitel 4.3.2.3).

Ein großer Java-Heap (über 1 GByte) erfordert eine Optimierung der Parameter zur Anpassung der Garbage Collection, die im folgenden Abschnitt beschrieben wird.



Große Datenmengen in FirstSpirit-Projekten oder eine hohe Anzahl gleichzeitig aktiver FirstSpirit-Redakteure können den Garbage-Collector der Java-VM in deren Standardkonfiguration überlasten. Die Überlastung macht sich durch Wartezeiten im Bereich größer als 10s in der Antwortzeit der FirstSpirit-Clients bemerkbar. Bei längeren Wartezeiten können auch Verbindungsabbrüche zwischen FirstSpirit-Client und -Server auftreten. Die Ursache für die Wartezeit ist die Garbage-Collection, die in der Standardkonfiguration den FirstSpirit-Server für bestimmte Operationen temporär vollständig anhält.

In der Log-Datei des Garbage-Collectors (`log/fs-wrapper.log` bzw. `log/fs-gc.log`) ist zu diesem Zeitpunkt die Meldung "Full GC" oder "time exceeded" zu sehen.

Um das vollständige Anhalten des FirstSpirit-Servers durch den Garbage-Collector zu verhindern, muss ein anderes Verfahren als die Standardkonfiguration für den Garbage Collector aktiviert werden. Die Java-VM von Oracle stellt dazu seit Version 1.6.0 den nebenläufig arbeitenden Garbage Collector zur Verfügung: Concurrent Mark Sweep GC (CMS-GC). Die Java-VM von IBM bietet einen vergleichbaren Garbage Collector zur Vermeidung langer GC-Pausen (`-Xgcpolicy:optavgpause`).

Die im folgenden Abschnitt genannten Parameter werden bereits während der Installation verwendet. Es ist nach der Installation nur noch eine Anpassung der Größe des Heaps in einigen Parametern notwendig, die Angaben in absoluten MByte enthalten. Alle Parameter, die nur Verhältnisse oder Prozentangaben enthalten, können unverändert beibehalten werden.

4.3.2.2 Allgemeine Java-Parameter

Folgende Parameter in der Datei `conf/fs-wrapper.conf` gelten für Java-VMs aller Hersteller bei Verwendung des FirstSpirit-Servers.

```
wrapper.java.additional.1=-Djava.awt.headless=true
wrapper.java.additional.2=-Djava.security.auth.login.config=conf/fs-jaas.conf
wrapper.java.additional.3=-Djava.security.policy=conf/fs-server.policy
wrapper.java.additional.4=-Dfile.encoding=UTF-8
```



4.3.2.2.1 Konfiguration der Java-VM von Oracle

Zur Verwendung des FirstSpirit-Servers in der Java-VM von Oracle sind nachfolgend beschriebene Zeilen in der Datei `conf/fs-wrapper.conf` notwendig.

Zunächst wird auf 64-Bit-Systemen die 64-Bit-Java-VM aktiviert:

```
wrapper.java.additional.5=-d64
```

Sofern nur die 32-Bit-Java-VM verwendet werden soll, wird der Parameter "-d64" entfernt.

Zur Konfiguration des CMS-GC dienen nachfolgende Zeilen:

```
wrapper.java.additional.6=-Xshare:off
wrapper.java.additional.7=-Xmn600M
wrapper.java.additional.8=-XX:PermSize=500M
wrapper.java.additional.9=-XX:MaxPermSize=500M
wrapper.java.additional.10=-XX:+DisableExplicitGC
wrapper.java.additional.11=-XX:SoftRefLRUPolicyMSPerMB=1
wrapper.java.additional.12=-XX:+UseParNewGC
wrapper.java.additional.13=-XX:+UseConcMarkSweepGC
wrapper.java.additional.14=-XX:+CMSParallelRemarkEnabled
wrapper.java.additional.15=-XX:+CMSClassUnloadingEnabled
wrapper.java.additional.16=-XX:SurvivorRatio=1
wrapper.java.additional.17=-XX:-UseLargePages
wrapper.java.additional.18=-Djava.rmi.dgc.leaseValue=3600000
```

Der bei `-Xmn` angegebene Wert (M steht hier für MByte) definiert den Bereich des Java-Heaps, der für temporäre Java-Objekte verwendet wird. Für FirstSpirit, das viele temporäre Objekte verwendet, empfiehlt sich hier 50% des bei `wrapper.java.initmemory` angegebenen Werts.

Der bei `-XX:MaxPermSize` angegebene Wert (M steht hier für MByte) definiert den Bereich des Java-Heaps, der für Java-Klassen und JSP-Seiten verwendet wird. Falls der interne Web-Server (Jetty) verwendet wird und FirstSpirit-Projekte viele JSP-Dateien umfassen oder viele Beanshell-Skripte in FirstSpirit-Vorlagen verwendet werden, sollte dieser Wert erhöht werden. Über den FirstSpirit Web-Monitor kann unter "Monitoring -> VM-Speicher" die Auslastung dieses Heap-Bereichs über einen Zeitraum der letzten Stunde beobachtet werden. Normalerweise belegt FirstSpirit relativ konstant 100 MByte in diesem Bereich.

Die Protokollierung des Garbage Collectors der Java-VM auf Seiten des FirstSpirit-Servers erfolgt standardmäßig in eine eigene Datei (`log/fs-gc.log`). Dafür sind folgende Parameter verantwortlich:



```
wrapper.java.additional.X=-verbose:gc
wrapper.java.additional.X=-XX:+PrintGCTimeStamps
wrapper.java.additional.X=-XX:+PrintGCDetails
wrapper.java.additional.X=-XX:+PrintGCDateStamps
wrapper.java.additional.X=-Xloggc:log/fs-gc.log
```

Das X in `wrapper.java.additional.X=` ist dabei immer ein Platzhalter für eine eindeutige Zahl.

Über den Parameter `-Xloggc` kann der Name und Speicherort der Log-Datei bei Bedarf geändert werden. Die Datei wird automatisch umbenannt bei Erreichen einer in `conf/fs-wrapper.conf` definierten Größe archiviert und nach mehreren Archivierungsschritten, ebenfalls in `conf/fs-wrapper.conf` einzustellen, gelöscht

Meldungen werden in die Log-Datei `fs-gc.log` sowie in eine Datei `fs-gc.*.log` geschrieben, wobei sich * aus Datum und Uhrzeit des ersten Eintrags ergibt. Wird eine fest eingestellte Dateigröße von 5 MB erreicht, wird die aktuelle Log-Datei in einer Datei `fs-gc.*.log.gz` komprimiert und archiviert, wobei sich * wiederum aus Datum und Uhrzeit des ersten Eintrags ergibt. Archivierte Dateien können mit dem Server-Auftrag "Clean up logs" gelöscht bzw. nach `firstspirit5/backup` verschoben werden.

Weiterführende Informationen zu den einzelnen Parametern sind unter <http://www.oracle.com/technetwork/java/javase/tech/vmoptions-jsp-140102.html> zu finden.

4.3.2.2.2 Konfiguration der Java-VM von IBM

Um den Garbage Collector zur Minimierung der Wartezeiten der Java-Anwendung zu aktivieren, sind folgende Zeilen in der Datei `conf/fs-wrapper.conf` notwendig:

```
wrapper.java.additional.5=-Xgcpolicy:optavgpause
```

Andere mögliche Parameter für `Xgcpolicy`:

`optthruput`: Standardeinstellung, verursacht aber lange Wartezeiten der Java-Anwendung. Der Vorteil ist ein ansonsten hoher Datendurchsatz.

`optavgpause`: Nebenläufiger GC mit minimalen Wartezeiten. Der Nachteil ist eine höhere CPU-Auslastung und geringerer Datendurchsatz.



gencon: Verbesserter nebenläufiger GC mit zusätzlicher Aufteilung des Heaps in mehrere Generationen.

Folgende Parameter aktivieren außerdem das Protokollieren der Garbage-Collector-Aufrufe in die Datei `log/fs-gc.log` (siehe dazu auch Kapitel 4.3.2.2.1 Seite 69):

```
wrapper.java.additional.6=-Xverbosegclog:log/fs-gc.log  
wrapper.java.additional.7=-verbose:gc
```

Alle weiteren Parameter im Bereich "wrapper.java.additional", außer den in Kapitel 4.3.2.2 genannten, müssen bei Verwendung der Java-VM von IBM entfernt werden.

Eine detaillierte Beschreibung der Konfiguration der Garbage Collection des IBM-JDK finden Sie im "Java Diagnostics Guide" im Kapitel "Reference, Command Line Options" auf <http://publib.boulder.ibm.com/infocenter/javasdk/v6r0/> und auf der Seite <http://www.ibm.com/developerworks/java/jdk/diagnosis/>.

4.3.2.3 Parameter des Java-Wrappers

`wrapper.java.command`: Java-Interpreter. Entweder nur `java` (unter Unix und Windows), sofern die Umgebungsvariable `PATH` auf das richtige JDK zeigt, oder ein absoluter Pfad zum Java-Interpreter des JDK z. B. `/opt/jdk1.7.0/bin/java` oder `c:\JDK1.7.0\bin\java.exe`.

`wrapper.java.maxmemory`: Maximale Heapsize für die Java-VM in MByte. Dies ist der Anteil am Speicher des Betriebssystems, den der FirstSpirit-Server maximal nutzen kann. Er sollte so groß wie möglich gewählt werden, aber nicht größer als das physikalische RAM angegeben werden. Bei 32-Bit-Systemen ist dieser Wert auf ungefähr 2 GByte begrenzt.

Falls auf einem Rechner mehrere FirstSpirit-Server oder andere Java-Prozesse betrieben werden, muss die Heapsize aller Java-Prozesse entsprechend aufgeteilt werden, so dass die Größe des verfügbaren RAMs nicht überschritten wird. Der hier definierte Wert findet sich in der Grafik zum Speicherverbrauch im FirstSpirit Server-Monitoring wieder (vgl. Abbildung 8-1 und Beschreibung in Kapitel 8.1.1).





Sofern ein Heap größer als 10 GByte eingestellt werden soll, muss zunächst beim Hersteller (e-Spirit AG) angefragt werden, da in solchen Fällen meist weitere, spezielle Parameter zum Konfigurieren des Garbage Collectors notwendig sind.

`wrapper.java.maxmemory.percent`: Dieser Parameter hat dieselbe Bedeutung wie `wrapper.java.maxmemory`. Die maximale Heapsize wird hier allerdings prozentual zum physikalischen RAM angegeben. Bei 32-Bit-Systemen bezieht sich der Wert nicht auf die Größe des Hauptspeichers sondern auf 2 GByte. Die Standardeinstellung ist 75 und braucht im Normalfall nicht geändert zu werden. Falls auf einem Rechner mehrere FirstSpirit-Server oder andere Java-Prozesse betrieben werden, muss die Heapsize aller Java-Prozesse entsprechend aufgeteilt werden, so dass die Größe des verfügbaren RAMs nicht überschritten wird.

`wrapper.java.initmemory`: Heapsize, die die Java-VM zu Beginn reserviert. Dieser Wert sollte auf 75% des bei `wrapper.java.maxmemory` angegebenen Werts eingestellt werden, um die restlichen 25% als Puffer für Situationen temporär hoher Auslastung in Reserve verfügbar zu haben. Die Java-VM versucht, den Speicherverbrauch auf den bei `initmemory` eingestellten Wert konstant zu halten. Falls während des Betriebs eine dauerhafte Speicherbelegung oberhalb dieses Werts beobachtet wird, ist eine Überlastung eingetreten, die einen höheren Startwert benötigt.

`wrapper.java.initmemory.percent`: Dieser Parameter hat dieselbe Bedeutung wie `wrapper.java.initmemory`. Die Heapsize wird hier allerdings prozentual zum physikalischen RAM angegeben. Bei 32-Bit-Systemen bezieht sich der Wert nicht auf die Größe des Hauptspeichers sondern auf 2 GByte. Die Standardeinstellung ist 50 und braucht im Normalfall nicht geändert zu werden.



Die Parameter

`wrapper.java.maxmemory` **und** `wrapper.java.maxmemory.percent` können alternativ verwendet werden, jedoch nicht gleichzeitig.

Dasselbe gilt für die Parameter

`wrapper.java.initmemory` **und** `wrapper.java.initmemory.percent`.



`wrapper.java.additional.X`: Parameter, die direkt an die Java-VM übergeben werden. Hier werden hauptsächlich Java-Parameter zur Konfiguration des Garbage Collectors eingetragen.



Nur ein Java-Parameter pro Zeile. Alle angegebenen Java-Parameter müssen eine eindeutige Nummerierung (X) enthalten. Solange der Parameter `wrapper.ignore_sequence_gaps` auf `true` steht, braucht die Nummerierung nicht fortlaufend zu sein.

`wrapper.*.timeout`: Maximale Bearbeitungszeiten in Sekunden für einige Systemzustände des FirstSpirit-Servers. Bei Überschreiten dieser Zeitangaben beendet der Wrapper den Java-Prozess, da angenommen wird, dass er sich in einem undefinierten Zustand befindet. Parameternamen für *: `startup`, `shutdown`, `jvm_exit`, `cpu`, `ping`.

`wrapper.timer_slow_threshold`: Falls der interne Zeitgeber des Wrappers um die angegebene Anzahl von Sekunden von der Systemuhr abweicht, wird eine Warnung nach `log/fs-wrapper.log` geschrieben. Dieser Parameter kann genutzt werden, um eine CPU-Überlastung zu erkennen, da in diesem Fall der Wrapper-Zeitgeber nicht genug Rechenzeit zum Aktualisieren erhält und langsamer laufen wird.

`wrapper.umask`: Nur Unix: Alle neugeschriebenen Dateien des FirstSpirit-Servers erhalten die Zugriffsattribute der angegebenen `umask`.

4.3.2.4 Protokollierung

`wrapper.logfile.*`: Parameter zur Protokollierung in die Datei `log/fs-wrapper.log`. Die maximale Dateigröße sowie Anzahl der Archivkopien kann geändert werden. Über den Parameter `loglevel` kann zwischen `INFO` für den Produktionsbetrieb und `DEBUG` für Testbetrieb umgeschaltet werden.

`wrapper.console.*`: Parameter zur Protokollierung auf die aktuelle Standardausgabe der Konsole. Die Protokollierung auf die Konsole ist nur aktiv, falls der FirstSpirit-Server unter Unix über "fs5 console" gestartet wurde oder unter Windows über "Startmenü -> FirstSpirit5 -> In Konsole starten"

`wrapper.syslog.*`: Nur Unix: Konfiguration zur Protokollierung in das systemweite



syslogd-System. Falls die Protokollierung des FirstSpirit-Servers über `fs-logging.conf` auf `stdout` eingestellt ist (siehe Kapitel 4.3.6), wird auch diese zum syslogd statt nach `fs-server.log` gesendet.

Der FirstSpirit-Server loggt einmalig beim Starten und zyklisch jede Stunde die VM-StartTime und die VM-Uptime:

Suchergebnis (2)	
INFO	<code>[-] 11.05.2009 16:08:33 (de.espirit.firstspirit.server.CMSServer): Uptime 25.245.018, StartTime 1.242.025.668.098 (11.05.2009 09:07:48)</code>
INFO	<code>[-] 11.05.2009 15:08:32 (de.espirit.firstspirit.server.CMSServer): Uptime 21.644.501, StartTime 1.242.025.668.098 (11.05.2009 09:07:48)</code>

Abbildung 4-1: FirstSpirit Server-Monitoring: Log VM-Starttime und VM-Uptime



Neben der Datei `fs-wrapper.log` werden u.a. für die Fehlersuche wichtige Logmeldungen auch in die Datei `fs-server.log` geschrieben (siehe Kapitel 4.3.6 Seite 97).

4.3.2.5 Systemdienst unter Windows



Änderungen im folgenden Bereich werden erst nach Neuregistrieren des FirstSpirit-Systemdienstes wirksam! Der Systemdienst kann über Startmenü -> FirstSpirit 5.0 -> Installation -> Dienst deregistrieren / registrieren neu registriert werden. Der FirstSpirit-Server wird dabei gestoppt und gestartet.

`wrapper.ntservice.name`: Objektname des FirstSpirit-Systemdienstes.

`wrapper.ntservice.displayname`: Angezeigter Name des Systemdienstes.

`wrapper.ntservice.description`: Beschreibungstext zum Systemdienst.

`wrapper.ntservice.dependency.x`: Für FirstSpirit notwendige Systemdienste, die in der Startreihenfolge vor FirstSpirit liegen sollen, z. B. MySQL. Für jeden Dienst eine neue Zeile mit eindeutiger Nummerierung (X) verwenden.

`wrapper.ntservice.starttype`: Startverfahren, entweder `AUTO_START` zum automatischen Starten beim Systemstart oder `DEMAND_START` zum



manuellen Starten.

`wrapper.ntservice.interactive`: Interaktion mit dem Desktop ist für den FirstSpirit-Server nicht notwendig, sollte immer auf `false` eingestellt sein.

4.3.2.6 Sonstige Parameter

Die folgenden Parameter sollten nicht geändert werden, weil FirstSpirit zur korrekten Funktion auf die dort voreingestellten Werte angewiesen ist:

```
wrapper.working.dir
wrapper.app.parameter.X
wrapper.java.classpath.X
wrapper.java.library.path.X
wrapper.java.mainclass
wrapper.max_failed_invocations
wrapper.on_exit.*
wrapper.restart.reload_configuration
wrapper.commandfile
wrapper.command.poll_interval
wrapper.ignore_sequence_gaps
```



4.3.3 Konfiguration der Datenbankanbindung (fs-database.conf)

Im Unterverzeichnis `conf` des FirstSpirit-Servers befindet sich die Datei `fs-database.conf`, die wichtige Konfigurationseinstellungen für die Anbindung einer Datenbank an einen FirstSpirit Server enthält und bei Bedarf angepasst werden muss.

Änderungen an der Konfigurationsdatei `fs-database.conf` können über die FirstSpirit Server- und Projektkonfiguration vorgenommen werden (siehe Kapitel 7.3.5 Seite 241). Die Änderungen werden dann in die Konfigurationsdatei geschrieben und auf dem Server aktualisiert. (Wenn Zugriff auf das Dateisystem besteht, können Änderungen an der `fs-database.conf` auch direkt über die Konfigurationsdatei erfolgen.)



Bei direkten Änderungen der Konfigurationsdatei `fs-database.conf` über das Dateisystem wird die Datei nicht automatisch auf dem Server aktualisiert. Änderungen sollten daher immer über die Server- und Projektkonfiguration erfolgen.

```
DATABASES=derby_project
derby_project.jdbc.layerclass=de.espirit.ormapper.or.layer.DerbyLayer
derby_project.jdbc.PASSWORD=p16062532
derby_project.jdbc.URL=jdbc:derby:projects/project_14110/derby;create=true
derby_project.jdbc.USER=user0
derby_project.jdbc.POOLMAX=1
derby_project.jdbc.POOLMIN=1
derby_project.jdbc.DRIVER=org.apache.derby.jdbc.EmbeddedDriver
```

Weitere Informationen zur Konfiguration der Datenbankanbindung siehe Kapitel 4.8 Seite 142. Die Parameter werden in den Kapiteln

- 4.8.4.1: Pflichtparameter
- 4.8.4.2: optionale Parameter
- 4.8.4.3: oracle-spezifische Parameter und
- 4.8.4.4: MS-SQL spezifische Parameter

ab Seite 158 ff. erläutert.



4.3.4 Konfiguration des Anmeldevorgangs (fs-jaas.conf)

Im Unterverzeichnis `conf` des FirstSpirit-Servers befindet sich die Datei `fs-jaas.conf`, die Konfigurationseinstellungen für den Anmeldevorgang am FirstSpirit-Server enthält.

Änderungen an der Konfigurationsdatei `fs-jaas.conf` können über die FirstSpirit Server- und Projektkonfiguration (siehe Kapitel 7.3.10 Seite 253) oder über das Server-Monitoring vorgenommen werden (siehe Kapitel 8.6.1.8 Seite 456). Die Änderungen werden dann in die Konfigurationsdatei geschrieben und auf dem Server aktualisiert. Wenn Zugriff auf das Dateisystem besteht, können Änderungen an der `fs-jaas.conf` auch direkt über die Konfigurationsdatei erfolgen. Kommentare werden mit `//` eingeleitet.

Im Kapitel 12.2 finden Sie die bei der Installation mit Standardwerten angelegte Datei.



Bei Änderungen der Konfigurationsdatei `fs-jaas.conf` über das Dateisystem wird die Datei automatisch auf dem Server aktualisiert (Standard: alle 60 Sek.). Ein Neustart des Servers ist nicht erforderlich.

Zur Benutzerauthentifizierung verwendet FirstSpirit den Java-Standard JAAS⁶. Folgende JAAS-Module sind bereits in FirstSpirit integriert und stellen verschiedene Verfahren zur Benutzerauthentifizierung bereit:

4.3.4.1 Passwortprüfung gegen die FirstSpirit-Benutzerdatenbank

JAAS-Modulname: `de.espirit.firstspirit.server.authentication.FSUserLoginModule`

Es wird die FirstSpirit-interne Benutzerdatenbank verwendet.

⁶ Java Authentication and Authorization Service:
<http://docs.oracle.com/javase/1.5.0/docs/guide/security/jgss/tutorials/>



4.3.4.2 LDAP

JAAS-Modulname: `de.espirit.firstspirit.server.authentication.LdapLoginModule`

Das `LdapLoginModule` bietet 2 Funktionen:

1. Authentifizierung: Es erfolgt die Prüfung der auf der FirstSpirit-Startseite eingegebenen Kombination aus Benutzernamen und Passwort gegen das angegebene LDAP-Verzeichnis. Für diesen Anwendungsfall wird das `LdapLoginModule` in der Datei `fs-jaas.conf` im Bereich `webplain` eingetragen.
2. Autorisierung: Nachdem die Authentifizierung über ein beliebiges JAAS-Modul erfolgt ist, werden nun die Informationen zur Gruppenzugehörigkeit des angemeldeten Benutzers aus dem LDAP-Verzeichnis ausgelesen. Falls die Authentifizierung über Passwort erfolgte, wird diese 2. Funktion automatisch schon während der Authentifizierung durchgeführt und es ist keine zusätzliche Konfiguration notwendig. Falls die Authentifizierung über ein passwortloses Ticket-Verfahren erfolgt, muss das `LdapLoginModule` in der Datei `fs-jaas.conf` im Bereich `webssso` in der Reihenfolge hinter dem verwendeten Authentifizierungsmodul eingetragen werden.

Für beide Funktionen wird ein externer LDAP-Server verwendet, z. B. die LDAP-Komponente eines Active Directory Servers. Über den Konfigurationsparameter `section` des `LdapLoginModuls` erfolgt der Verweis auf eine in `fs-server.conf` definierte LDAP-Section, in der die Zugangsdaten zum LDAP-Server eingetragen sind, siehe Kapitel 4.3.1.10. Es darf immer nur 1 LDAP-Section als Parameter übergeben werden. Falls mehrere LDAP-Verzeichnisse verwendet werden sollen, also mehrere LDAP-Sections, muss für jede LDAP-Section das `LdapLoginModul` erneut in die Datei `fs-jaas.conf` eingefügt werden.



4.3.4.3 Anmelde-Ticket aus FirstSpirit-Benutzerdatenbank

JAAS-Modulname: de.espirit.firstspirit.server.authentication.**FSTicketLoginModule**

Zur Authentifizierung reicht ein Berechtigungsnachweis (Ticket) aus, der zuvor vom FirstSpirit-Server generiert wurde. Der Berechtigungsnachweis wird bei der Anmeldung auf der FirstSpirit-Startseite generiert und über den Webbrowser weitergereicht. Dieses Verfahren ist das Standardverfahren nach der FirstSpirit-Installation.

4.3.4.4 Anmelde-Ticket aus Windows-NETBIOS-Domäne (NTLM)

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. 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.

JAAS-Modulname: de.espirit.firstspirit.server.authentication.**NTLMLoginModule**

Es wird ein Berechtigungsnachweis (Ticket) akzeptiert, der bei der Anmeldung innerhalb einer Windows-Domäne erstellt wird. Ein Redakteur braucht sich damit nur einmal an seinem Arbeitsplatzrechner anzumelden, da der Webbrowser den Berechtigungsnachweis automatisch an FirstSpirit übermittelt. Als Webbrowser wird für dieses Anmeldeverfahren derzeit nur Microsoft Internet Explorer unterstützt. Über den Parameter `domains` werden die zur Anmeldung zugelassenen Windows-Domänen angegeben. Optional zusätzlich können Domänen-Server angegeben werden.

Einträge für den Parameter "domains" sind in dieser Form möglich:
"Browser-Domain:Domain-Controller1,Domain-Controller2".

Es können Einträge für mehrere Domains vorgenommen werden, die alle nacheinander zur Anmeldung überprüft werden. Als Trennzeichen wird ; verwendet.

Beispiel:

```
"Browser-Domain1:dc1,dc2;Browser-Domain2:dc3,dc4"
```

Über den Parameter `userAgents`: Hier kann ein Suchmuster angegeben werden, um die NTLM-Anmeldung nur für ausgewählte Webbrowser zu aktivieren, da NTLM einen nicht ganz standardkonformen HTTP-Header ("WWW-Authenticate:



Negotiate") verwendet, den einige ältere Webbrowser als Fehler interpretieren. Um NTLM bei allen Webbrowsern zu verwenden ".*" eintragen.

Standardwert: ".*MSIE.*"

Das Modul unterstützt NETBIOS und Active Directory Domains. "Browser-Domain" ist die vom Webbrowser in der Anmeldeberechtigung an den FirstSpirit-Server übergebene Domain. Bei der Anmeldung wird ein zur Browser-Domain passender Eintrag gesucht und dann die Anmeldeberechtigung zur Prüfung an den angegebenen Domain-Controller gesendet.

Ist keine Domain eingetragen, wird die Anmeldeberechtigung unabhängig von der vom Browser übergebenen Domain immer an den/die eingetragenen Domain-Controller geprüft, Beispiel "Domaincontroller1,Domaincontroller2".

- Falls als Browser **Firefox** verwendet wird, ist nur folgende Konfiguration sinnvoll, weil Firefox die Domain des Benutzerkontos nicht zum Server überträgt: "Domaincontroller1,Domaincontroller2".
- Falls als Browser **Internet Explorer** verwendet wird, sind folgende Konfigurationen sinnvoll: "Domaincontroller1,Domaincontroller2" oder "Browser-Domain:Domaincontroller1,Domaincontroller2".
- Sofern beide Browser verwendet werden sollen, können die Konfigurationen mit ; getrennt kombiniert werden.



Falls bei Verwendung des Internet-Explorers die Anmeldung mit Berechtigungsnachweis nicht funktioniert, sind die Sicherheitseinstellungen wie in Abbildung 4-2 gezeigt anzupassen. Die Benutzerauthentifizierung sollte auf "Automatische Anmeldung mit aktuellem Benutzernamen" eingestellt sein. Zusätzlich muss der Hostname des FirstSpirit-Servers bei "Vertrauenswürdige Sites" hinzugefügt werden.



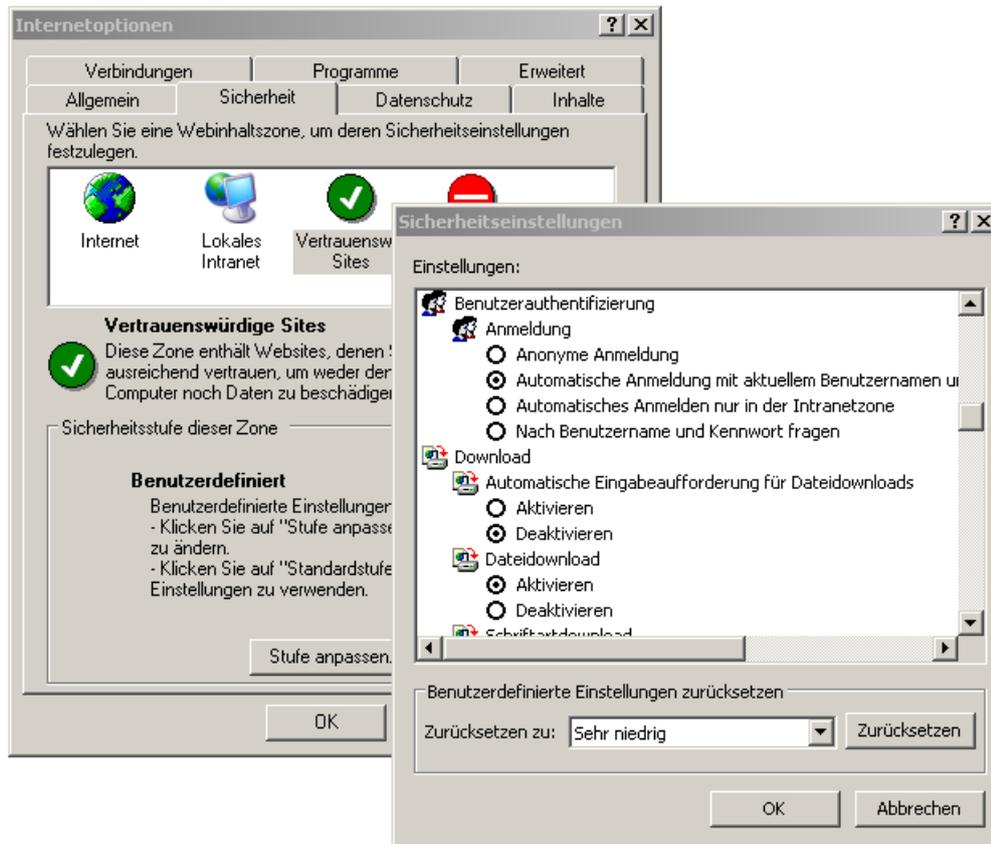


Abbildung 4-2: Sicherheitseinstellungen anpassen

Einstellungen im Betriebssystem zur Anpassung an das Verhalten unter NTLM(v1)

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":



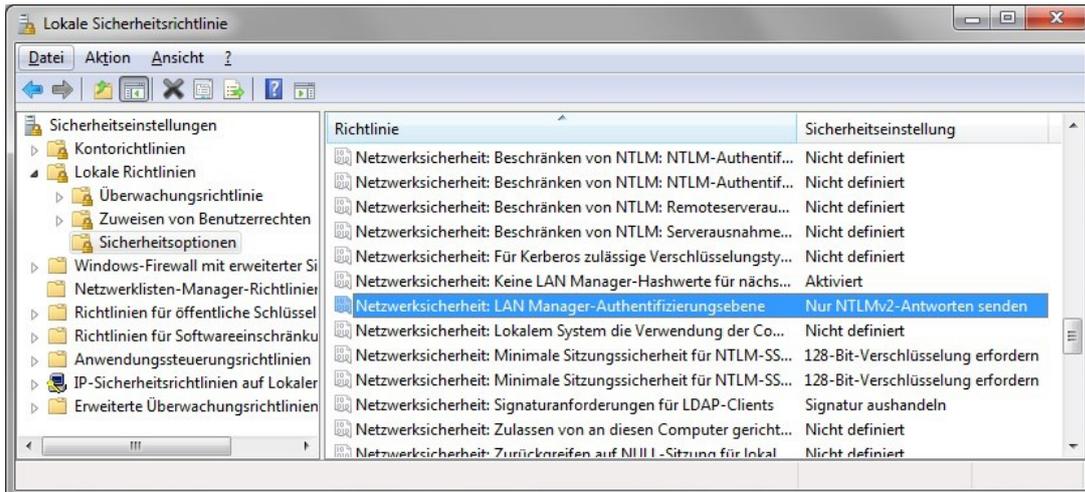


Abbildung 4-3: Netzwerksicherheit: LAN-Manager-Authentifizierungsebene

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):

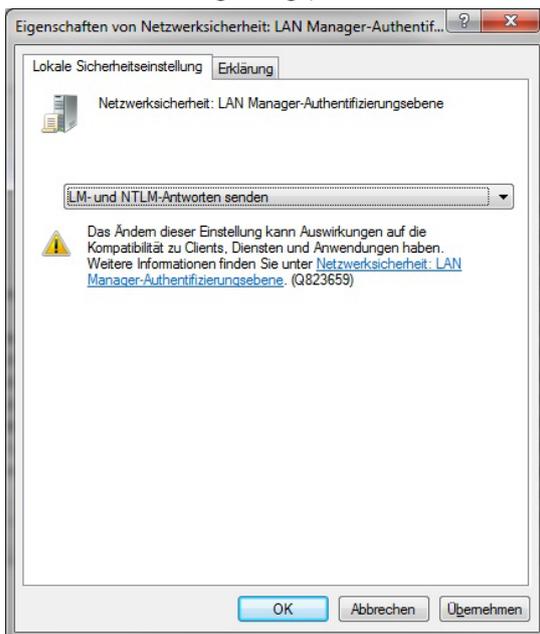


Abbildung 4-4: Standardeinstellung in Windows XP

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





Neben dem NTLM-Loginmodul steht auch das Kerberos-Loginmodul zur Verfügung (siehe Kapitel 4.3.4.5 Seite 83). Für die Verwendung von Kerberos muss – im Gegensatz zu NTLM – keine Änderung der Einstellungen im Betriebssystem vorgenommen werden und ist die zu bevorzugende Variante.

4.3.4.5 Kerberos-Ticket (Integrierte Windows-Anmeldung)

JAAS-Modulname: de.espirit.firstspirit.server.authentication.**KerberosLoginModule**

Zur Anmeldung wird ein Kerberos-Ticket akzeptiert, das vom Webbrowser an den FirstSpirit-Server übergeben wird. Der FirstSpirit-Redakteur braucht sich bei Verwendung dieses Login-Moduls nur einmal morgens an seinem Arbeitsplatzrechner (Windows, Mac, GNU/Linux) anzumelden, unter der Voraussetzung, dass eine Kerberos-Infrastruktur vorhanden ist, und kann anschließend FirstSpirit ohne erneute Passworteingabe benutzen. Unter Microsoft Windows ist Kerberos seit Windows 2003 und XP als "Integrierte Windows-Anmeldung" bekannt und ersetzt damit NTLM.

Parameter:

`useFullPrincipal`: Legt fest, ob der vollständige Kerberos-Anmeldename inklusive @-Zeichen und Realm (Wert "true") oder ohne @ und Realm (Wert "false") als FirstSpirit-Benutzername verwendet wird. "false" ist bei Systemen ausreichend, deren Benutzerkonten alle in einem Kerberos-Realm (entspricht unter Windows einer Active Directory Domain) eingetragen sind. Falls Anmeldungen von mehreren Kerberos-Realms oder Active Directory Domains erfolgen, muss "true" angegeben werden, da meistens der reine Benutzername über mehrere Domains nicht eindeutig ist.
Standardwert: "false"

`userAgents`: Hier kann ein Suchmuster angegeben werden, um die Kerberos-Anmeldung nur für ausgewählte Webbrowser zu aktivieren, da Kerberos einen nicht ganz standardkonformen HTTP-Header ("WWW-Authenticate: Negotiate") verwendet, den einige ältere Webbrowser als Fehler interpretieren. Um Kerberos bei allen Webbrowsern zu verwenden ".*" eintragen.
Standardwert:
".*(Firefox|Iceweasel|Konqueror|MSIE|Opera|Safari|Shiretoko).*"



Das KerberosLoginModule wird im Bereich `webssso` der Datei `fs-jaas.conf` eingetragen. Zusätzlich muss folgender neuer Bereich in derselben Datei am Dateiende eingefügt werden:

```
com.sun.security.jgss.accept {
  com.sun.security.auth.module.Krb5LoginModule required
  principal="HTTP/fs5host.mydomain.net@MYDOMAIN.NET"
  keyTab="/opt/firstspirit5/conf/fs5host-HTTP.keytab"
  useKeyTab="true"
  storeKey="true"
  isInitiator="false"
  doNotPrompt="true"
  debug="true";
};
```

Die Pfade und Domain-Bezeichnungen müssen entsprechend des lokalen Systems angepasst werden. Folgende Parameter sind anzupassen:

`principal`: Hier wird der Service-Principal-Name des FirstSpirit-Servers angegeben.

`keyTab`: Hier wird der Pfad zur Kerberos-Keytab-Datei angegeben, die den privaten Schlüssel, meistens in verschiedenen Verschlüsselungsverfahren (z. B. RC4, DES und AES), passend zum Service-Principal-Name enthält. Diese Datei muss zunächst erstellt werden, wie im nachfolgenden Abschnitt beschrieben.

Hinweise zum Service-Principal-Name (SPN):

Das Schlüsselwort "HTTP" gilt für die Verwendung von HTTP und HTTPS.

Der im SPN angegebene Hostname inklusive DNS-Domain muss der wirkliche Hostname des Servers sein. Falls ein virtueller Webserver verwendet wird, gibt es zwei Möglichkeiten: Sofern der virtuelle Webserver als CNAME-record im DNS eingetragen ist, wird im SPN der Hostname eingetragen, auf den der CNAME verweist. Falls der virtuelle Webserver als A-record im DNS eingetragen ist, wird der im A-record eingetragene Hostname im SPN verwendet. In beiden Fällen muss die IP-Adresse, auf den der im SPN eingetragene Hostname zeigt, wieder zurück auf den Hostnamen im SPN verweisen.



Erstellen der Kerberos-Keytab-Datei im Beispiel unter Microsoft Active Directory:

Zum Erstellen der Datei auf einem Kerberos-Server in einer Microsoft Active Directory Domain werden die zusätzlich zu installierenden Windows Support Tools⁷ benötigt, die von Microsoft auf den Installationsmedien des Betriebssystems mitgeliefert werden oder von <http://microsoft.com> heruntergeladen werden können.

Auf dem Windows Domain-Controller wird zunächst ein normales Benutzerkonto angelegt. Das Passwort darf nicht ablaufen und der Benutzer darf es nicht ändern können. Das Passwort ist irrelevant und wird im nächsten Schritt überschrieben. Als Benutzername sinnvollerweise "Hostname"- "Dienstname" angeben, also beispielsweise fs5host-HTTP. Um die Sicherheit zu erhöhen, kann die Option "Benutzer bei Delegation nicht vertrauen" aktiviert werden. Die Option "DES-Verschlüsselung für dieses Konto verwenden" darf nicht aktiviert sein, ansonsten funktioniert Kerberos mit RC4-Verschlüsselung nicht, die z. B. von Windows 2008 und Windows 7 verwendet wird.

Nun wird auf dem Windows Domain-Controller ein privater Schlüssel zum Service-Principal-Name mit RC4-Verschlüsselung erstellt, die normalerweise Standard ist in gemischten Netzen, die Windows XP, Vista, 7, 2003 oder 2008 sowie andere Betriebssysteme verwenden:

```
ktpass -princ HTTP/fs5host.mydomain.net@MYDOMAIN.NET \  
+rndpass -mapuser fs5host-HTTP \  
-crypto RC4-HMAC-NT -ptype KRB5_NT_PRINCIPAL \  
-out fs5host-http-rc4.keytab
```

Falls die verwendete Version des ktpass kein +rndpass bietet, kann hier auch ein manuell eingegebenes Zufallspasswort über -pass PASSWORT verwendet werden.

Falls zur Erhöhung der Sicherheit andere Kryptoalgorithmen verwendet werden sollen und diese vom Kerberos-Realm und den Clients unterstützt werden, können weitere keytab-Dateien erstellt werden, beispielsweise für AES256:

```
ktpass -princ HTTP/fs5host.mydomain.net@MYDOMAIN.NET \  
+rndpass -mapuser fs5host-HTTP \  
-crypto AES256-SHA1 -ptype KRB5_NT_PRINCIPAL \  
-out fs5host-http-aes256.keytab
```

⁷ http://en.wikipedia.org/wiki/Windows_Support_Tools



Ein Aufruf von `ktpass -h` zeigt die verfügbaren Kryptoalgorithmen an, wobei zu beachten ist, dass diese nur verwendet werden, wenn sie von allen anderen Domain-Servern und dem jeweiligen Client unterstützt werden.

Falls Probleme auftreten, kann folgendermaßen die Liste aller Service-Principal-Namen des Benutzerkontos angezeigt werden:

```
setspn -l fs5host-HTTP
```

Falls Fehler bei der Eingabe erfolgten, kann ein Service-Principal-Name über folgenden Aufruf entfernt werden:

```
setspn -d HTTP/fs5host.mydomain fs5host-HTTP
```

Die über `ktpass` erstellte `keytab`-Datei wird nun auf den FirstSpirit-Server bzw. den externen Application-Server kopiert, zu dem Pfad, wie er in `fs-jaas.conf` bei `keyTab` angegeben ist.

Beispiel: `/opt/firstspirit5/conf/fs5host-HTTP.keytab`.

Eine Überprüfung der Datei kann über folgenden Aufruf unter Unix erfolgen:

```
kinit -V -k -t fs5host-HTTP.keytab HTTP/fs5host.domain.net@DOMAIN.NET
```

Als Resultat sollte "Authenticated to Kerberos v5" angezeigt werden.

Falls mehrere Kryptoalgorithmen verwendet werden, müssen die einzelnen gerade erzeugten `Keytab`-Dateien zunächst zusammengefügt werden. Dazu das Dienstprogramm `ktutil` unter Unix starten:

```
/usr/sbin/ktutil
```

und folgende Eingaben vornehmen, um die `Keytab`-Datei im wie in `fs-jaas.conf` bei `keyTab` angegeben Pfad abzulegen:

```
rkt krb5-fs5host-HTTP-rc4.keytab
rkt krb5-fs5host-HTTP-aes256.keytab
wkt /opt/firstspirit5/conf/fs5host-HTTP.keytab
q
```

Falls für den FirstSpirit-Server ein externer Application-Server eingesetzt wird statt des integrierten Jetty, muss folgender Parameter, beispielsweise für Tomcat über die Umgebungsvariable "CATALINA_OPTS" dem Applications-Server beim Start übergeben werden:

```
-Djava.security.auth.login.config=/opt/firstspirit5/conf/fs-jaas.conf
```



Die Verwendung von Kerberos-Tickets vergrößert den HTTP-Request-Header auf bis zu einige KByte Länge, die die Standardkonfiguration bezüglich der Maximallänge von teilweise nur 4 kByte bei einigen Webservern überschreiten. Hier die Konfiguration für verschiedene Webserver, um die maximale Länge des HTTP-Request-Header zu vergrößern:

Jetty (in FirstSpirit integriert):

In der Datei `/opt/firstspirit5/conf/fs-webapp.xml` im Abschnitt `org.mortbay.jetty.nio.SelectChannelConnector` folgenden Parameter hinzufügen:

```
<Set name='requestHeaderSize'>65536</Set>
```

Tomcat:

In der Datei `tomcat/conf/server.xml` im Abschnitt "`<Connector protocol="HTTP/1.1"`" folgenden Parameter hinzufügen:

```
maxHttpHeaderSize="65536"
```

und im Abschnitt "`<Connector protocol="AJP/1.3"`" folgenden Parameter hinzufügen:

```
packetSize="65536"
```

Apache httpd:

In der Datei `httpd.conf`, oder bei virtuellen Hosts der äquivalenten Datei, folgenden Parameter hinzufügen:

```
LimitRequestLine 65536
```



Falls die Kerberos-Server nicht über DNS ermittelt werden können, also keine SRV-Record wie `_kerberos._udp.mydomain.net` oder `_kerberos._tcp.mydomain.net` vorhanden sind, wird auf Server-Seite die Datei `/etc/krb5.conf` bzw. `c:\windows\krb5.ini` benötigt:

```
[libdefaults]
    default_realm = MYDOMAIN.NET

[domain_realm]
    .mydomain.net = MYDOMAIN.NET
    mydomain.net = MYDOMAIN.NET

[realms]
    MYDOMAIN.NET = {
        kdc = dc1.mydomain.net
        kdc = dc2.mydomain.net
        kdc = dc3.mydomain.net
        default_domain = mydomain.net
    }
```

Die Konfiguration der Kerberos-basierten passwortlosen Anmeldung ist nun auf Server-Seite abgeschlossen.

Protokolliert werden die Kerberos-Fehlermeldungen in den Dateien `/opt/firstspirit5/log/fs-server.log` und `/opt/firstspirit5/log/fs-wrapper.log` sowie, falls Tomcat als externer Application-Server eingesetzt wird, in `tomcat/logs/firstspirit.log`.

Sobald die Kerberos-basierte Anmeldung von den Arbeitsplatzrechnern erfolgreich getestet wurde, muss in der Datei `/opt/firstspirit5/conf/fs-jaas.conf` der Parameter `debug="true"` in `debug="false"` geändert werden, um unnötig viele Logmeldungen zu vermeiden.

Sofern die Anmeldung nicht funktioniert, sollte zunächst in den Log-Dateien des FirstSpirit-Servers (`fs-wrapper.log` und `fs-server.log`) sowie bei Verwendung eines externen Application-Servers in dessen Log-Datei (z. B. `firstspirit.log` und `catalina.out`) nachgesehen werden. Ein häufiger Fehler sind zu große Zeitdifferenzen der einzelnen Rechneruhren, die bei Verwendung von Kerberos im Bereich von wenigen Minuten synchron laufen müssen.

Konfiguration der Clients

Auf Client-Seite sind abhängig vom verwendeten Webbrowser folgende Konfigurationen notwendig (falls passwortlose Kerberos-basierte Anmeldung bereits auf den Arbeitsplatzrechnern für andere Webserver innerhalb des Unternehmens-



Netzes verwendet wird, ist keine Konfigurationsänderung notwendig):

Internet Explorer (Windows):

In den Internetoptionen bei "vertrauenswürdige Sites" folgende Einträge einfügen: `https://*.mydomain.net` bzw. `http://*.mydomain.net`, falls nur HTTP verwendet wird. Dann bei den Internetoptionen unter "Erweitert" im Bereich Sicherheit "Integrierte Windows-Authentifizierung" aktivieren. Folgende Konfiguration ist eventuell zusätzlich notwendig: In den Internetoptionen bei Sicherheit in der Zone "Vertrauenswürdige Sites" "Stufe anpassen" wählen und im Bereich Benutzerauthentifizierung "Automatische Anmeldung mit aktuellem Benutzernamen und Kennwort" aktivieren.

Firefox (Windows, Mac OS, GNU/Linux):

Als URL in der Adresszeile `about:config` eingeben und bei dem Parameter `network.negotiate-auth.trusted-uris` den Domain-Namen des FirstSpirit-Servers mit führendem Punkt eintragen. Es können mehrere Domains durch Komma getrennt eingetragen werden. Beispiel: `.mydomain.net`

Safari (Mac OS):

Mac OS bietet bereits standardmäßig eine vollständige Kerberos-Integration, sofern das verwendete Benutzerkonto ein Netzwerk-basiertes Benutzerkonto ist und der Arbeitsplatzrechner an der Active Directory Domain bzw. Kerberos-Realm angemeldet ist. Es ist keine Konfigurationsänderung notwendig. Bei lokalen Benutzerkonten erfolgt beim ersten Zugriff auf die FirstSpirit-Startseite eine Abfrage des eigenen Kerberos-Benutzernames (`benutzername@MYDOMAIN.NET`) Principals inklusive Passwort.

Konqueror (GNU/Linux):

Sofern Kerberos im Betriebssystem des Arbeitsplatzrechners aktiviert wurde, also ein Kerberos-Ticket automatisch über `/etc/pam.d/common-auth` beim Anmelden und Entsperren des Bildschirms angefordert wird, ist keine weitere Konfiguration notwendig.



Falls Probleme beim Einrichten der Kerberos-basierten Anmeldung auftreten, kann Firefox mit Debug-Log gestartet werden, um zu sehen, welches Ticket der Browser sendet.

Windows:

Eingabeaufforderung starten (`cmd.exe`) und eingeben:

```
set NSPR_LOG_MODULES=negotiateauth:5
cd "\program files\mozilla firefox" || bzw. Pfad zur Firefox-
Installation
firefox -console
```

Unix:

Terminal-Fenster (`konsole`, `xterm`, o.ä.) öffnen und eingeben:

```
export NSPR_LOG_MODULES=negotiateauth:5
firefox
```



Sicherheitshinweis: Das `KerberosLoginModule` sollte aus Sicherheitsgründen in Produktsystemen zur erfolgreichen Verhinderung von Wiedereinspielungsangriffen (*Replay-Attacks*) nur in Verbindung mit `HTTPS` eingesetzt werden!

4.3.4.6 Anmelde-Ticket vom SAP-Server

JAAS-Modulname: `de.espirit.firstspirit.server.authentication.SAPLoginModule`

Es wird ein Berechtigungsnachweis (Ticket) akzeptiert, der bei der Anmeldung an einen SAP-Server erstellt wurde.

4.3.4.7 Anmeldung über Windows

JAAS-Modulname: `de.espirit.firstspirit.server.authentication.WindowsLoginModule`

Es wird die Anmeldung mittels Berechtigungsnachweis für den FirstSpirit-JavaClient ermöglicht, sofern dieser nicht über Java Web Start gestartet wird. Für den JavaClient über Java Web Start ist das `NTLMLoginModule` (siehe Kapitel 4.3.4.4 Seite 79) ausreichend. Dieses Modul kann nur für einen FirstSpirit-Server verwendet werden, der unter Windows installiert ist.



4.3.4.8 Allgemeine Hinweise zur JAAS-Konfiguration

Für alle Login-Module gilt, dass ein Benutzerkonto nach erfolgreicher Authentifizierung automatisch in das FirstSpirit-System übernommen wird. Als eindeutige Kennzeichnung wird der Anmeldename verwendet, so dass die Zuordnung von Benutzerkonten zu Projekten auch in Projektexporten gewährleistet ist.



Das automatische Anlegen von Benutzerkonten kann unterdrückt werden, indem der Parameter `JAAS.autoCreateUser` in die Datei `fs-server.conf` eingefügt und auf den Wert `false` gesetzt wird:

```
JAAS.autoCreateUser=false
```

Wird der Parameter nicht gesetzt, gilt standardmäßig der Wert `true`. Somit werden Benutzerkonten automatisch neu angelegt, wenn `JAAS.autoCreateUser` nicht gesetzt wird.

Die Login-Module können den FirstSpirit-Komponenten `JavaClient`, `WebClient`, `Webmonitor` und `Access-API` zugeordnet werden. Zur Zuordnung werden als Zwischenschritt zunächst symbolische Namen gewählt, die dann später einzelnen FirstSpirit-Komponenten zugeordnet werden. Unter den einzelnen symbolischen Namen werden in der Datei `fs-jaas.conf` jeweils 1 oder mehrere Login-Module eingetragen. Bei Eintragung mehrerer Login-Module werden diese in der angegebenen Reihenfolge abgearbeitet, bis eine erfolgreiche Authentifizierung des Benutzers erfolgt ist. Es sollte darauf geachtet werden, dass passwortlose Authentifizierungsverfahren mit Berechtigungsnachweis **vor** solchen mit Passwortüberprüfung eingetragen werden. Zusätzlich muss jedem Login-Modul das JAAS-Attribut `optional` zugewiesen werden. "Optional" bedeutet hier, dass mindestens eines der Login-Module eine erfolgreiche Authentifizierung durchgeführt haben muss, um dem Benutzer die Anmeldung an FirstSpirit zu erlauben. Andere JAAS-Attribute wie `sufficient`, `required` oder `requisite` dürfen für FirstSpirit nicht verwendet werden, da ansonsten FirstSpirit-spezifische Anmeldeattribute nicht von einem Login-Modul an das nächste weitergereicht werden. Diese FirstSpirit-spezifischen Anmeldeattribute sind auch der Grund dafür, dass externe JAAS-Module nur mit einer zusätzlichen Wrapper-Klasse für FirstSpirit verwendet werden können.

Als Standardbelegung werden folgende symbolische Namen verwendet: `plain`, `sso`, `webplain`, `websso`, `system`.



Die Zuordnung der symbolischen Namen zu den einzelnen FirstSpirit-Komponenten geschieht in der Datei `fs-server.conf` über die Parameter `JAAS.*`.

Hier die Standardkonfiguration, wie sie während der Installation definiert wird:

```
JAAS=${cmsroot}/conf/fs-jaas.conf
JAAS.admin=sso
JAAS.client=sso
JAAS.system=system
JAAS.websso=websso
JAAS.webnonsso=webplain
```

Zuordnung der FirstSpirit-Komponenten zu den Parameter-Namen:

- **JavaClient:** `JAAS.client`
- **Server- und Projektkonfiguration:** `JAAS.admin`
- **alle FirstSpirit-Webanwendungen (WebClient, Startseite, Server-Monitoring) mit Authentifizierung über SSO:** `JAAS.websso`
- **alle FirstSpirit-Webanwendungen (WebClient, Startseite, Server-Monitoring) ohne Authentifizierung über SSO:** `JAAS.webnonsso`
- **Access-API:** `JAAS.system`

4.3.4.9 Konfigurationsbeispiele

Standardkonfiguration:

In Verbindung mit der in Kapitel 12.2 aufgeführten Standardkonfiguration der Datei `fs-jaas.conf` ergibt sich z. B. für den JavaClient folgendes Anmeldeverfahren:

1. Beim Aufruf der FirstSpirit-Startseite über den Webbrowser wird der Benutzer zur Eingabe des Benutzernamens und Passworts aufgefordert. Diese Daten beziehen sich auf die Einträge in der FirstSpirit-Benutzerdatenbank, die über die Server- und Projektkonfiguration verwaltet wird. Nach erfolgreicher Authentifizierung wird der Berechtigungsnachweis (Ticket) generiert, der vom Webbrowser später weitergereicht wird.
2. Beim Starten des JavaClient über Web Start wird der zuvor erstellte Berechtigungsnachweis vom Webbrowser über den JavaClient an den FirstSpirit-Server zur Prüfung weitergereicht. Es ist keine weitere Passwordeingabe notwendig.
3. Falls der Berechtigungsnachweis abgelaufen war oder nicht an den FirstSpirit-Server weitergereicht werden konnte, wird als Ausweidlösung der Benutzer vom JavaClient zur Passwordeingabe aufgefordert.



Anmeldung an Windows-Domäne mit Verwendung von LDAP:

```

/* access api authentication (e.g., for remote projects) */
system {
  de.espirit.firstspirit.server.authentication.FSUserLoginModule sufficient hash="true";
  de.espirit.firstspirit.server.authentication.FSTicketLoginModule sufficient;
};

/* java-/admin-client authentication without sso */
plain {
  de.espirit.firstspirit.server.authentication.LdapLoginModule optional section="LDAP";
  de.espirit.firstspirit.server.authentication.FSUserLoginModule optional;
};

/* java-/admin-client authentication sso */
sso {
  de.espirit.firstspirit.server.authentication.FSTicketLoginModule sufficient;
  de.espirit.firstspirit.server.authentication.LdapLoginModule optional section="LDAP";
  de.espirit.firstspirit.server.authentication.FSUserLoginModule optional;
};

/* web authentication (for preview, webedit, webmonitor) without sso */
webplain {
  de.espirit.firstspirit.server.authentication.LdapLoginModule optional section="LDAP";
  de.espirit.firstspirit.server.authentication.FSUserLoginModule optional;
};

/* web authentication (for preview, webedit, webmonitor) with sso */
websso {
  de.espirit.firstspirit.server.authentication.FSTicketLoginModule sufficient;
  //de.espirit.firstspirit.server.authentication.KerberosLoginModule optional
  useFullPrincipal="false" userAgents=".*";
  de.espirit.firstspirit.server.authentication.NTLMLoginModule optional
  domains="E-SPIRIT:dc1.e-spirit.de,dc2.e-spirit.de;dc1.e-spirit.de,dc2.e-spirit.de";
  de.espirit.firstspirit.server.authentication.LdapLoginModule optional section="LDAP";
  de.espirit.firstspirit.server.authentication.FSUserLoginModule optional;
};

//enable for KerberosLoginModule only:

//com.sun.security.jgss.accept {
//  com.sun.security.auth.module.Krb5LoginModule required
//  principal="HTTP/fs5.e-spirit.de@E-SPIRIT.DE"
//  keyTab="/opt/firstspirit5/conf/krb5-fs5-HTTP.keytab"
//  useKeyTab="true"
//  storeKey="true"
//  isInitiator="false"
//  doNotPrompt="true"
//  debug="true";
//};

```

Ausschnitt aus der Datei /opt/firstspirit5/conf/fs-server.conf:

```

LDAP.NAME=e-spirit.de
LDAP.HOST_URL=ldap://dc1.e-spirit.de ldap://dc2.e-spirit.de
LDAP.SSL=FALSE
LDAP.AUTHENTICATION=SEARCH_BIND
LDAP.SEARCH.BIND_DN=ldaptechuser
LDAP.SEARCH.BIND_PASSWORD=apassword
LDAP.SEARCH.BASE_DN=DC=e-spirit,DC=de
LDAP.SEARCH.FILTER=(sAMAccountName=$USER_LOGIN$)
LDAP.IMPORT_USER=TRUE
LDAP.IMPORT_USER.GROUP_ATTRIBUTE=memberof
LDAP.IMPORT_USER.LOGIN_ATTRIBUTE=sAMAccountName
LDAP.IMPORT_USER.NAME_ATTRIBUTE=givenName,sn
LDAP.IMPORT_USER.EMAIL_ATTRIBUTE=mail
LDAP.IMPORT_USER.PHONE_ATTRIBUTE=telephoneNumber
LDAP.IMPORT_USER.ABBREVIATION_ATTRIBUTE=initials

```



4.3.5 Konfiguration Lizenz (fs-license.conf)

Im Unterverzeichnis `conf` des FirstSpirit-Servers befindet sich die Datei `fs-license.conf`. Die Datei enthält die FirstSpirit Lizenz und darf nicht verändert werden.

Über das FirstSpirit Server-Monitoring können die Lizenzparameter der `fs-license.conf` angezeigt werden (siehe Kapitel 8.6.1.2 Seite 451). Außerdem kann über das Server-Monitoring eine neue Lizenzdatei eingefügt werden. Beim Einfügen einer neuen Konfigurationsdatei `fs-license.conf` ist kein Neustart des Servers erforderlich. Die Datei wird automatisch auf dem Server aktualisiert.



Manipulationen an der `fs-license.conf` führen zu einer ungültigen Lizenz. Sollten Änderungen notwendig werden (z. B. Wechsel der IP-Adresse), wenden Sie sich bitte an den Hersteller. In einer `fs-license.conf` müssen nicht alle im Beispiel beschriebenen Informationen vorhanden sein.

```
#FIRSTspirit license
#Mon Jan 02 10:34:22 CET 2012
license.MAXPROJECTS=5
license.EXPDATE=15.12.2012
license.MODULES=personalisation,search,integration,newsletter,portal,form_edit,enterprise_search
license.ARCHIVE=1
license.VERSION=5
license.USER=e-spirit
license.WORKFLOW=1
license.DOCUMENTGROUP=1
license.MAXSESSIONS=20
license.WEBEDIT=1
license.MAXUSER=20
license.OFFICE_INTEGRATION=1
license.APPTAB_SLOTS=5
-----begin FirstSpirit license key-----
...
-----end FirstSpirit license key-----
```

`license.MAXPROJECTS`: Maximale Anzahl von Projekten, die mit dieser Lizenz auf dem Server angelegt werden dürfen. Deaktivierte Projekte werden hierbei nicht mit angerechnet.

`license.EXPDATE`: Zeitpunkt, an dem die Lizenz ausläuft. Der FirstSpirit-Server wird sich zu diesem Zeitpunkt selbstständig beenden. Sind die entsprechenden Parameter in der Konfigurationsdatei `fs-server.conf` gesetzt (siehe Kapitel 4.3.1.9 Seite 49), erfolgt vor



dem Ablaufdatum eine Warnung per E-Mail.

`license.VERSION`: Softwareversion von FirstSpirit, für die diese Lizenz gültig ist.

`license.USER`: Name des Lizenznehmers.

`license.FEATURES`: Lizenzabhängige Zusatzfunktionalitäten, die über diese Lizenz freigeschaltet werden.

`license.ARCHIVE`: Ist hier der Wert "1" gesetzt, ist die Archivfunktion aktiviert, die es ermöglicht, generierte Seiten zu archivieren (weiterführende Dokumentation siehe Kapitel 7.9 Seite 423).

`license.WORKFLOW`: Ist hier der Wert "1" gesetzt, ist die Funktionalität "Workflow" aktiviert, die es ermöglicht, Arbeitsabläufe zu erstellen, die in festgelegten Arbeitsschritten durchlaufen werden können (weiterführende Dokumentation siehe *FirstSpirit Handbuch für Entwickler – Grundlagen*).

`license.DOCUMENTGROUP`: Dieser Wert ist standardmäßig auf "1" gesetzt. Somit ist die Funktionalität "Dokumentengruppe", die es ermöglicht, Seitenreferenzen in der Struktur-Verwaltung zu einer Gruppe zusammenzufassen und so ein Ergebnisdokument, beispielsweise eine PDF-Datei, zu erzeugen standardmäßig aktiviert. Weiterführende Dokumentation siehe *FirstSpirit Online Dokumentation / Weiterführende Themen / Dokumentengruppen*.

`license.MAXSESSIONS`: Maximale Anzahl an Sitzungen, die gleichzeitig auf dem Server geöffnet werden können. Hierbei werden Server-interne Sessions (Preview, Generierung) nicht mitgezählt. Wird die Anzahl der maximalen Sitzungen überschritten, so können noch maximal zwei Serveradministrator-Sessions geöffnet werden (sofern nicht bereits 2 geöffnet wurden), um ggf. bestehende Sitzungen zu beenden.

`license.WEBEDIT`: Ist hier der Wert "1" gesetzt, ist die Funktionalität "WebClient" aktiviert, die es ermöglicht, redaktionelle Inhalte direkt innerhalb der Vorschauseite im Browserfenster zu bearbeiten.

`license.MAXUSER`: Maximale Anzahl an Benutzern, die mit dieser Lizenz auf dem Server angelegt werden dürfen.

`license.IP`: Kommaseparierte Liste der Server-IP-Adressen, für welche die Lizenz gültig ist. Wird der FirstSpirit-Server auf einem Rechner mit einer anderen IP-Adresse gestartet, ist die Lizenz ungültig.



`license.MODULES`: Lizenzparameter für Module, mehrere Module können dabei mit einem Komma voneinander getrennt werden.

`license.SCOPE`: Unterscheidung zwischen Einzel- bzw. Konzernlizenz (`license.SCOPE=SINGLE` oder `license.SCOPE=CORPORATE`)

`license.TYPE`: Angabe des Lizenz-Typs (`PRODUCTION`: "Produktiv", `DEVELOPMENT`: "Entwicklung", `DEMO`: "Demonstration", `STAGING`: "Qualitätssicherung", `TRAINING`: "Schulung"). Der Lizenz-Typ wird auf der FirstSpirit-Startseite sowie auf der Projekteinstiegsseite im FirstSpirit-JavaClient mit einem entsprechenden Logo angezeigt. Im Falle einer "Produktiv"-Lizenz wird das Projekt-Logo, das in der Server- und Projektkonfiguration für das jeweilige Projekt ausgewählt wurde (siehe Kapitel 7.4.2 Seite 285), angezeigt. Bei den anderen Lizenz-Typen wird statt dem Projekt-Logo das Lizenz-Logo angezeigt.

`license.KEY`: Lizenz-Schlüssel

`license.OFFICE_INTEGRATION`: Ist hier der Wert "1" gesetzt, können Microsoft Office-, OpenOffice- oder Google Docs-Text-Dokumente im FirstSpirit AppCenter verwendet werden.

`license.APPTAB_SLOTS`: Maximale Anzahl von Applikationsintegrationen (JavaClient und WebClient), die auf die Applikations-API zugreifen 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. (Siehe auch Dokumentation *FirstSpirit AppCenter*, Kapitel 1.5)

Für weitere Lizenzschlüssel siehe auch betreffende Modul-Dokumentationen.



4.3.6 Konfiguration Logging (fs-logging.conf)

Im Unterverzeichnis `conf` des FirstSpirit-Servers befindet sich die Datei `fs-logging.conf`, die wichtige Konfigurationseinstellungen für die "Log"-Ausgaben enthält und bei Bedarf angepasst werden muss.

Änderungen an der Konfigurationsdatei `fs-logging.conf` können über das FirstSpirit Server-Monitoring vorgenommen werden (siehe Kapitel 8.6.1.3 Seite 452). Die Änderungen werden dann in die Konfigurationsdatei geschrieben und auf dem Server aktualisiert. (Wenn Zugriff auf das Dateisystem besteht, können Änderungen an der `fs-logging.conf` auch direkt über die Konfigurationsdatei erfolgen.)

Auftretende Fehler und Infomeldungen werden dem Loggingsystem "log4j"⁸ übergeben. Über das Framework kann eine Gewichtung der Log-Ausgaben vorgenommen werden. Das Konfigurationsbeispiel zeigt beispielsweise die Kategorien `DEBUG`, `INFO` und `ERROR`. Potentiell können aber noch weitere Kategorien konfiguriert werden (z. B. `FATAL`, `WARN`). Ausnahmen bilden die beiden Stufen `ALL` und `OFF`, die das Logging entweder komplett deaktivieren (`OFF`) oder alle Meldungen ungefiltert ausgeben (`ALL`).

Im Installationsumfang von FirstSpirit sind weitere Logging-Dateien enthalten. Die Aktivierung einer bestimmten Logging-Konfiguration, sowie die Filterung und Art der Ausgabe kann zur Laufzeit über das Server-Monitoring konfiguriert werden (siehe Kapitel 8.6.1.3 Seite 452).

Konfigurationsdateien müssen der Syntax

`fs-logging.meinLogging.conf` folgen, damit sie von FirstSpirit erkannt werden und über das Server-Monitoring aktiviert werden können.

```
log4j.rootCategory=INFO, fs

log4j.logger.org.eclipse.jetty=WARN
log4j.logger.org.apache.jasper=WARN
log4j.logger.org.apache.log4j.jmx=ERROR
log4j.logger.de.espirit.FirstSpirit.server.ExecutionManagerImpl= INFO
log4j.logger.org.apache.commons.httpclient=INFO

# fs
log4j.appender.fs=de.espirit.FirstSpirit.server.logging.FSAppender
log4j.appender.fs.consoleLogging=false
log4j.appender.fs.plainLogging=false
log4j.appender.fs.datedLogging=true
log4j.appender.fs.maxFileSize=5MB
```

⁸ Weiterführende Informationen <http://logging.apache.org/log4j/docs/documentation.html>



```
log4j.appender.fs.buffer=8192
log4j.appender.fs.flushCycle=10
```

Weiterführende Informationen zum Logging-Framework "log4j" und eine Beschreibung der Parameter siehe: <http://logging.apache.org/log4j/docs/documentation.html>

Angabe von Pfaden unter Windows: Soll eine Datei für die Ausgabe der Log-Dateien angegeben werden, so ist der Pfad folgendermaßen anzugeben (Pfadangabe durch / getrennt): `log4j.appender.file.File=D:/FirstSpirit5/log/err.log`

Beispiel:

```
# file
log4j.rootCategory=ERROR, file
log4j.appender.file=org.apache.log4j.RollingFileAppender
log4j.appender.file.File=D:/FirstSpirit5/log/err.log
log4j.appender.file.MaxFileSize=5MB
log4j.appender.file.MaxBackupIndex=5
log4j.appender.file.layout=org.apache.log4j.PatternLayout
log4j.appender.file.layout.ConversionPattern=%-5p %d (%c) %m%n
```

Spezifische Parameter des FSAppenders:

Parameter zur Auswahl des Log-Verfahrens: Die im Folgenden beschriebenen Konfigurationsparameter dienen zur Auswahl des Log-Verfahrens und können unabhängig voneinander an- und ausgeschaltet werden. Erlaubte Werte sind hier 0 für "deaktiv" und 1 für "aktiv".

`log4j.appender.fs.consoleLogging`: Schalter für die Ausgabe der Log-Meldungen auf die Konsole (Kommandozeile/Shell), von der FirstSpirit gestartet wurde.

`log4j.appender.fs.plainLogging`: Schalter für die Ausgabe der Log-Meldungen als Klartext in eine Datei. Die Log-Datei für den Server hat den festen Namen `fs-server.log`.

`log4j.appender.fs.datedLogging`: Schalter für die Ausgabe der Log-Meldungen als Klartext in eine Datei. Die Log-Datei erhält hier immer den Zusatz des Datums des ersten Eintrags, also beispielsweise `fs-server.20070322_100436.log` (mit Ersteintrag vom 22. März 2007 um 10:04:36 Uhr). Dieser Parameter muss aktiviert werden, um die Suche innerhalb der Log-Dateien des Server-Monitorings zu ermöglichen.



Parameter zur Konfiguration der maximalen Dateigröße:

`log4j.appender.fs.maxFileSize`: Dieser Parameter beeinflusst die maximal erlaubte Größe für eine Log-Datei. Der Wert gibt diese Größe in Bytes an. Standardwert sind 5 MByte.

Konfiguration der Log-Rotation: Abhängig vom Log-Verfahren erfolgt bei Erreichen der maximal erlaubten Größe der Log-Datei ein Zurücksetzen oder eine Rotation.

1. Ist der Parameter `plainLogging` aktiviert und der Parameter `datedLogging` deaktiviert, so wird die aktuelle Log-Datei (z. B. `fs-server.log`) umbenannt. Sie bekommt einen mit dem Datum des ersten Eintrags versehenen Zusatz (z. B. `fs-server.20070322_100436.log`). Die umbenannte Datei wird anschließend komprimiert und erhält die zusätzliche Endung `.gz`. Währenddessen wird in eine neu angelegte Log-Datei mit dem ursprünglichen Namen weiter geloggt.
2. Ist der Parameter `plainLogging` deaktiviert und der Parameter `datedLogging` aktiviert, so wird die aktuelle Log-Datei (z. B. `fs-server.20070322_100436.log`) komprimiert und erhält die zusätzliche Endung `.gz`. Währenddessen wird weiter in eine neu angelegte Log-Datei (mit neuem Datumzusatz) geloggt.
3. Sind die Parameter `plainLogging` und `datedLogging` aktiviert, so wird parallel in die Log-Datei ohne Datumzusatz und in die Log-Datei mit Datumzusatz geloggt. Eine Rotation erfolgt wie unter 2. (die Log-Datei ohne Datumzusatz wird lediglich zurückgesetzt).

Parameter zur Konfiguration des Log-Verhaltens:

`log4j.appender.fs.buffer` Beeinflusst die intern zu benutzende Puffergröße (in Bytes) (Standardwert: 8192 bytes). Der Puffer nimmt Log-Nachrichten auf und speichert diese, bis die Puffergröße erreicht ist. Erst dann werden die Nachrichten in die Log-Datei geschrieben, um unnötige und zeitintensive Schreiboperationen zu sparen. Bei Beendigung des Servers wird der Puffer auch bei Nichterreichen der Puffergröße geschrieben.

`log4j.appender.fs.flushCycle`: Bestimmt die maximale Zeit (in Sekunden) zwischen zwei Schreibvorgängen. Ist diese Zeit abgelaufen, wird der Puffer auch bei Nichterreichen der Puffergröße geschrieben.





Neben der Datei `fs-server.log` werden u.a. für die Fehlersuche wichtige Logmeldungen auch in die Datei `fs-wrapper.log` geschrieben (siehe Kapitel 4.3.2.4 Seite 73).

4.3.7 Konfiguration des Webservers (`fs-webapp.xml`)

Für die Vorschauerstellung, die Server- und Projektkonfiguration und das Arbeiten mit dem WebClient stellt FirstSpirit einen integrierten Webserver inklusive Servlet-Engine zur Verfügung, der bei der Installation automatisch konfiguriert und aktiviert wird. Eingesetzt wird Jetty⁹.

Falls erforderlich, kann der integrierte Webserver teilweise oder vollständig durch eine andere Kombination aus Webserver und Servlet-Engine ersetzt werden, um z. B. PHP oder ASP auf Apache bzw. IIS in FirstSpirit-Projekten verwenden zu können. Dazu siehe Kapitel 4.5 Seite 108.

Vorteile des integrierten Webservers:

- Unabhängig vom Betriebssystem, da 100% Java.
- Einfache Konfiguration.
- Integration der Konfiguration projektspezifischer Webanwendungen.

Im Unterverzeichnis `conf` des FirstSpirit-Servers befindet sich die Datei `fs-webapp.xml`, die Konfigurationseinstellungen des internen Webservers enthält.

Änderungen an der Konfigurationsdatei `fs-webapp.xml` können über das FirstSpirit Server-Monitoring vorgenommen werden (siehe Kapitel 8.6.1.6 Seite 455). Die Änderungen werden dann in die Konfigurationsdatei geschrieben. Wenn Zugriff auf das Dateisystem besteht, können Änderungen an der `fs-webapp.xml` auch direkt über die Konfigurationsdatei erfolgen.

Die Konfigurationsdatei mit den bei der Installation eingetragenen Standardwerten finden Sie im Kapitel 12.3.

⁹ Weiterführende Informationen <http://www.mortbay.org/>





Damit Änderungen in der Konfigurationsdatei wirksam werden, ist ein Neustart des Webservers über das FirstSpirit Server-Monitoring erforderlich (siehe Kapitel 8.6.2.2 Seite 460).

Die Konfigurationsdatei besteht in der Standardkonfiguration nach der Installation aus folgenden Elementen, die nachfolgend beschrieben werden:

- Konnektoren
- Webanwendungen
- Protokollierung

Weitere Konfigurationen finden Sie in der Dokumentation zum Jetty: <http://docs.codehaus.org/display/JETTY/Jetty+Documentation>

4.3.7.1 Konnektoren

Als Standardeinstellung ist nur der HTTP-Konnektor aktiviert:

```
<Call name="addConnector">
  <Arg>
    <New class="org.mortbay.jetty.nio.SelectChannelConnector">
      <Set name="port"><SystemProperty name="HTTP_PORT" /></Set>
      <Set name="maxIdleTime">30000</Set>
      <Set name="Acceptors">1</Set>
      <Set name="statsOn">false</Set>
      <Set name="lowResourcesConnections">1000</Set>
      <Set name="lowResourcesMaxIdleTime">500</Set>
    </New>
  </Arg>
</Call>
```

Allgemeine Parameter aller Konnektoren:

port: TCP-Port des Konnektors. Auf Unix-Systemen kann hier nur ein Wert größer als 1024 angegeben werden, siehe auch Kapitel 4.3.7.4.

host (optional): Bindeadresse des Konnektors. Um den Konnektor nur auf bestimmten IP-Adressen eines Servers verfügbar zu machen. IP-Nr. oder Hostname kann hier angegeben werden.

MaxIdleTimeMs: Falls ein Client länger als die hier angegebene Zeit in ms untätig ist, wird die Verbindung getrennt. Das bedeutet nicht, dass sich ein FirstSpirit-Benutzer erneut anmelden muss, da die Sitzungsdaten im



Webbrowser weiterhin gültig sind, nachdem die Verbindung automatisch erneut aufgebaut wird. Zur Konfiguration der Zeitüberschreitung für die FirstSpirit-Sitzung siehe Kapitel 4.3.1.14.

Weitere mögliche Konnektoren sind AJP (Kapitel 4.5) und HTTPS (Kapitel 4.7).

4.3.7.2 Webanwendungen

Hier werden nur die FirstSpirit-internen Webanwendungen eingetragen, eine Änderung der Konfiguration ist nicht notwendig. (Benutzerdefinierte projektspezifische Webanwendungen werden in der Datei `data/server/fs-webapp-project.xml` eingetragen.)

```
<New class="org.mortbay.jetty.webapp.WebAppContext">
    <Arg><Ref id="Server" /></Arg>
    <Arg><SystemProperty name="WEBAPP_ROOT_PATH" /></Arg>
    <Arg><SystemProperty name="WEBAPP_ROOT_URL" /></Arg>
</New>

<New class="org.mortbay.jetty.webapp.WebAppContext">
    <Arg><Ref id="Server" /></Arg>
    <Arg><SystemProperty name="WEBAPP_WEBMON_PATH" /></Arg>
    <Arg><SystemProperty name="WEBAPP_WEBMON_URL" /></Arg>
</New>

<New class="org.mortbay.jetty.webapp.WebAppContext">
    <Arg><Ref id="Server" /></Arg>
    <Arg><SystemProperty name="WEBAPP_WEBEDIT5_PATH" /></Arg>
    <Arg><SystemProperty name="WEBAPP_WEBEDIT5_URL" /></Arg>
</New>

<New class="org.mortbay.jetty.webapp.WebAppContext">
    <Arg><Ref id="Server" /></Arg>
    <Arg><SystemProperty name="WEBAPP_STAGING_PATH" /></Arg>
    <Arg><SystemProperty name="WEBAPP_STAGING_URL" /></Arg>
</New>
```



4.3.7.3 Protokollierung

Die Protokollierung von Client-Zugriffen analog zu einem access.log bei Apache kann mit folgendem Eintrag aktiviert werden:

```
<Call name="addHandler">
  <Arg>
    <New class="org.mortbay.jetty.handler.RequestLogHandler">
      <Set name="requestLog">
        <New id="RequestLogImpl" class="org.mortbay.jetty.NCSARequestLog">
          <Arg><SystemProperty name="cmsroot" />/log/fs-access_YYYY_MM_DD.log</Arg>
          <Set name="retainDays">31</Set>
          <Set name="append">true</Set>
          <Set name="extended">true</Set>
        </New>
      </Set>
    </New>
  </Arg>
</Call>
```

4.3.7.4 Verwendung von Standard-Portnummern unter Unix

Auf Unix-Systemen können für die Konnektoren nur Portnummern größer als 1024 angegeben werden, da der FirstSpirit-Server nicht als `root` gestartet wird. Um die Standardwerte 80 für http oder 443 für https verwenden zu können, muss daher entweder eine Weiterleitung des TCP-Ports konfiguriert oder ein zusätzlicher externer Webserver eingesetzt werden (Kapitel 4.5).

Eine Weiterleitung des TCP-Ports, hier im Beispiel von 80 nach 8000, kann unter Linux mit folgendem Aufruf zur lokalen Firewall-Konfiguration durchgeführt werden:

```
iptables -t nat -I PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 8000
```

Auf anderen Unix-Systemen kann ebenfalls eine interne Firewall zur Weiterleitung verwendet werden oder alternativ `rinetd`¹⁰.

¹⁰ `rinetd`: <http://www.boutell.com/rinetd/>



4.4 Anbindung an einen LDAP-Server

4.4.1 Authentifizierung über LDAP

Es gibt in FirstSpirit die Möglichkeit die Authentifizierung eines Benutzers mithilfe eines LDAP¹¹-Servers durchzuführen. Ein Benutzer bekommt ein Flag zugewiesen, welches anzeigt, ob er ein externer Benutzer ist oder nicht. Im FirstSpirit-Server lassen sich verschiedene LDAP-Konfigurationen (sogenannte Sections) anlegen (siehe Kapitel 4.3.4 Seite 77) und konfigurieren (siehe Kapitel 4.3.1.10 Seite 50). Ein LDAP-Benutzer wird dabei mit genau einer Section assoziiert (siehe 7.2.4.2 Seite 225). Die Authentifizierung kann potentiell auf drei unterschiedliche Arten erfolgen:

1. **LDAP Bind:** Name und Passwort werden an den LDAP-Server geschickt. Dazu muss der "Distinguished Name" (DN), das heißt der eindeutige Schlüssel zur Identifizierung des Benutzers innerhalb des LDAP-Servers bekannt sein. Falls der DN existiert, wird das übergebene Passwort mithilfe der "Bind"-Operation geprüft. Ein Beispiel für die Verwendung von LDAP Bind findet sich in Kapitel 4.3.1.10 Seite 50.
2. **LDAP Search & Bind:** Falls der "Distinguished Name" (DN) eines Benutzers nicht bekannt ist, kann man ihn innerhalb eines Teilbaums des LDAP-Servers suchen. Dazu muss ein Suchfilter und ein Startknoten definiert werden. Beispiel:

```
SEARCH.FILTER=(cn=$USER_LOGIN$)
SEARCH.BASE_DN=dc=mycompany,dc=com
```

Dieser Filter sucht nach allen Einträgen im LDAP-Baum, bei denen das Attribut "cn" gleich dem eingegebenen Login-Namen ist. Startknoten ist der Knoten mit dem DN "dc=mycompany,dc=com". Wird ein solcher Knoten gefunden, wird ein "Bind" ausgeführt. (siehe LDAP Bind).

3. **LDAP Search & Compare:** (Diese Option funktioniert äquivalent zu Punkt 2.) Allerdings wird, nachdem ein passender Knoten gefunden worden ist, keine "Bind"-Operation durchgeführt. Stattdessen wird das eingegebene Passwort mit einem beliebigen LDAP-Attribut verglichen.

Beispiel:

```
SEARCH.COMPARE.PASSWORD_ATTRIBUTE_NAME=mail
```

¹¹ LDAP (Lightweight Directory Access Protocol)



In diesem Fall muss das eingegebene Passwort mit dem Inhalt des "mail"-Attributes des LDAP-Knotens übereinstimmen.

Bei erfolgreicher LDAP-Authentifizierung wird der Benutzer, falls er bisher nicht auf dem FirstSpirit-System bekannt ist, als externer Benutzer auf dem FirstSpirit-Server angelegt (siehe Kapitel 7.4.7.2 Seite 302). Dazu muss der Konfigurationsparameter `JAAS.autoCreateUser` auf den Wert `true` konfiguriert sein (Standardeinstellung) (vgl. Kapitel 4.3.4.8 Seite 91). Der Login des externen LDAP-Benutzers wird dabei automatisch nach FirstSpirit übernommen. Das Passwort eines **externen LDAP-Benutzers** wird aber nach der erstmaligen erfolgreichen Authentifizierung in FirstSpirit geleert. Logins mit einem leeren Passwort werden von FirstSpirit-Server abgewiesen. Ein externer LDAP-Benutzer kann sich somit nur am FirstSpirit-Server anmelden, wenn der LDAP-Server während des Anmeldevorgangs erreichbar ist. Bei **internen LDAP-Benutzern** bleibt das Passwort in FirstSpirit erhalten. Der Benutzer kann sich in diesem Fall sowohl mit dem FirstSpirit-Passwort als auch mit dem LDAP-Passwort anmelden.

Authentifizierung auch dann auf dem FirstSpirit-Server zu authentifizieren, wenn der LDAP-Server nicht erreichbar war. Dieses Verhalten führte aber dazu, dass Passwortänderungen im LDAP nicht an FirstSpirit weitergereicht wurden. Ab FirstSpirit Version 4.1.33 wird das Passwort eines **externen LDAP-Benutzers** daher nach erfolgreicher Authentifizierung in FirstSpirit geleert. Logins mit einem leeren Passwort werden von FirstSpirit-Server abgewiesen. Ein externer LDAP-Benutzer kann sich somit nur noch am FirstSpirit-Server anmelden, wenn der LDAP-Server während des Anmeldevorgangs erreichbar ist. Bei **internen LDAP-Benutzern** bleibt das Passwort in FirstSpirit erhalten. Der Benutzer kann sich in diesem Fall sowohl mit dem FirstSpirit-Passwort als auch mit dem LDAP-Passwort anmelden.

Eine Sonderrolle bei der LDAP-Anmeldung hat der **Serveradministrator** (Login: Admin). Dieser Benutzer wird bei der Installation des FirstSpirit-Servers automatisch angelegt. Das Passwort des Serveradministrators wird nie geleert, unabhängig davon ob er als "externer" oder "interner" Benutzer konfiguriert wurde. Es wird daher empfohlen, das Passwort des Serveradministrators (initial: "Admin") direkt nach der Installation des FirstSpirit-Servers zu ändern.



4.4.2 LDAP-Attribute an einen FirstSpirit Benutzer binden

Neben der reinen Authentifizierung ist es möglich, beliebige LDAP-Attribute an die Benutzerattribute eines CMS-Benutzers zu binden. Dazu muss der Parameter `LDAP.IMPORT_USER` in der Konfigurationsdatei `fs-server.conf` auf den Wert `TRUE` gesetzt sein (siehe Kapitel 4.3.1.10 Seite 50).

Außerdem muss innerhalb der `fs-server.conf` eine Attribut-Zuordnung erfolgen:

```
LDAP.IMPORT_USER.<cms-attribut>_ATTRIBUTE=<ldap-attribut->
```

Alle so definierten LDAP-Attribute werden beim ersten Login des jeweiligen Benutzers automatisch importiert.

Sollen mehrere Attribute aus dem LDAP-Server auf ein Attribut in FirstSpirit abgebildet werden, so müssen die einzelnen Attribute per Komma (,) getrennt werden.

Über den Parameter

```
LDAP.MULTI_VALUE_SEPARATOR=[Trennzeichen]
```

kann das Trennzeichen definiert werden, mit dem die ausgelesenen Attributwerte getrennt werden. Mit der Konfiguration

```
LDAP.MULTI_VALUE_SEPARATOR=:  
LDAP.IMPORT_USER.NAME_ATTRIBUTE=givenName,sn
```

können z. B. Vor- und Nachname mit einem Doppelpunkt getrennt ausgegeben werden.

Folgende CMS-Benutzerattribute können beim Login durch LDAP-Attribute überschrieben werden (Konfigurationsbeispiel siehe Kapitel 4.3.1.10 Seite 50):

- **Benutzername:** Name des FirstSpirit Benutzers.
- **E-Mail:** E-Mail-Adresse des FirstSpirit Benutzers.
- **Telefon:** Telefonnummer des FirstSpirit Benutzers.
- **Kürzel:** Kürzel des FirstSpirit Benutzers.



4.4.3 Verwendung von TLS oder SSL

Falls der FirstSpirit-Server über TLS/SSL den LDAP-Server anbinden soll, muss zunächst das Zertifikat des LDAP-Servers in den Zertifikatsspeicher von FirstSpirit importiert werden. Dazu wird das Java-Werkzeug `keytool` aus dem "bin"-Verzeichnis des JDKs genutzt. Falls das Zertifikat in einem anderen Format vorliegt, kann es über das externe Dienstprogramm `openssl`¹² in das von `keytool` importierbare PEM-Format konvertiert werden. Beispielaufruf zur Konvertierung

```
openssl x509 -inform DER -in mycompany.der -outform PEM -out mycompany.crt
```

Liegt das Zertifikat z. B. in der Datei "mycompany.crt" und wurde zuvor in das Installationsverzeichnis des FirstSpirit-Servers gewechselt, so lässt es sich folgendermaßen in den Zertifikatsspeicher importieren:

```
keytool -import -file mycompany.crt -alias ldapserver.mydomain.net -keystore conf/fs-truststore.jks -storepass changeit
```

Zusätzlich müssen Pfad und Passwort des Zertifikatsspeichers als Java-Parameter in `fs-wrapper.conf` (Kapitel 4.3.2) eingetragen werden:

```
wrapper.java.additional.X=-Djavax.net.ssl.trustStore=conf/fs-truststore.jks  
wrapper.java.additional.Y=-Djavax.net.ssl.trustStorePassword=changeit
```

Für Testinstallationen kann ein selbstsigniertes Zertifikat verwendet werden, das folgendermaßen erstellt wird, nachdem in das Installationsverzeichnis des FirstSpirit-Servers gewechselt wurde:

```
keytool -genkey -alias ldapserver.mydomain.net -keyalg RSA -validity 1000 -  
keystore conf/fs-keystore.jks -storepass changeit
```

Bei der Abfrage nach "Vor- und Nachname" muß der vollqualifizierte Hostname (Hostname inkl. Domain) angegeben werden.

Nach dem Neustart des FirstSpirit-Servers kann die Kommunikation zum LDAP-Server über TLS oder SSL erfolgen.

¹² <http://www.openssl.org/>



4.5 Einbinden in externen Webserver

Als HTTP-Server und Servlet-Engine wird im FirstSpirit-Server standardmäßig Jetty verwendet. Sollen in FirstSpirit-Projekten spezielle, serverseitige Implementierungen verwendet werden (z. B. PHP oder ASP), die von Jetty nicht ausgewertet werden können, muss zusätzlich ein externer Webserver eingebunden werden. Die FirstSpirit-Webanwendungen im internen Jetty-Webserver leiten dann die HTTP-Anfragen für spezielle Dateitypen über HTTP an den externen Webserver weiter.

In diesem Kapitel wird die Verwendung des Apache HTTP-Servers in Kombination mit PHP beschrieben. Andere Webserver können nach dem gleichen Prinzip eingebunden werden, sofern sie eine Weiterleitung über HTTP oder AJP an die Servlet-Engine ermöglichen.

Der in FirstSpirit 5 integrierte Webserver Jetty 8 bietet als Standards Servlet 3.0 und JSP 2.1 und ist nicht vollständig kompatibel zu AJP. Falls in FirstSpirit-Projekten Webanwendungen den vollständigen J2EE-Standard benötigen oder AJP zur Anbindung eines HTTP-Servers oder Loadbalancers, bietet es sich daher an, Apache Tomcat zu verwenden. Dazu siehe Kapitel 4.5.2.

Eine weitere Konfigurationsmöglichkeit ist die Lastverteilung auf mehrere Servlet-Engines (siehe Kapitel 4.5.5 Seite 129).

4.5.1 Apache HTTP-Server mit Servlet-Engine Jetty

Der Apache HTTP-Server in Version 2.2 oder 2.4 wird in dieser Konfiguration in Verbindung mit der in FirstSpirit integrierten Servlet/JSP-Engine des Jetty-Webserver verwendet. Da der Jetty-Webserver nicht vollständig kompatibel zu AJP 1.3 ist, muss in diesem Fall HTTP als Protokoll zwischen Apache und Jetty eingesetzt werden.

Zur Realisierung der HTTP-Verbindung zur Servlet-Engine wird `mod_proxy_http`¹³ eingesetzt.

Die Konfigurationsumgebung des Apache ist abhängig vom Betriebssystem und wird meistens über mehrere Konfigurationsdateien verteilt. Eine übliche Konvention ist, für allgemeine Parameter die Datei `/etc/apache2/httpd.conf` zu verwenden, für

¹³ http://httpd.apache.org/docs/2.2/mod/mod_proxy.html



Modulkonfiguration das Verzeichnis `/etc/apache2/mods-available` und für jeden virtuellen Webserver eine Datei unter `/etc/apache2/sites-available`.

Für FirstSpirit sollte ein eigener virtueller Webserver verwendet werden, der über folgende Einträge so konfiguriert wird, dass HTTP-Anfragen für die FirstSpirit-Webanwendungen über `mod_proxy_http` an Jetty weitergeleitet werden. Von den unter Jetty installierten FirstSpirit-Webanwendungen wird dann bei Bedarf automatisch zusätzlich eine interne HTTP-Verbindung zur Auslieferung spezieller Dateitypen wie PHP oder ASP aufgebaut.

Bevor Apache konfiguriert wird, sollte die FirstSpirit-Jetty-Konfiguration überprüft werden. In der Datei `firstspirit5/conf/fs-server.conf` dürfen die Parameter `WEBAPP_ROOT_PATH` und `WEBAPP_ROOT_URL` nicht eingetragen sein und es muss folgender Parameter definiert sein:

```
INTERNAL_SERVLET_ENGINE=1
```

Es folgt nun die Apache-Konfiguration:

In dem Konfigurationsbeispiel müssen bei den mit "EXAMPLE" markierten Zeilen jeweils die Pfade und Adressen an die lokale Konfiguration angepasst werden. Der Eintrag `"# EXAMPLE"` muss anschließend entfernt werden, ansonsten werden Syntaxfehler beim Start des Webservers angezeigt.

Nach erfolgter Konfiguration und Neustart des Apache-Servers ist die FirstSpirit-Startseite über Apache erreichbar, beispielsweise `http://fs5.yourdomain.net`.

Modulkonfiguration PHP:

```
LoadModule php5_module modules/libphp5.so # EXAMPLE
AddType application/x-httpd-php .php .phtml
AddType application/x-httpd-php-source .phps
```

Modulkonfiguration `mod_proxy_http`:

```
LoadModule proxy_module modules/mod_proxy.so # EXAMPLE
LoadModule proxy_http_module modules/mod_proxy_http.so # EXAMPLE
LoadModule rewrite_module modules/mod_rewrite.so # EXAMPLE
LoadModule proxy_balancer_module modules/mod_proxy_balancer.so # EXAMPLE

ProxyRequests Off
<Proxy *>
  AddDefaultCharset off
  order deny,allow
  deny from all
</Proxy>
```



Virtueller Webserver:

```

<VirtualHost *:80>
ServerName fs5.yourdomain.net           # EXAMPLE
ServerAlias fs5                         # EXAMPLE

LogLevel warn
CustomLog /var/log/apache2/fs5.access.log combined # EXAMPLE
ErrorLog /var/log/apache2/fs5.error.log # EXAMPLE

ServerSignature off
UseCanonicalName off
AddDefaultCharset off
ProxyRequests off
RewriteEngine on
ProxyPreserveHost on

DocumentRoot /opt/firstspirit5/web      # EXAMPLE

DirectoryIndex index.html index.jsp index.php

# Protect configuration files.
<LocationMatch "\.htaccess|WEB-INF/">
    order deny,allow
    deny from all
</LocationMatch>

# Protect FirstSpirit previews, to be accessible
# from Servlet-Engine only, not from Web-Browser.
<LocationMatch preview_cache>
    order deny,allow
    deny from all
    # All LAN addresses where Servlet-Engine is connecting from:
    allow from 127.0.0.1
    allow from 10.11.12.13 # EXAMPLE
    allow from 172.111.12.13 # EXAMPLE
</LocationMatch>

# status monitor for mod_proxy and balancer
<Location /balancer-manager>
    SetHandler balancer-manager
    order deny,allow
    deny from all
    # allow access from administration network only
    allow from 192.168.1. # EXAMPLE
</Location>

<Proxy balancer://fshttp>
    # set to hostname of FirstSpirit Server (Jetty)
    # and to port given by HTTP_PORT in fs-server.conf
    BalancerMember http://localhost:8000 retry=10 # EXAMPLE
</Proxy>

# forward requests for FirstSpirit-Webapps to Servlet-Engine
RewriteCond %{REQUEST_URI} !^/balancer-manager
RewriteCond %{REQUEST_URI} !^/server-status
RewriteCond %{REQUEST_URI} !^/fs5preview(_[0-9]+)?/preview_cache
RewriteCond %{REQUEST_URI} !^/fs5webedit(_[0-9]+)?/preview_cache
RewriteRule ^/(.*) balancer://fshttp/$1 [proxy,last]

</VirtualHost>

```



Falls für Tests oder zur Administration die Startseite direkt über den Jetty unter Umgehung des Apache-httpd angezeigt werden soll, muss im Browser als Startseite <http://fs5server:8000> eingegeben werden. Der Port 8000 entspricht dabei dem bei `HTTP_PORT` in `fs-server.conf` eingetragenen Port des Jetty-Servers.

Zur Verwendung von https in der Kombination Apache-httpd und Jetty sind folgende Änderungen notwendig:

In der Apache-Konfiguration müssen folgende Parameter ergänzt werden:

```
SSLEngine on
SSLProxyEngine on
SSLCertificateFile /etc/ssl/certs/mydomain.pem
```

Der bei `SSLCertificateFile` angegebene Dateipfad muss zu einem gültigen TLS/SSL-Zertifikat zeigen.

Die Zeile

```
BalancerMember http://localhost:8000 retry=10
```

ändern in

```
BalancerMember https://localhost:8443 retry=10
```

In der Datei `firstspirit5/conf/fs-webapp.xml` den Bereich "HTTPS-Connector" aktivieren und dort bei `port` ebenfalls 8443 eintragen. Das mitgelieferte selbstsignierte und bereits in der Konfiguration eingetragene Zertifikat im Keystore `conf/fs-keystore.jks` kann für diese Konfiguration problemlos verwendet werden, weil der Jetty-HTTPS-Connector nur intern zwischen Apache-httpd und Jetty verwendet wird.

4.5.2 Apache HTTP-Server mit Servlet-Engine Tomcat

Der Apache HTTP-Server in Version 2.2.22¹⁴ oder neuer wird in dieser Konfiguration in Verbindung mit der Servlet-Engine Tomcat 6 oder 7 verwendet. Als Protokoll zwischen Apache und Tomcat wird AJP eingesetzt.

¹⁴Wegen eines Fehlers in `mod_proxy_ajp`, der erst mit Version 2.2.22 behoben wurde, wird für FirstSpirit die Verwendung des Apache httpd in Version 2.2.22 oder 2.4 oder neuer empfohlen. Bei Verwendung der fehlerhaften Version des `mod_proxy_ajp` verursacht ein einzelner langsam bearbeiteter `http-request`, der den in `httpd.conf` definierten `Timeout`-Wert überschreitet, das Abtrennen des gesamten Tomcat-Workers. Durch Verwendung eines genügend großen Werts für Parameter "timeout", z. B. 1200s, wie in den folgenden Konfigurationstexten, kann das Problem in älteren Versionen des `mod_proxy_ajp` weitgehend umgangen werden.



Der in FirstSpirit 5 integrierte Webserver Jetty 8 bietet als Standards Servlet 3.0 und JSP 2.1 und ist nicht vollständig kompatibel zu AJP. Falls in FirstSpirit-Projekten Webanwendungen den vollständigen J2EE-Standard benötigen oder AJP zur Anbindung eines HTTP-Servers oder Loadbalancers, bietet es sich daher an, Apache Tomcat zu verwenden.

Zurzeit existieren als AJP-Connector für Apache `mod_jk` und `mod_proxy_ajp`¹⁵. In diesem Kapitel wird die Verwendung von `mod_proxy_ajp` beschrieben, da dieses Modul seit Version 2.2 zum Lieferumfang gehört und somit die Installation gegenüber `mod_jk` vereinfacht. Eine Lastverteilung (Loadbalancing) auf mehrere Servlet-Engines kann mit beiden Modulen realisiert werden (siehe Kapitel 4.5.5 Seite 129).

Die Konfigurationsumgebung des Apache ist abhängig vom Betriebssystem und wird meistens über mehrere Konfigurationsdateien verteilt. Eine übliche Konvention ist, für allgemeine Parameter die Datei `/etc/apache2/httpd.conf` zu verwenden, für Modulkonfiguration das Verzeichnis `/etc/apache2/mods-available` und für jeden virtuellen Webserver eine Datei unter `/etc/apache2/sites-available`.

Für FirstSpirit sollte ein eigener virtueller Webserver verwendet werden, der über folgende Einträge so konfiguriert wird, dass HTTP-Anfragen für die FirstSpirit-Webanwendungen über `mod_proxy_ajp` an Tomcat weitergeleitet werden. Von den unter Tomcat installierten FirstSpirit-Webanwendungen wird dann bei Bedarf automatisch zusätzlich eine interne HTTP-Verbindung zur Auslieferung spezieller Dateitypen wie PHP oder ASP aufgebaut.

In dem Konfigurationsbeispiel müssen bei den mit "EXAMPLE" markierten Zeilen jeweils die Pfade und Adressen an die lokale Konfiguration angepasst werden. Der Eintrag "# EXAMPLE" muss anschließend entfernt werden, ansonsten werden Syntaxfehler beim Start des Webserver angezeigt.

Modulkonfiguration PHP:

```
LoadModule php5_module modules/libphp5.so # EXAMPLE
AddType application/x-httpd-php .php .html
AddType application/x-httpd-php-source .phps
```

¹⁵ http://httpd.apache.org/docs/2.2/mod/mod_proxy.html



Modulkonfiguration mod_proxy_ajp:

```

LoadModule proxy_module modules/mod_proxy.so # EXAMPLE
LoadModule proxy_ajp_module modules/mod_proxy_ajp.so # EXAMPLE
LoadModule rewrite_module modules/mod_rewrite.so # EXAMPLE
LoadModule proxy_balancer_module modules/mod_proxy_balancer.so # EXAMPLE

ProxyRequests Off
<Proxy *>
  AddDefaultCharset off
  order deny,allow
  deny from all
</Proxy>

```

Virtueller Webserver:

```

<VirtualHost *:80>
  ServerName fs5.yourdomain.net # EXAMPLE
  ServerAlias fs5 # EXAMPLE

  LogLevel warn

  LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"
  \"%{COOKIE}i\" %{BALANCER_WORKER_ROUTE}e %D" route_and_requesttime

  CustomLog /var/log/apache2/fs5.access.log route and requesttime # EXAMPLE
  ErrorLog /var/log/apache2/fs5.error.log # EXAMPLE

  ServerSignature off
  UseCanonicalName off
  AddDefaultCharset off
  ProxyRequests off
  RewriteEngine on

  DocumentRoot /opt/firstspirit5/web # EXAMPLE

  DirectoryIndex index.html index.jsp index.php

  # Protect configuration files.
  <LocationMatch "\.htaccess|/WEB-INF/">
    order deny,allow
    deny from all
  </LocationMatch>

  # Protect FirstSpirit previews, to be accessible
  # from Servlet-Engine only, not from Web-Browser.
  <LocationMatch preview cache>
    order deny,allow
    deny from all
    # All LAN addresses where Servlet-Engine is connecting from:
    allow from 127.0.0.1
    allow from 10.11.12.13 # EXAMPLE
    allow from 172.111.12.13 # EXAMPLE
  </LocationMatch>

  # status monitor for mod_proxy and balancer
  <Location /balancer-manager>
    SetHandler balancer-manager
    order deny,allow
    deny from all
    # allow access from administration network only
    allow from 192.168.1. # EXAMPLE
  </Location>

  <Proxy balancer://fsajp>
    BalancerMember ajp://localhost:8009 retry=10 connectiontimeout=10 ping=5 ttl=1800
    timeout=1200 # EXAMPLE
  </Proxy>

  # forward requests for FirstSpirit-Webapps to Servlet-Engine

```

```
RewriteCond %{REQUEST_URI} !^/balancer-manager
RewriteCond %{REQUEST_URI} !^/manager/
RewriteCond %{REQUEST_URI} !^/server-status
RewriteCond %{REQUEST_URI} !^/fs5preview(_[0-9]+)?/preview_cache
RewriteCond %{REQUEST_URI} !^/fs5webedit(_[0-9]+)?/preview_cache
RewriteRule ^/(.*) balancer://fsajp/$1 [proxy,last]

</VirtualHost>
```

Eine Konfiguration für eine Tomcat Servlet-Engine, die auf demselben Host wie der FirstSpirit Server betrieben wird, ist im nächsten Kapitel (4.5.3) beschrieben und eine Konfiguration für eine Tomcat Servlet-Engine, die auf einem eigenen Host betrieben wird in Kapitel 4.5.4.

4.5.3 Servlet-Engine Tomcat

Der in FirstSpirit 5 integrierte Webserver Jetty 8 bietet als Standards Servlet 3.0 und JSP 2.1 und ist nicht vollständig kompatibel zu AJP. Falls in FirstSpirit-Projekten Webanwendungen den vollständigen J2EE-Standard benötigen oder AJP zur Anbindung eines HTTP-Servers oder Loadbalancers, bietet es sich an, Apache Tomcat 6¹⁶ oder 7 zu verwenden, dessen Konfiguration für FirstSpirit im folgenden Abschnitt beschrieben wird.

Tomcat kann auf demselben Host wie der FirstSpirit-Server betrieben werden (dieses Kapitel) oder auf einem eigenen Host (Kapitel 4.5.4) oder zur Lastverteilung auf mehreren Host (Kapitel 4.5.5).

Tomcat wird in dieser Konfiguration auf demselben Host wie der FirstSpirit-Server betrieben und benötigt lesenden und schreibenden Zugriff auf das Verzeichnis firstspirit5/web. Weil gleichzeitig der Prozess des FirstSpirit-Server Lese- und Schreibzugriff auf demselben Ordner benötigt, muss Tomcat unter demselben Benutzerkonto betrieben werden, wie der FirstSpirit-Server, also in der Standardinstallation unter Benutzerkonto "fs5".

Falls aus Sicherheitsgründen unbedingt Tomcat unter einem anderen Benutzerkonto betrieben werden muss, sollte die im Kapitel 4.5.4 beschriebene Variante verwendet werden, die auch auf demselben Host einsetzbar ist.

¹⁶ <http://tomcat.apache.org>. Ein Fehler (https://issues.apache.org/bugzilla/show_bug.cgi?id=50700) in den Tomcat-Versionen 6.0.30 bis 6.0.32 verhindert das Auslesen der Kontext-Parameter, die in der Datei `context.xml` für FirstSpirit notwendig sind. Es sollte also entweder Tomcat 6.0.29 oder mindestens 6.0.33 oder 7.0 verwendet werden.



Tomcat kann entweder eigenständig mit dem in Tomcat integrierten HTTP-Server betrieben werden oder zusätzlich um den Apache HTTP-Server ergänzt werden, wie in Kapitel 4.5.2 beschrieben.

Bevor Tomcat konfiguriert wird, ist die FirstSpirit-Konfiguration zu ändern. In der Datei `firstspirit5/conf/fs-server.conf` müssen folgende Parameter definiert werden, um Jetty zu deaktivieren und die Standard-Webanwendung zu definieren, die beim Aufruf der FirstSpirit-Startseite über Tomcat, beispielsweise `http://fs5.yourdomain.net:8080`, verwendet wird:

```
INTERNAL_SERVLET_ENGINE=0
WEBAPP_ROOT_PATH=${WEB_DIR}/ROOT
```

Anschließend den FirstSpirit-Server herunterfahren, das Verzeichnis `firstspirit5/web/fs5root` entfernen und den FirstSpirit-Server wieder starten.

Die Installation von Tomcat, Version 6 oder 7, verwendet wird "Binary Distribution Core" von <http://tomcat.apache.org>, erfolgt in diesem Beispiel nach `/opt/firstspirit5/tomcat`. Falls die Installation in ein anderes Verzeichnis erfolgt, muss im nachfolgenden Konfigurationsbeispiel an allen Stellen die relative Pfadangabe `"${catalina.home}/.."` durch die absolute Angabe `"/opt/firstspirit5/"` ersetzt werden.

In `tomcat/conf/server.xml` den Eintrag `appBase` bei `<Host>` auf das Verzeichnis `web` der FirstSpirit-Installation ändern:

```
<Host name="localhost" appBase="${catalina.home}/../web" unpackWARs="true"
autoDeploy="true" xmlValidation="false" xmlNamespaceAware="false">
```

In `tomcat/conf/server.xml` die Kodierung für URI-Parameter auf UTF-8 einstellen, dazu bei den bereits vorhandenen HTTP- und AJP-Konnektoren den Parameter `URIEncoding` hinzufügen:

```
<Connector port="8080" protocol="HTTP/1.1" URIEncoding="UTF-8" />
<Connector port="8009" protocol="AJP/1.3" URIEncoding="UTF-8" />
```

In der Datei `tomcat/conf/web.xml` das Überprüfen und Kompilieren der JSP-Dateien bei jeder Änderung ohne Verzögerung aktivieren. Dazu die folgenden bereits vorhandenen Zeilen der Datei

```
<servlet>
  <servlet-name>jsp</servlet-name>
  <servlet-class>org.apache.jasper.servlet.JspServlet</servlet-class>
  <init-param>
    <param-name>fork</param-name>
    <param-value>>false</param-value>
```



```

</init-param>
<init-param>
  <param-name>xpoweredBy</param-name>
  <param-value>>false</param-value>
</init-param>
<load-on-startup>3</load-on-startup>
</servlet>

```

durch folgende Zeilen ersetzen:

```

<servlet>
  <servlet-name>jsp</servlet-name>
  <servlet-class>org.apache.jasper.servlet.JspServlet</servlet-class>
  <init-param>
    <param-name>fork</param-name>
    <param-value>>false</param-value>
  </init-param>
  <init-param>
    <param-name>xpoweredBy</param-name>
    <param-value>>false</param-value>
  </init-param>
  <init-param>
    <param-name>development</param-name>
    <param-value>true</param-value>
  </init-param>
  <init-param>
    <param-name>modificationTestInterval</param-name>
    <param-value>0</param-value>
  </init-param>
  <load-on-startup>3</load-on-startup>
</servlet>

```

Falls die Möglichkeit des Auflistens der Verzeichnisinhalte zur Entwicklung von Webanwendungen im Staging-Bereich des FirstSpirit-Servers benötigt wird, in derselben Datei die Zeilen

```

<servlet>
  <servlet-name>default</servlet-name>
  <servlet-class>org.apache.catalina.servlets.DefaultServlet</servlet-class>
  <init-param>
    <param-name>debug</param-name>
    <param-value>0</param-value>
  </init-param>
  <init-param>
    <param-name>listings</param-name>
    <param-value>>false</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>

```

durch folgende Zeilen ersetzen:

```

<servlet>
  <servlet-name>default</servlet-name>
  <servlet-class>org.apache.catalina.servlets.DefaultServlet</servlet-class>
  <init-param>
    <param-name>debug</param-name>
    <param-value>0</param-value>
  </init-param>
  <init-param>
    <param-name>listings</param-name>

```



```
<param-value>true</param-value>
</init-param>
<load-on-startup>1</load-on-startup>
</servlet>
```

Um den Tomcat-Manager zur Status-Überwachung zu aktivieren, in der Datei `tomcat/conf/tomcat-users.xml` ein Benutzerkonto für die Rolle "manager" des Tomcat-Managers einfügen:

Tomcat 6:

```
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>
  <role rolename="manager"/>
  <user username="Admin" password="tomcat-password" roles="manager"/>
</tomcat-users>
```

Tomcat 7:

```
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>
  <role rolename="manager-gui"/>
  <role rolename="manager-script"/>
  <user username="Admin" password="tomcat-password" roles="manager-gui"/>
</tomcat-users>
```

Die Datei `tomcat/conf/Catalina/localhost/manager.xml` mit folgendem Inhalt zum Aktivieren des Tomcat-Managers anlegen:

```
<Context docBase="${catalina.home}/webapps/manager"
  privileged="true" antiResourceLocking="false"
  antiJARLocking="false">

  <ResourceLink name="users" global="UserDatabase"
    type="org.apache.catalina.UserDatabase"/>
</Context>
```

Die vorhandene Datei `tomcat/conf/context.xml`¹⁷ durch folgenden Inhalt ersetzen, um das sofortige Prüfen auf Dateiänderungen (HTML) zu aktivieren. Über den bei "firstspirit.host" angegebenen Hostnamen oder die IP-Adresse muss von Tomcat aus der FirstSpirit-Server im lokalen Netz erreichbar sein. Die Angabe bei "firstspirit.port" muss mit dem Port des FirstSpirit-Servers, wie in `firstspirit5/conf/fs-server.conf` über Parameter `SOCKET_PORT` definiert, übereinstimmen. Zusätzlich wird das Speichern persistenter Session-Informationen deaktiviert, weil die FirstSpirit-Webanwendungen dies nicht unterstützen:

¹⁷ Ein Fehler (https://issues.apache.org/bugzilla/show_bug.cgi?id=50700) in den Tomcat-Versionen 6.0.30 bis 6.0.32 verhindert das Auslesen der Kontext-Parameter, die in der Datei `context.xml` für FirstSpirit notwendig sind. Es sollte also entweder Tomcat 6.0.29 oder mindestens 6.0.33 oder 7.0 verwendet werden.



```
<?xml version='1.0' encoding='utf-8'?>
<Context allowLinking="true" cachingAllowed="false" useHttpOnly="true">

  <WatchedResource>WEB-INF/web.xml</WatchedResource>

  <!-- disable session persistence across Tomcat restarts -->
  <Manager pathname="" />

  <Parameter name="firstspirit.host" value="fs5server" override="false" />
  <Parameter name="firstspirit.port" value="1088" override="false" />

</Context>
```

In `tomcat/conf/catalina.properties` den Eintrag zu `"common.loader"` um `fs-webrt.jar` und `shared/lib` der FirstSpirit-Installation erweitern (**alles ohne Leerzeichen in eine Zeile schreiben**):

```
common.loader=${catalina.home}/lib,${catalina.home}/lib/*.jar,
  ${catalina.home}/../data/fslib/fs-webrt.jar,
  ${catalina.home}/../shared/lib/*.jar
```

Soll statt `${catalina.home}` unter Windows ein anderer Pfad verwendet werden, muss folgende kleingeschriebene Schreibweise der Laufwerksbuchstaben verwendet werden (**alles ohne Leerzeichen in eine Zeile schreiben**):

```
common.loader=${catalina.home}/lib,${catalina.home}/lib/*.jar,
d:/Programme/FirstSpirit5/data/fslib/fs-webrt.jar,
d:/Programme/FirstSpirit5/shared/lib/*.jar
```

Die Datei `tomcat/lib/log4j.properties` mit folgendem Inhalt erstellen, um das Logging der FirstSpirit-Webanwendungen in eine eigene Datei umzuleiten:

```
log4j.rootCategory=INFO, fs

# change INFO in the following line to DEBUG
# for detailed FirstSpirit logging:
log4j.logger.de.espirit=INFO

log4j.logger.org.eclipse.jetty=WARN
log4j.logger.org.apache.catalina=INFO
log4j.logger.org.apache.jasper=WARN
log4j.logger.org.apache.log4j.jmx=ERROR
log4j.logger.org.apache.commons.httpclient=INFO

log4j.appender.fs=org.apache.log4j.RollingFileAppender
log4j.appender.fs.File=${catalina.home}/logs/firstspirit.log
log4j.appender.fs.MaxFileSize=10MB
log4j.appender.fs.MaxBackupIndex=9
log4j.appender.fs.layout=org.apache.log4j.PatternLayout
log4j.appender.fs.layout.ConversionPattern=[%d] %t %c %-5p - %m%n
```

Bei Verwendung des Tomcat 6 die Datei `log4j-1.2.*.jar` von <http://logging.apache.org/log4j/1.2/download.html> herunterladen und nach `tomcat/lib/` kopieren. Bei der Verwendung des Tomcat 7 ist dies nicht notwendig.





Bei der Verwendung von Tomcat statt Jetty wird an allen Stellen in der Server- und Projektkonfiguration (in den Server-Eigenschaften vgl. Kapitel 7.3.12 und 7.3.13 ab Seite 261, in den Projekt-Eigenschaften Kapitel 7.4.18 Seite 327) sowie in der Datei `fs-server.conf` überall "InternalJetty" als Bezeichner des Webserver angezeigt, da aus der Sicht des FirstSpirit-Servers der Tomcat-Server genau die gleichen Dateien an derselben Stelle verwendet wie der Jetty-Webserver.

Für die von Tomcat verwendete Java-VM gelten die in Kapitel 4.6.3 (Seite 134) beschriebenen Anforderungen an die Konfiguration eines Application-Servers. Bei Tomcat erfolgt die Konfiguration über die Datei `tomcat/bin/setenv.sh` mit dem folgenden Inhalt, der den JVM-Parametern aus `firstspirit5/conf/fs-wrapper.conf` entspricht. Die Angaben für die Java-Heapsize (Xmx, Xms, Xmn) müssen entsprechend des verfügbaren RAMs angepasst werden

```
# use same JVM path as given in firstspirit5/conf/fs-wrapper.conf
# with parameter wrapper.java.command
JAVA_HOME=/opt/java/jdk1.7.0

# Tomcat Heapsize settings
# set Xmx and Xms to max of 75% of available RAM, max 10000M
# set Xmn to 40% of Xmx
# Change jmxremote port to any free available port
# and consider activated jmx password security.
CATALINA_OPTS="\
-Xmx4000M -Xms4000M -Xmn1600m \
-XX:PermSize=500M -XX:MaxPermSize=500M \
-XX:InitialCodeCacheSize=128M \
-XX:ReservedCodeCacheSize=128M \
-XX:SurvivorRatio=1 \
-XX:SoftRefLRUPolicyMSPerMB=1 \
-XX:+NeverTenure \
-XX:-UseLargePages \
-XX:+UseParNewGC \
-XX:+UseConcMarkSweepGC \
-XX:+CMSParallelRemarkEnabled \
-XX:+CMSClassUnloadingEnabled \
\
-Djava.awt.headless=true \
-Dfile.encoding=UTF-8 \
-Djava.net.preferIPv4Stack=true \
\
-Dcom.sun.management.jmxremote \
-Dcom.sun.management.jmxremote.ssl=false \
-Dcom.sun.management.jmxremote.authenticate=false \
-Dcom.sun.management.jmxremote.port=8006 \
\
-verbose:gc \
-XX:+PrintGCTimeStamps \
-XX:+PrintGCDateStamps \
-XX:+PrintGCDetails \

```



```
-Xloggc:/opt/tomcat/logs/tomcat-gc.log \  
"  
CATALINA_PID=/opt/firstspirit5/tomcat/work/catalina.pid
```

In der Datei `firstspirit5/tomcat/bin/catalina.sh` die Zeile

```
FORCE=0
```

in

```
FORCE=1
```

ändern, um Tomcat definiert beenden zu können, ohne dass ein Prozess "hängenbleibt". Falls `catalina.sh` nicht geändert werden soll, ist bei jedem Herunterfahren der Parameter `-force` hinzuzufügen, also `catalina.sh stop -force`.

Tomcat wird nun über folgende Zeile gestartet:

```
/opt/firstspirit5/tomcat/bin/catalina.sh start
```

Zum Beenden:

```
/opt/firstspirit5/tomcat/bin/catalina.sh stop
```

Um Tomcat automatisch beim Systemstart gleichzeitig mit FirstSpirit zu starten, in `/etc/init.d/fs5` folgende Zeile im Bereich "start" hinzufügen

```
su - $FSUSER -c "$FSDIR/../tomcat/bin/catalina.sh start"
```

und folgende Zeile im Bereich "stop":

```
su - $FSUSER -c "$FSDIR/../tomcat/bin/catalina.sh stop -force"
```



4.5.4 Servlet-Engine Tomcat auf dediziertem Host

Der in FirstSpirit 5 integrierte Webserver Jetty 8 bietet als Standards Servlet 3.0 und JSP 2.1 und ist nicht vollständig kompatibel zu AJP. Falls in FirstSpirit-Projekten Webanwendungen den vollständigen J2EE-Standard benötigen oder AJP zur Anbindung eines HTTP-Servers oder Loadbalancers, bietet es sich an, Apache Tomcat 7¹⁸ zu verwenden, dessen Konfiguration für FirstSpirit im folgenden Abschnitt beschrieben wird.

Tomcat kann auf demselben Host wie der FirstSpirit-Server betrieben werden (Kapitel 4.5.3) oder auf einem eigenen, dedizierten Host (dieses Kapitel) oder zur Lastverteilung auf mehreren Host (Kapitel 4.5.5).

Tomcat wird in dieser Konfiguration auf einem anderen Host als der FirstSpirit-Server betrieben, um eine bessere Lastverteilung bezüglich der Ressourcen CPU und RAM zu ermöglichen.

Tomcat kann entweder eigenständig mit dem in Tomcat integrierten HTTP-Server betrieben werden oder zusätzlich um den Apache HTTP-Server ergänzt werden, wie in Kapitel 4.5.2 beschrieben.

Bevor Tomcat konfiguriert wird, ist die FirstSpirit-Konfiguration zu ändern. In der Datei `firstspirit5/conf/fs-server.conf` müssen folgende Parameter hinzugefügt werden, um die Standard-Webanwendung zu definieren, die beim Aufruf der FirstSpirit-Startseite über Tomcat, in diesem Beispiel `http://fs5.yourdomain.net:8080`, verwendet wird und über welche URL der Tomcat aus Sicht der Clients später erreichbar sein soll, um beim Versand von Nachrichten außerhalb von aktiven Client-Verbindungen eine gültige URL zu generieren:

```
WEBAPP_ROOT_PATH=${WEB_DIR}/ROOT

# URL-Parameter for workflow mail
URL=http://fs5server.domain.net:8080
fs.url.hostname=fs5server.domain.net
fs.url.httpport=8080
```

¹⁸ <http://tomcat.apache.org>. Ein Fehler (https://issues.apache.org/bugzilla/show_bug.cgi?id=50700) in den Tomcat-Versionen 6.0.30 bis 6.0.32 verhindert das Auslesen der Kontext-Parameter, die in der Datei `context.xml` für FirstSpirit notwendig sind. Es sollte also entweder Tomcat 6.0.29 oder mindestens 6.0.33 oder 7.0 verwendet werden.



Falls https verwendet wird oder statt des http-Verbindungsmodus des FirstSpirit-Clients der Socket-Modus angewendet werden soll, siehe erweiterte Parameter zu "fs.url" in Kapitel 4.3.1.1.

Anschließend den FirstSpirit-Server herunterfahren, das Verzeichnis firstspirit5/web/fs5root entfernen und den FirstSpirit-Server wieder starten.

Die Installation von Tomcat, Version 6 oder 7, verwendet wird "Binary Distribution Core" von <http://tomcat.apache.org>, erfolgt in diesem Beispiel nach `/opt/tomcat`.

In `tomcat/conf/server.xml` die Kodierung für URI-Parameter auf UTF-8 einstellen, dazu bei den bereits vorhandenen HTTP- und AJP-Konnektoren den Parameter `URIEncoding` hinzufügen:

```
<Connector port="8080" protocol="HTTP/1.1" URIEncoding="UTF-8" />
<Connector port="8009" protocol="AJP/1.3" URIEncoding="UTF-8" />
```

In der Datei `tomcat/conf/web.xml` das Überprüfen und Kompilieren der JSP-Dateien bei jeder Änderung ohne Verzögerung aktivieren. Dazu die folgenden bereits vorhandenen Zeilen der Datei

```
<servlet>
  <servlet-name>jsp</servlet-name>
  <servlet-class>org.apache.jasper.servlet.JspServlet</servlet-class>
  <init-param>
    <param-name>fork</param-name>
    <param-value>>false</param-value>
  </init-param>
  <init-param>
    <param-name>xpoweredBy</param-name>
    <param-value>>false</param-value>
  </init-param>
  <load-on-startup>3</load-on-startup>
</servlet>
```



durch folgende Zeilen ersetzen:

```
<servlet>
  <servlet-name>jsp</servlet-name>
  <servlet-class>org.apache.jasper.servlet.JspServlet</servlet-class>
  <init-param>
    <param-name>fork</param-name>
    <param-value>>false</param-value>
  </init-param>
  <init-param>
    <param-name>xpoweredBy</param-name>
    <param-value>>false</param-value>
  </init-param>
  <init-param>
    <param-name>development</param-name>
    <param-value>true</param-value>
  </init-param>
  <init-param>
    <param-name>modificationTestInterval</param-name>
    <param-value>0</param-value>
  </init-param>
  <load-on-startup>3</load-on-startup>
</servlet>
```

Falls die Möglichkeit des Auflistens der Verzeichnisinhalte zur Entwicklung von Webanwendungen im Staging-Bereich des FirstSpirit-Servers benötigt wird, in derselben Datei die Zeilen

```
<servlet>
  <servlet-name>default</servlet-name>
  <servlet-class>org.apache.catalina.servlets.DefaultServlet</servlet-class>
  <init-param>
    <param-name>debug</param-name>
    <param-value>0</param-value>
  </init-param>
  <init-param>
    <param-name>listings</param-name>
    <param-value>>false</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
```

durch folgende Zeilen ersetzen:

```
<servlet>
  <servlet-name>default</servlet-name>
  <servlet-class>org.apache.catalina.servlets.DefaultServlet</servlet-class>
  <init-param>
    <param-name>debug</param-name>
    <param-value>0</param-value>
  </init-param>
  <init-param>
    <param-name>listings</param-name>
    <param-value>true</param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
```



Tomcat 6

Um den Tomcat-Manager zur Status-Überwachung zu aktivieren, in der Datei `tomcat/conf/tomcat-users.xml` ein Benutzerkonto für die Rolle "manager" des Tomcat-Managers einfügen:

```
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>
  <role rolename="manager"/>
  <user username="Admin" password="tomcat-password" roles="manager"/>
</tomcat-users>
```

Tomcat 7

Um den Tomcat-Manager zur Status-Überwachung über die URL `http://fs5.yourdomain.net:8080/manager/html` als Benutzer "Admin" verwenden zu können und das Deployment der WebApps automatisiert über FirstSpirit durchzuführen, in der Datei `tomcat/conf/tomcat-users.xml` ein Benutzerkonto für jede der Rollen "manager-gui" und "manager-script" des Tomcat-Managers einfügen:

```
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>
  <role rolename="manager-gui"/>
  <role rolename="manager-script"/>
  <user username="Admin" password="tomcat-password" roles="manager-gui"/>
  <user username="fsdeploy" password="deploy-password" roles="manager-script"/>
</tomcat-users>
```

Die vorhandene Datei `tomcat/conf/context.xml`¹⁹ durch folgenden Inhalt ersetzen, um das sofortige Prüfen auf Dateiänderungen (HTML) zu aktivieren. Über den bei "firstspirit.host" angegebenen Hostnamen oder die IP-Adresse muss von Tomcat aus der FirstSpirit-Server im lokalen Netz erreichbar sein. Die Angabe bei "firstspirit.port" muss mit dem Port des FirstSpirit-Servers, wie in `firstspirit5/conf/fs-server.conf` über Parameter `SOCKET_PORT` definiert, übereinstimmen. Zusätzlich wird das Speichern persistenter Session-Informationen deaktiviert, weil die FirstSpirit-Webanwendungen dies nicht unterstützen:

¹⁹ Ein Fehler (https://issues.apache.org/bugzilla/show_bug.cgi?id=50700) in den Tomcat-Versionen 6.0.30 bis 6.0.32 verhindert das Auslesen der Kontext-Parameter, die in der Datei `context.xml` für FirstSpirit notwendig sind. Es sollte also entweder Tomcat 6.0.29 oder mindestens 6.0.33 oder 7.0 verwendet werden.



```
<?xml version='1.0' encoding='utf-8'?>
<Context allowLinking="true" cachingAllowed="false" useHttpOnly="true">

  <WatchedResource>WEB-INF/web.xml</WatchedResource>

  <!-- disable session persistence across Tomcat restarts -->
  <Manager pathname="" />

  <Parameter name="firstspirit.host" value="fs5server" override="false" />
  <Parameter name="firstspirit.port" value="1088" override="false" />

</Context>
```

Die Datei `tomcat/lib/log4j.properties` mit folgendem Inhalt erstellen, um das Logging der FirstSpirit-Webanwendungen in eine eigene Datei umzuleiten:

```
log4j.rootCategory=INFO, fs

# change INFO in the following line to DEBUG
# for detailed FirstSpirit logging:
log4j.logger.de.espirit=INFO

log4j.logger.org.mortbay=WARN
log4j.logger.org.apache.catalina=INFO
log4j.logger.org.apache.jasper=WARN
log4j.logger.org.apache.log4j.jmx=ERROR
log4j.logger.org.apache.commons.httpclient=INFO

log4j.appender.fs=org.apache.log4j.RollingFileAppender
log4j.appender.fs.File=${catalina.home}/logs/firstspirit.log
log4j.appender.fs.MaxFileSize=10MB
log4j.appender.fs.MaxBackupIndex=9
log4j.appender.fs.layout=org.apache.log4j.PatternLayout
log4j.appender.fs.layout.ConversionPattern=[%d] %t %c %-5p - %m%n
```

Die Datei `log4j-1.2.*.jar` von <http://logging.apache.org/log4j/1.2/download.html> herunterladen und nach `tomcat/lib/` kopieren. *Dieser Schritt muss nur bei der Verwendung eines Tomcat 6 und nicht bei der Verwendung eines Tomcat 7 durchgeführt werden.*

Für die von Tomcat verwendete Java-VM gelten die in Kapitel 4.6.3 (Seite 134) beschriebenen Anforderungen an die Konfiguration eines Application-Servers. Bei Tomcat erfolgt die Konfiguration über die Datei `tomcat/bin/setenv.sh` mit dem folgenden Inhalt, der den JVM-Parametern aus `firstspirit5/conf/fs-wrapper.conf` entspricht. Die Angaben für die Java-Heapsize (`Xmx`, `Xms`, `Xmn`) müssen entsprechend des verfügbaren RAMs angepasst werden:



```
# use same JVM path as given in firstspirit5/conf/fs-wrapper.conf
# with parameter wrapper.java.command
JAVA_HOME=/opt/java/jdk1.7.0

# Tomcat Heapsize settings
# set Xmx and Xms to max of 75% of available RAM, max 10000M
# set Xmn to 40% of Xmx
# Change jmxremote port to any free available port
# and consider activated jmx password security.
CATALINA_OPTS="\
-Xmx4000M -Xms4000M -Xmn1600m \
-XX:PermSize=500M -XX:MaxPermSize=500M \
-XX:InitialCodeCacheSize=128M \
-XX:ReservedCodeCacheSize=128M \
-XX:SurvivorRatio=1 \
-XX:SoftRefLRUPolicyMSPerMB=1 \
-XX:+NeverTenure \
-XX:-UseLargePages \
-XX:+UseParNewGC \
-XX:+UseConcMarkSweepGC \
-XX:+CMSParallelRemarkEnabled \
-XX:+CMSClassUnloadingEnabled \
\
-Djava.awt.headless=true \
-Dfile.encoding=UTF-8 \
-Djava.net.preferIPv4Stack=true\
\
-Dcom.sun.management.jmxremote \
-Dcom.sun.management.jmxremote.ssl=false \
-Dcom.sun.management.jmxremote.authenticate=false \
-Dcom.sun.management.jmxremote.port=8006 \
\
-verbose:gc \
-XX:+PrintGCTimeStamps \
-XX:+PrintGCDateStamps \
-XX:+PrintGCDetails \
-Xloggc:/opt/tomcat/logs/tomcat-gc.log \
"

CATALINA_PID=/opt/firstspirit5/tomcat/work/catalina.pid
```

In der Datei `firstspirit5/tomcat/bin/catalina.sh` die Zeile

```
FORCE=0
```

in

```
FORCE=1
```

ändern, um Tomcat definiert beenden zu können, ohne dass ein Prozess "hängenbleibt". Falls `catalina.sh` nicht geändert werden soll, ist bei jedem Herunterfahren der Parameter `"-force"` hinzuzufügen, also `"catalina.sh stop -force"`.



Tomcat wird nun über folgende Zeile gestartet:

```
/opt/tomcat/bin/catalina.sh start
```

Zum Beenden:

```
/opt/tomcat/bin/catalina.sh stop
```

Um Tomcat automatisch beim Systemstart zu starten, die Datei `/etc/init.d/tomcat` mit folgendem Inhalt erstellen, Dateizugriffsrechte über `chmod a+rx /etc/init.d/tomcat` definieren und je nach Betriebssystem über `insserv`, `update-rc.d` oder `chkconfig` der Systemstartumgebung hinzufügen:

```
#!/usr/bin/bash

### BEGIN INIT INFO
# Provides: tomcat
# Required-Start: $local_fs $network
# Should-Start: $netdaemons $named $syslog $remote_fs sendmail
# Required-Stop: $local_fs $network
# Should-Stop: $netdaemons $named $syslog $remote_fs sendmail
# Default-Start: 2 3 5
# Default-Stop: 0 1 6
# Short-Description: FirstSpirit-Tomcat
# Description: FirstSpirit-Tomcat
# chkconfig: 235 95 05
### END INIT INFO

TCUSER=tomcat
TCDIR=/opt/tomcat

case "$1" in

start)
    su - $TCUSER -c "$TCDIR/bin/catalina.sh start" \
    && touch /var/lock/subsys/tomcat_$TCUSER
    ;;

stop)
    su - $TCUSER -c "$TCDIR/bin/catalina.sh stop -force" \
    && rm /var/lock/subsys/tomcat_$TCUSER
    ;;

restart)
    $0 stop
    $0 start
    ;;

*)
    echo "Usage: $0 { start | stop | restart }"
    exit 1
    ;;

esac
```



In der FirstSpirit Server- und Projektkonfiguration unter Server-Eigenschaften muss nun ein Webserver des Typs "Tomcat" angelegt werden, beispielsweise mit dem Namen "PreviewTomcat" (siehe dazu Kapitel 7.3.12.5 Seite 267).

Parameter:

- Webserver-URL: leer lassen
- Webverzeichnis: leer lassen
- Tomcat-Benutzer: siehe username bei der Rolle "manager-script" in der Datei `tomcat/conf/tomcat-users.xml`
- Tomcat-Passwort: das dem zuvor eingetragenen Benutzer gehörende Passwort, siehe in der Datei `tomcat/conf/tomcat-users.xml`
- Tomcat-Manager-URLs:
Tomcat 7: `http://tomcathost:8080/manager/text`
Tomcat 6: `http://tomcathost:8080/manager/html`
(lokaler Hostname des Tomcat-Servers, wie er vom FirstSpirit-Server aus erreichbar ist, sowie Tomcat-HTTP-Port. Werden mehrere Tomcat-Instanzen eingesetzt, werden die URLs kommasepariert eingegeben.)

Nun in den Server-Eigenschaften die FirstSpirit-Web-Applikationen von InternalJetty auf PreviewTomcat umschalten und über die Schaltfläche "Installieren" die jeweiligen Webanwendungen auf dem Tomcat-Server installieren. Diese Installation erfolgt über das zuvor eingetragene Skript automatisch. Sie muss zurzeit (FirstSpirit 5.0) bei jedem FirstSpirit-Update zur Aktualisierung der Web-Applikationen wiederholt werden.

Falls projektlokale Web-Applikationen (Kapitel 7.4.18 Seite 327) verwendet werden, müssen diese ebenso auf PreviewTomcat umgeschaltet werden.



Wurde die `fs-server.jar`-Datei aktualisiert, müssen die auf dem Tomcat-Server installierten FirstSpirit-Web-Applikationen erneut auf diesem installiert und aktiviert werden. Diese Handlung muss für alle auf dem Tomcat-Webserver installierten Web-Applikationen in allen Projekten durchgeführt werden. Ein automatisches Update erfolgt nicht (siehe auch: Kapitel 7.4.18.4 Seite 331).



4.5.5 Externe Servlet-Engine und Lastverteilung auf mehrere Servlet-Engines

In diesem Kapitel wird die Konfiguration zur Lastverteilung auf zwei Servlet-Engines beschrieben, die auf den Hosts tomcat-host1 und tomcat-host2 installiert werden. Als Lastverteiler wird der Apache HTTP-Server mit `mod_proxy_balancer` in Version 2.2 verwendet.

Es können auch andere Load-Balancer und Servlet-Engines bzw. Application-Server verwendet werden, wobei dann zu beachten ist, Session-Affinity auf Basis des Cookie-Eintrags für die Session-ID zu aktivieren. Damit wird sichergestellt, dass eine Benutzer-Session vom ersten Aufruf der FirstSpirit-Startseite bis zur Abmeldung nach Arbeitsende immer von derselben Servlet-Engine bearbeitet wird. Tomcat verwendet als Cookie-Eintrag für die Session-ID den Eintrag "JSESSIONID", der bei anderen Server-Kombinationen entsprechend angepasst werden muss.

Diese Konfiguration dient nur zur Lastverteilung und nicht zur Erhöhung der Ausfallsicherheit! Ein Session-Failover zwischen den Servlet-Engines findet nicht statt und wird von FirstSpirit auch nicht unterstützt, da die FirstSpirit-Webanwendungen ihren Zustand nicht persistent serialisieren können.

Die statischen Dateien der FirstSpirit-Webanwendungen, die von FirstSpirit automatisch bei jeder Aktualisierung nach `/opt/firstspirit5/web` geschrieben werden, und die über FirstSpirit generierten Dateien zur Anzeige im Webserver werden über NFS (readonly) für die einzelnen Servlet-Engines bereitgestellt.

Zunächst die Konfiguration des HTTP-Servers und der Servlet-Engines, wie in den Kapiteln 4.5 bis 4.5.3 beschrieben, durchführen. Anschließend in der Apache-Konfiguration weitere Servlet-Engines als "BalancerMember" eintragen und die Session-Affinity auf Cookie-Wert JSESSIONID definieren.

Ergänzung zur Apache-Konfiguration des virtuellen Webserver im Bereich "<Proxy balancer">:

```
<Proxy balancer://fsajp>
  BalancerMember ajp://tomcat-host1:8009 loadfactor=100 route=node1 retry=10
  connectiontimeout=10 ping=5 ttl=1800 timeout=1200
  BalancerMember ajp://tomcat-host2:8009 loadfactor=100 route=node2 retry=10
  connectiontimeout=10 ping=5 ttl=1800 timeout=1200
  BalancerMember ajp://tomcat-host3:8009 loadfactor=100 route=node3 retry=10
  connectiontimeout=10 ping=5 ttl=1800 timeout=1200
  ProxySet stickysession=JSESSIONID
  ProxySet lbmethod=byrequests
  ProxySet timeout=30
  #ProxySet nofailover=On
</Proxy>
```



In der Tomcat-Konfigurationsdatei `tomcat/conf/server.xml` die Zeile

```
<Engine name="Catalina" defaultHost="localhost">
```

ändern in

```
<Engine name="Catalina" defaultHost="localhost" jvmRoute="nodeX">
```

und dabei "nodeX" durch den jeweiligen bei `BalancerMember` eingetragenen Routenamen der Servlet-Engine ersetzen.

Außerdem alle Vorkommen von `${catalina.home}/../`, wo auf die FirstSpirit-Verzeichnisse `web`, `server/lib` und `data/fslib`, in `conf/server.xml` und `conf/catalina.properties` verwiesen wird, durch `/import/firstspirit5/` ersetzen.

Das Verzeichnis `/opt/FirstSpirit5` des FirstSpirit-Servers über NFS read-only allen Servlet-Engines als lokalen Mountpoint `/import/firstspirit5` bereitstellen.

Die Preview-Cache-Verzeichnisse müssen in dieser Konfiguration außerhalb des nur lesbaren Dateisystems `/import/firstspirit5` liegen, da sie von den Webanwendungen `fs5preview` und `fs5webedit` beschrieben werden. Dazu werden beide Preview-Cache-Verzeichnisse über Symlinks auf jeweils lokale Dateisystem umgeleitet. Auf dem FirstSpirit-Server dazu Folgendes aufrufen:

```
mkdir -p /opt/firstspirit5/previewcache/fs5preview
mkdir -p /opt/firstspirit5/previewcache/fs5webedit
rm -rf /opt/firstspirit5/web/fs5preview/preview_cache
rm -rf /opt/firstspirit5/web/fs5webedit/preview_cache
ln -s /opt/firstspirit5/previewcache/fs5preview \
    /opt/firstspirit5/web/fs5preview/preview_cache
ln -s /opt/firstspirit5/previewcache/fs5webedit \
    /opt/firstspirit5/web/fs5webedit/preview_cache
```

Auf jeder Tomcat-Instanz werden die lokalen Cache-Verzeichnisse erstellt:

```
mkdir -p /opt/firstspirit5/previewcache/fs5preview
mkdir -p /opt/firstspirit5/previewcache/fs5webedit
```

Andere Verzeichnisse unterhalb von `/opt/firstspirit5` werden auf Seite der Tomcat-Instanz nicht benötigt, da sie bereits im Lesezugriff über NFS auf `/import/firstspirit5` erreichbar sind.

Zusätzlich muss in der Konfigurationsdatei `/opt/firstspirit5/conf/fs-server.conf` der Hostname oder die IP-Adresse des FirstSpirit-Servers eingetragen werden, damit diese als Zieladresse für die TCP-Verbindung (FirstSpirit-



SOCKET) von den Servlet-Engines zum FirstSpirit-Server benutzt werden kann:

```
HOST=my-local-FirstSpirit-hostname
```

Anschließend FirstSpirit-Server neu starten, Servlet-Engines starten und Apache neu starten und bei Bedarf über <http://fs5.yourdomain.net/balancer-manager/> die Erreichbarkeit der Servlet-Engines prüfen.

Die FirstSpirit-Startseite kann nun beispielsweise über <http://fs5.yourdomain.net> aufgerufen werden.

4.6 Einbinden in externen Application-Server

Sofern der Application-Server mit Zugriff auf das Verzeichnis `web` der FirstSpirit-Installation konfiguriert werden kann, bietet sich die unter Kapitel 4.5.2 beschriebene Konfiguration an.

Falls kein Zugriff auf das Dateisystem des FirstSpirit-Servers vom Application-Server aus möglich ist, kann die im Folgenden beschriebene Konfiguration verwendet werden:

Die FirstSpirit-Webanwendungen liegen in Form einer WAR-Datei vor. Die WAR-Datei wird vom FirstSpirit-Server bei jedem Start automatisch mit passender Konfiguration erstellt. Die Verbindungsparameter Socket-Host und Socket-Port zwischen FirstSpirit-Webanwendungen und FirstSpirit-Server werden dazu aus der Datei `fs-server.conf` ausgelesen und in den Deployment-Descriptor der WAR-Datei zur Konfiguration der FirstSpirit-Webanwendungen eingetragen.

Zur Konfiguration der Nutzung eines externen Application-Servers sind folgende Schritte auszuführen:

1. In der Konfigurationsdatei `/opt/firstspirit5/conf/fs-server.conf` muss der Hostname oder die IP-Adresse des FirstSpirit-Servers eingetragen werden, damit diese als Zieladresse für die TCP-Verbindung (FirstSpirit-SOCKET) von den Servlet-Engines zum FirstSpirit-Server benutzt werden kann:

```
HOST=my-local-FirstSpirit-hostname
```

2. Zum Einbinden eines externen Application-Servers die Anwendung zur Server- und Projektkonfiguration öffnen. Die Startseite zum Aufruf der Anwendung wird, unabhängig vom externen Application-Server, durch den in FirstSpirit integrierten Webserver bereitgestellt. Damit ist sichergestellt, dass bei Konfigurationsproblemen des externen Application-Servers die



Verwaltung des FirstSpirit-Servers weiterhin möglich ist. Im Bereich "Server-Eigenschaften" (siehe Kapitel 7.3.12 Seite 261) können nun beliebige Webserver-Steuerungen hinzugefügt und konfiguriert werden. Zunächst muss der gewünschte externe Webserver hinzugefügt werden (siehe 7.3.12.3 Seite 265). Das Eingabefeld "Webverzeichnis" bleibt bei dieser Installation über WAR-Dateien leer, da FirstSpirit keinen Zugriff auf das Dateisystem des Application-Servers hat.

3. Jede FirstSpirit-Webanwendung muss auf den neu konfigurierten externen Webserver umgeschaltet werden, wie in Kapitel 7.3.13.5 auf Seite 273 beschrieben.
4. Für das Einbinden eines externen Application-Servers, ohne Zugriff auf das Dateisystem des FirstSpirit-Servers, sind WAR-Dateien zum Installieren der FirstSpirit-Webanwendungen verfügbar. Die WAR-Dateien können über die Server- und Projektkonfiguration wie in Kapitel 7.3.13.5 (Seite 273 f.) beschrieben, automatisch erzeugt und heruntergeladen werden. Die Installation der WAR-Dateien erfolgt anschließend manuell über das jeweilige Web-Interface des Application-Servers.
5. Die FirstSpirit Web-Applikationen (z. B. Server-Monitoring) können nun auf dem Webserver installiert werden (siehe Kapitel 7.3.13.5 Seite 273).

4.6.1 Einbinden in Websphere Application Server

FirstSpirit macht keine (über das *Technische Datenblatt* hinausgehende) Vorgaben hinsichtlich des eingesetzten Application Servers. In der Praxis hat sich aber gezeigt, dass speziell beim Einsatz von IBM Websphere folgende Besonderheiten zu beachten sind.

Zusätzlich zu der zuvor erwähnten allgemeinen Konfiguration zum Einbinden in einen externen Application Server sind bei Verwendung von IBM Websphere als Application Server folgende Konfigurationen notwendig:

Die Datei `firstspirit5/data/fslib/fs-webrt.jar` muss in den globalen Klassenpfad des Websphere-Servers eingetragen werden, da die von e-Spirit digital signierten JARs in den Webanwendungen unter WEB-INF/lib nicht geladen werden können. Die Datei `fs-webrt.jar` darf dazu nicht als "gemeinsame Bibliothek" eingebunden werden, sondern muss Websphere als Parameter der Java-VM übergeben werden.

Die Parameter der Java-VM findet man im Webinterface zur Websphere-Administration unter "Server > Anwendungsserver > Servername > Java- und Prozessverwaltung > Prozessdefinition > Java Virtual Machine". Dort wird unter



"Klassenpfad" der vollständige Pfad zur Datei `fs-webrt.jar` angegeben, zum Beispiel `/opt/firstspirit5/data/fslib/fs-webrt.jar`. Falls der FirstSpirit-Server auf einem anderen Server betrieben wird als Websphere, empfiehlt es sich, das Verzeichnis `opt/firstspirit5/data/fslib` über NFS an Websphere weiterzugeben, um die Datei `fs-webrt.jar` nicht manuell bei jeder Aktualisierung des FirstSpirit-Servers zum Websphere-Server kopieren zu müssen.

Ebenfalls in den Klassenpfad der Java-VM des Websphere-Servers müssen alle JAR-Dateien eingetragen werden, die in von e-Spirit ausgelieferten Modulen verwendet werden. Folgendes Vorgehen ist nach Aktivierung eines Moduls für die Bereiche Vorschau (`fs5preview`), WebClient (`fs5webedit`) oder Staging (`fs5staging`) notwendig: Zunächst die über die Server- und Projekteigenschaften heruntergeladene WAR-Datei/en öffnen mittels ZIP-Programm oder `jar xvf fs_<Modulname>.war` und alle enthaltenen JAR-Dateien mit Namensmuster `fs-*.jar` aus dem Verzeichnis `WEB-INF/lib` in das Klassenpfad-Verzeichnis des Websphere-Servers kopieren, wo bereits `fs-webrt.jar` abgelegt wurde. Im Websphere-Admin-Interface die kopierten JAR-Dateien analog zu `fs-webrt.jar` dem Klassenpfad hinzufügen und Websphere neu starten. Dieses Vorgehen ist nach jeder Aktualisierung von FirstSpirit notwendig!

Außerdem sind bei "Generische JVM-Argumente" folgende Einträge notwendig:
`-Djava.awt.headless=true -Dclient.encoding.override=UTF-8`

Folgender Eintrag ist in der Datei `fs-server.conf` beim Betrieb unter Websphere notwendig:

```
WEBAPP_ROOT_URL=/fs5root
```

(Für weitere Informationen zu diesem Parameter siehe auch Kapitel 4.3.1.7 Seite 43)

Die FirstSpirit-Startseite ist dann auf dem Websphere-Server als `http://fs5host.domain.net/fs5root` erreichbar. Um die Erreichbarkeit über die einfachere URL `http://fs5host.domain.net` zu ermöglichen, muss im HTTP-Server des Websphere-Servers folgende Umleitungsregel definiert werden:

```
RewriteEngine On  
RewriteRule ^/$ /fs5root/ [redirect,last]
```

In der Datei `fs-server.conf` muss ebenfalls der Parameter `preview.cacheFileWithTimestamp=*` gesetzt werden (für weitere Informationen zu diesem Parameter siehe Kapitel 4.3.1.8 Seite 47).





Die Java-VM des Websphere Application Servers muss so konfiguriert werden, dass sie Java-Klassen zu nicht mehr vorhandenen JSP-Dateien automatisch regelmäßig entfernt. Bei der Java-VM von Oracle geschieht das über Verwendung des Parameters

`-XX:+CMSClassUnloadingEnabled`
in Kombination mit
`-XX:+UseConcMarkSweepGC`

Um diese Änderungen im Websphere anschließend zu aktivieren, muss der Websphere-Server neu gestartet werden!

4.6.2 Logging für FirstSpirit-Webanwendungen

Das Logging für FirstSpirit-Webanwendungen bei Einsatz externer Application-Servern erfolgt in eine eigene Datei auf dem (Remote-)FirstSpirit-Server im Verzeichnis `firstspirit5/log`, deren Dateiname nach dem Schema

```
fs-webapp-[Hostname]-[Port].log
```

gebildet wird, z. B. `fs-webapp-localhost-1088.log`. Hostname und Portnummer beziehen sich dabei auf den Application-Server, auf dem die Webanwendung installiert ist. Standardmäßig werden Meldungen mit dem Level `WARN` und höher in die Logdatei aufgenommen. Zu Debug-Zwecken kann das Level über den Parameter `WEBAPP_LOG_LEVEL` in der Konfigurationsdatei `fs-server.conf` aber temporär auch auf ein niedrigeres Level eingestellt werden (siehe dazu Kapitel 4.3.1.7 Seite 43).

Die Übertragung erfolgt nicht in Echtzeit, Duplikate werden herausgefiltert. Wie im Falle anderer FirstSpirit-Logdateien werden die `fs-webapp`-Logdateien ebenfalls in `*.gz`-Dateien komprimiert und archiviert. Diese erhalten einen Zeitstempel im Dateinamen. Archivierte Dateien können mit dem Server-Auftrag "Clean up logs" gelöscht bzw. nach `firstspirit5/backup` verschoben werden.

4.6.3 Anforderungen an externen Application-Server

Die Anforderungen der FirstSpirit-Web-Applications entsprechen zunächst denen jeder Web-Application, dass ein pausenloser Betrieb sichergestellt ist. Das bedeutet vor allem, eine vorausplanende Konfiguration der Java-VM bezüglich Heap-Größe und Garbage Collector vorzunehmen. Für die Konfiguration der Java-VM eines externen Application-Servers gelten diesselben Grundlagen wie die in Kapitel 4.3.2.1



"Konfiguration der Java-VM" bezüglich des FirstSpirit-Servers beschrieben. Folgende Parameter dienen als Anhaltspunkt zur Konfiguration, falls die Java-VM von Oracle verwendet wird:

```
-Xms4000m \  
-Xmx4000m \  
-XX:MaxPermSize=512m \  
-XX:NewRatio=20 \  
-XX:+UseConcMarkSweepGC \  
-XX:+CMSClassUnloadingEnabled \  
-XX:-UseLargePages \  
-Djava.awt.headless=true \  
-Dcom.sun.management.jmxremote \  
-Dcom.sun.management.jmxremote.ssl=false \  
-Dcom.sun.management.jmxremote.authenticate=false \  
-Dcom.sun.management.jmxremote.port=5555
```

Der Parameter `-Xms` definiert die initiale Heap-Größe (entspricht `wrapper.java.initmemory` in `fs-wrapper.conf`), der Parameter `-Xmx` die maximale Heap-Größe (entspricht `wrapper.java.maxmemory` in `fs-wrapper.conf`). Ein Wert von maximal 75% des verfügbaren Hauptspeichers ist empfehlenswert. Der Heap sollte nicht unnötig groß gewählt werden, da FirstSpirit versucht, den gesamten freien Speicher als Cache zu belegen. Die Parameter `com.sun.management.jmxremote` dienen zum Monitoring der Heap-Auslastung, auch während des Produktionsbetriebs, so dass Trends bezüglich der Speicherbelegung erkannt werden können und bei Bedarf eine Vergrößerung der Heapsize erfolgen kann. Aktuelle Monitoring-Systeme bieten eine JMX-Schnittstelle, um darüber Web-Application-Server überwachen zu können. Zur Abfrage der aktuellen Werte ist die Web-Application "Lambda-Probe" empfehlenswert (<http://www.lambdaprobe.org>). Darüber hinaus kann das Programm `jconsole` (<http://docs.oracle.com/javase/1.5.0/docs/guide/management/jconsole.html>) zur interaktiven Abfrage verwendet werden. Es ist Bestandteil jedes JDK (nicht JRE) von Oracle.

Gerade bei Application-Servern ist die Konfiguration der Parameters `MaxPermSize` wichtig, da in diesem Speicherbereich für jede in Verwendung befindliche JSP-Datei eine Java-Klasse angelegt wird. Der Bereich muss also genügend groß gewählt werden. Über JMX kann die aktuelle Belegung abgefragt werden.



4.7 Konfiguration des HTTPS-Servers

Der in FirstSpirit integrierte Webserver kann für HTTPS konfiguriert werden, um die übertragenden Daten der Webanwendungen (WebClient, Startseite und Server-Monitoring) mittels TLS/SSL zu verschlüsseln.

Dazu muß zunächst ein Server-Zertifikat über das beim JDK mitgelieferte Programm `keytool` installiert werden und anschließend der HTTPS-Listener des Webserver aktiviert werden.

4.7.1 Sicherheitszertifikat für einen Testserver installieren

Für Testinstallationen kann das selbstsignierte Zertifikat aus dem mitgelieferten Zertifikatsspeicher (`conf/fs-keystore.jks`) verwendet werden. Dazu kann direkt mit der Änderung der Webserver-Konfiguration fortgefahren werden (Kapitel 4.7.2.2 Seite 138). Falls ein eigenes Testzertifikat mit anderem Hostnamen erstellt werden soll, kann dies über folgenden Aufruf geschehen:

```
keytool -genkeypair -alias fs5.yourdomain.net -keyalg RSA -validity 1000 -  
keystore conf/fs-keystore.jks -storepass changeit
```

Bei der Abfrage nach "Vor- und Nachname" (CN) muß der für den Client sichtbare vollqualifizierte Hostname (Hostname inkl. Domain) des FirstSpirit-Servers angegeben werden.

Um ein Zertifikat mit angegebenen Aliasnamen, in diesem Fall "jetty", aus dem Zertifikatsspeicher zu entfernen:

```
keytool -delete -alias jetty -keystore conf/fs-keystore.jks -storepass changeit
```

Auflisten aller Zertifikate:

```
keytool -list -v -keystore conf/fs-keystore.jks -storepass changeit
```

Anschließend erfolgt die Änderung der Webserver-Konfiguration des FirstSpirit-Servers (Kapitel 4.7.2.2 Seite 138).

Um das selbstsignierte Test-Zertifikat auf Seiten des FirstSpirit-JavaClients verwenden zu können, falls dieser nicht über Java Web Start aufgerufen wird oder das Zertifikat von Web Start nicht an die Java-VM übergeben werden kann, sind folgende Parameter beim Aufruf des JavaClients hinzuzufügen und die Zertifikatsdatei auf den Client-Rechner zu kopieren:

```
-Djavax.net.ssl.trustStore=pfad/zur/datei/fs-keystore.jks  
-Djavax.net.ssl.trustStorePassword=changeit
```



4.7.2 Vertrauenswürdiges Sicherheitszertifikat installieren

Damit ein Sicherheitszertifikat als vertrauenswürdig angesehen wird, muss es von einer offiziellen Zertifizierungsstelle (CA), z. B. <http://thawte.com>, digital unterschrieben (signiert) werden. Um ein solches Sicherheitszertifikat zu erstellen, sind für FirstSpirit 2 Verfahren möglich: entweder über `openssl`²⁰ oder über das bei Java mitgelieferte `keytool`. Über `openssl` erstellte Zertifikate haben den Vorteil, von beliebigen anderen Webservern wie Apache, IIS, Tomcat und anderen verwendet werden zu können. Ein über `keytool` erstelltes Zertifikat kann ausschließlich für Java-basierte Webserver verwendet werden.

4.7.2.1 Erstellen eines Sicherheitszertifikats über `keytool`

Zunächst wird ein privater Schlüssel erstellt. Dazu ist folgender Befehl einzugeben, nachdem in das Installationsverzeichnis des FirstSpirit-Servers gewechselt wurde:

```
keytool -genkey -keystore conf/fs-keystore.jks -storepass=mypass -  
alias fs5.yourdomain.net -keyalg RSA -keysize 1024 -validity 3650
```

Mit "keysize" wird die Schlüssellänge und mit "validity" die Gültigkeit in Tagen angegeben. Bei der Abfrage nach dem Key-Password sollte das gleiche wie bei "-storepass" angegeben werden. Bei der Abfrage nach "Vor- und Nachname" bzw. "first and last name" muß der vollqualifizierte Hostname angegeben werden, wie er vom Client aus sichtbar ist, z. B. `fs5.yourdomain.net`.

Im nächsten Schritt muss eine Zertifizierungsanforderung erzeugt werden:

```
keytool -certreq -keystore conf/fs-keystore.jks -storepass  
changeit -alias fs5.yourdomain.net -file request.csr
```

Die Datei "request.csr" wird anschließend der Zertifizierungsstelle zugesendet. Das erhaltene Antwortzertifikat (`public.pem`) muss in den Zertifikatsspeicher importiert werden:

```
keytool -import -trustcacerts -keystore conf/fs-keystore.jks \  
-storepass changeit -alias fs5.yourdomain.net -file public.pem
```

Falls die Zertifizierungsstelle Zertifikate über eine Hierarchie herausgibt (certificate chain), müssen alle Zertifikate der Hierarchie entweder schon im Zertifikatsspeicher der Java-VM vorliegen oder in den Zertifikatsspeicher importiert werden, ansonsten wird von `keytool` die Fehlermeldung "Failed to establish chain from reply"

²⁰ <http://www.openssl.org/>



ausgegeben. Zum Importieren der Zertifikate der Hierarchie muss der Befehl für jede Datei aufgerufen und dabei bei "-alias" z. B. immer der Dateiname angegeben werden:

```
keytool -import -trustcacerts -keystore conf/fs-keystore.jks -
storepass changeit -alias chain_CA_1 Zertifikat1 -file
chain_CA_1.pem
```

In einigen Fällen kann es vorkommen, dass die Zertifizierungsstellen Zertifikate versenden, die vom keytool nicht verstanden werden. Diese Zertifikate können mit OpenSSL umgewandelt werden, z. B.:

```
openssl x509 -in public.crt -out public.pem -outform PEM
```

4.7.2.2 Erstellen eines Sicherheitszertifikats über openssl

Zunächst den privaten Schlüssel erstellen:

```
openssl genrsa -out private.key 1024
```

Nun wird eine Zertifizierungsanforderung erstellt (`request.csr`), die anschließend an die Zertifizierungsstelle (CA) zum Signieren übergeben wird:

```
openssl req -new -key private.key -out request.csr
```

Die Zertifizierungsstelle liefert nun den öffentlichen signierten Schlüssel ("Zertifikat") zurück, meistens im PEM-Format als Textdatei (`public.pem`), die mit "-----BEGIN CERTIFICATE-----" beginnt. Privater und signierter öffentlicher Schlüssel müssen nun in einem Zertifikatsspeicher im PKCS12-Format für den FirstSpirit-Webserver zusammengefaßt werden. Dazu zunächst in das Installationsverzeichnis des FirstSpirit-Servers wechseln und nachfolgendes Kommando zum Erstellen des Zertifikatsspeichers eingeben. Dabei wird nach einem zu wählenden Passwort für den Zertifikatsspeicher gefragt. Dieses Passwort wird gleich in die Webserver-Konfiguration eingetragen. Im Beispiel ist dort "changeit" gewählt.

```
openssl pkcs12 -inkey private.key -in public.pem -export -out
conf/fs-keystore.p12 -caname root
```



4.7.2.3 Änderung der Webserver-Konfiguration des FirstSpirit-Servers

In der Konfigurationsdatei `fs-webapp.xml` muß der HTTPS-Listener aktiviert werden. Die Konfigurationsparameter sind in der Datei bereits eingetragen, aber durch Kommentarzeichen "`<!-- -->`" deaktiviert. Folgende Konfigurationsparameter sind notwendig (Bitte Groß-/Kleinschreibung beachten):

```
<!--
  HTTPS-Connector
  =====

  * for additional parameters read
  http://wiki.eclipse.org/Jetty/Howto/Configure_SSL
  * if NIO is not available, use
  org.eclipse.jetty.server.ssl.SslSocketConnector

-->
<New id="sslContextFactory"
class="org.eclipse.jetty.http.ssl.SslContextFactory">
<Set name="keyStore"><SystemProperty name="cmsroot" />/conf/fs-
keystore.p12</Set>
<Set name="keyStorePassword">PASSWORD</Set>
<Set name="keyStoreType">pkcs12</Set>
</New>
<Call name="addConnector">
<Arg><New class="org.eclipse.jetty.server.ssl.SslSelectChannelConnector">
<Arg><Ref id="sslContextFactory"/></Arg>
<Set name="Port">8443</Set>
<Set name="maxIdleTime">30000</Set>
<Set name="Acceptors">2</Set>
<Set name="AcceptQueueSize">100</Set>
</New></Arg>
</Call>
```

Bei "Password" handelt es sich um das Passwort des Zertifikatsspeichers, bei "KeyPassword" um das Passwort des Zertifikats (bzw. des privaten Schlüssels), das normalerweise dasselbe wie für den Zertifikatsspeicher ist.

Falls ein Keystore des Typs "JKS" verwendet werden soll, muss "pkcs12" durch "jks" ersetzt werden.

Unter Windows wird statt der Dateiendung `.p12` für pkcs12-Dateien auch die Endung `.pfx` verwendet.



Falls das HTTPS-Zertifikat Zwischenzertifikate benötigt (intermediate certificates), was mittlerweile eigentlich bei allen Zertifizierungsstellen üblich ist und diese noch dem Zertifikatsspeicher hinzugefügt werden müssen, ist folgendes Vorgehen zu empfehlen:

- Im Beispiel liegt der geheime Schlüssel und das erhaltene Server-Zertifikat, ohne die Zwischenzertifikate, in der Datei servercert.pfx im Format PKCS12 vor, was das übliche Format für Java-basierte Webserver ist.
- Zunächst muss der geheime Schlüssel inklusive des Serverzertifikats in das PEM-Format konvertiert werden:

```
openssl pkcs12 -export -in servercert.pfx -out serverkeycert.pem
```
- Nun mit einem Texteditor die Datei serverkeycert.pem öffnen und den enthaltenen geheimen Schlüssel (PRIVATE KEY) und das Zertifikate (CERTIFICATE) in jeweils einzelne Dateien serverkey.pem und servercert.pem kopieren, dabei jeweils den Inhalt zwischen den Zeilen "---- BEGIN..." bis "---- END..." und diese Zeilen selbst übernehmen, den Rest entfernen.
- Die Zwischenzertifikate liegen im Beispiel in den Dateien intermediate1.pem und intermediate2.pem vor, wobei sich die Zertifizierungskette insgesamt folgendermassen zusammensetzt: Zertifizierungsstelle (CA) -> Intermediate 1 -> Intermediate 2 -> Serverzertifikat
- Falls die einzelnen Zertifikate im Format DER vorliegen, z.B. mit Dateierdung .cer oder .crt, also nicht direkt lesbareren ASCII-Text mit Zeilen "---- BEGIN..." enthalten, müssen die einzelnen Dateien zunächst ins PEM-Format konvertiert werden:

```
openssl x509 -in intermediate1.cer -inform DER -out intermediate1.pem -outform PEM
```

```
openssl x509 -in intermediate2.cer -inform DER -out intermediate2.pem -outform PEM
```
- Alle Zertifikate werden jetzt in einer Datei zusammengefasst, **wobei die Reihenfolge eingehalten werden muss**, die der Zertifizierungskette von der Zertifizierungsstelle bis zum Serverzertifikat entspricht! Falls die Reihenfolge vertauscht wird, zeigt der Server beim Start keine Fehlermeldung an und erst beim Verbindungsaufbau durch Java Webstart wird ein Zertifikatsfehler eingeblendet.

```
cat intermediate1.pem intermediate2.pem servercert.pem > serverall.pem
```
- Zum Schluß werden der geheime Schlüssel und die Zertifikatskette in einer einzigen PKCS12-Datei zusammengefasst, dabei ein zufälliges Passwort eintragen, was dem in der Datei fs-webapp.xml entspricht:

```
openssl pkcs12 -export -inkey serverkey.pem -in
```



```
serverall.pem -out fs-keystore.p12 -passout pass:PASSWORD
```

- Folgende temporäre Dateien sollten wieder entfernt werden:
servercert.pem
serverkey.pem
serverkeycert.pem
serverall.pem
intermediate1.pem
intermediate2.pem
- Abschließend die Datei fs-keystore.p12 in den Ordner firstspirit5/conf verschieben.

Nach einem Neustart des FirstSpirit-Servers ist die Startseite nun zusätzlich über <https://fs5.yourdomain.net:8443> neben <http://fs5.yourdomain.net:8000> verfügbar.

Weitere Informationen zur HTTPS-Konfiguration des von FirstSpirit verwendeten Webservers Jetty finden Sie unter:

http://wiki.eclipse.org/Jetty/Howto/Configure_SSL



4.8 Datenbankanbindung

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

FirstSpirit stellt eine graphische Oberfläche zur Verfügung, die es dem Benutzer ermöglicht, strukturierte Datenbank-Tabellen anzulegen, zu modifizieren und Abfragen zu formulieren. Dazu wurde von FirstSpirit eine Datenbank-Abstraktionsschicht implementiert, die das universelle FirstSpirit Content Typsystem auf das konkret zu verwendende Datenbanksystem abbildet. Mithilfe dieser Architektur können direkt alle Datenbanken als Datenquellen-Verwaltung verwendet werden, für die eine Datenbankabstraktionsschicht implementiert ist (zurzeit: MySQL, Oracle, PostgreSQL, DB2, MS-SQL-Server, Derby). Genauere Informationen können dem aktuellen *Technischen Datenblatt* entnommen werden.

In der Regel kann ein Projektexport bzw. -import von einem Datenbanksystem auf ein anderes Datenbanksystem erfolgen.

4.8.1 Ablage der JDBC-Treiberdateien

Es bestehen zwei unterschiedliche Möglichkeiten, die JDBC-Treiberdateien in den FirstSpirit-Server zu integrieren:

1. Als FirstSpirit-Modul: (empfohlen)

Um verschiedene Versionen eines JDBC-Treibers gleichzeitig in verschiedenen FirstSpirit-Projekten verwenden zu können und den JDBC-Treiber im laufenden Betrieb des FirstSpirit-Servers ohne Neustart austauschen zu können, besteht die Möglichkeit der Integration der JDBC-Treiber als FirstSpirit-Modul. Hier wird die JAR-Datei des JDBC-Treibers inklusive einer zusätzlichen Deskriptor-Datei in einer FSM-Datei (ZIP-Archiv) zusammengefasst. In der Layer-Konfiguration wird über den zusätzlichen Parameter `module` (siehe Kapitel 4.8.4.2 Seite 159) dann auf dieses FirstSpirit-Modul verwiesen.

2. Über das Verzeichnis `shared/lib`:

Die entsprechenden Treiberdateien müssen in das Verzeichnis `firstspirit5/shared/lib` als JAR- oder ZIP-Dateien kopiert werden, damit sie im CLASSPATH der Java-VM des FirstSpirit-Servers liegen. Anschließend ist ein Neustart des FirstSpirit-Servers notwendig.





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

4.8.2 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 4.8.2.1 Seite 143)
- `module.xml`-Datei (siehe Kapitel 4.8.2.2 Seite 144)

optional:

- `web.xml`-Datei (siehe Kapitel 4.8.2.2 Seite 144)

Diese Dateien werden in einer FSM-Datei (ZIP-Archiv) zu einem FirstSpirit-Modul zusammengefasst (siehe Kapitel 4.8.2.3 Seite 149).

4.8.2.1 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).



Zu Einschränkungen und Hinweisen zu den Treibern der speziellen Datenbank-Typen siehe auch Kapitel 4.8.6 Seite 166 sowie die anderen Unterkapitel in Kapitel 4.8.



4.8.2.2 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 9.1**-Datenbank dar:

```
<module>
  <name>JDBC_PostgreSQL_9_1</name>
  <displayname>Database Driver PostgreSQL 9.1</displayname>
  <version>9.1.902.1</version>
  <description>JDBC Driver for PostgreSQL 9.1 databases</description>
  <vendor>PostgreSQL Global Development Group</vendor>

  <resources>
    <resource scope="module">lib/postgresql-9.1-
902.jdbc4.jar</resource>
  </resources>

  <components>
    <web-app>
      <name>WebApp_PostgreSQL_9_1</name>
      <displayname>WebApp PostgreSQL 9.1</displayname>
      <description>Provides the JDBC Driver in a web
application.</description>
      <web-xml>web.xml</web-xml>
      <web-resources>
        <resource scope="module"
          name="postgresql"
          version="9.1.902"
          minVersion="9.1.1"
          maxVersion="9.1.9999">lib/postgresql-9.1-
902.jdbc4.jar
        </resource>
      </web-resources>
    </web-app>
  </components>

  <configuration>
    <DRIVER>org.postgresql.Driver</DRIVER>
  </configuration>

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

Das Modul ist zweigeteilt: ein Teil definiert die Ressourcen für den FirstSpirit-Server, der andere für Webanwendungen (innerhalb des `<web-app>`-Tags). Dieser JDBC-Treiber kann somit im FirstSpirit-Server **und** in Webanwendungen 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, technischer Name für die Komponenten vergeben werden. Dabei dürfen nur Groß- bzw. Kleinbuchstaben (A-Z, a-z) und Zahlen (0-9) verwendet werden. Der Name wird beim Installieren des Moduls validiert. Module, die dieser Konvention nicht entsprechen, können nicht installiert werden.

Der technische Name wird beispielsweise zur Anzeige in der FirstSpirit Projekt- und Serverkonfiguration (sofern kein Anzeigename für das Modul definiert wurde), zur Überprüfung des Moduls bei der Aktualisierung und der Installation und zum Anlegen von Dateien und Ordnern auf der Festplatte verwendet. Der für die Server-Komponente vergebene Name muss auch in der Datenbank-Layer-Konfiguration angegeben werden (siehe Kapitel 4.8.3.2 Seite 151).

`<displayname>`: Über dieses Tag kann ein optionaler Anzeigename für das Modul vergeben werden (verfügbar ab Version 5.0R2). Sofern ein Anzeigename definiert ist, wird dieser in allen FirstSpirit-Oberflächen angezeigt, beispielsweise in der Modul-Übersicht des FirstSpirit-Servers (siehe Abbildung 4-7). Der für die Webanwendungs-Komponente vergebene Anzeigename wird auch in den Projekt-Eigenschaften, Bereich "Web-Komponenten" verwendet (siehe Abbildung 4-11). Das Pflicht-Attribut `<name>` wird weiterhin als eindeutiger, technischer Name verwendet. Ist kein Anzeigename definiert, wird der technische Name in den Oberflächen angezeigt.

`<description>`: Über dieses Tag kann eine Beschreibung zur Komponente 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 Webanwendungskomponente 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. *9.1.902* für Version 9.1 Build 902.

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 9.1.1 bis 9.1.999 genutzt werden kann. Wird nun ein zweiter Treiber durch ein weiteres Modul zur Verfügung gestellt, z. B. Build 903, so kann dieser auch von 9.1.1 bis 9.1.999 genutzt werden. In diesem Fall wird dann nur der höhere Treiber (also 903) in die Webanwendung kopiert bzw. übernommen.

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

<DRIVER>: Enthält den vollständigen Klassennamen des verwendeten JDBC-Treibers, beispielsweise *org.postgresql.Driver* für PostgreSQL. (Siehe auch Parameter *DRIVER* in Kapitel 4.8.4.1 Seite 158 sowie die Kapitel mit den datenbanktyp-spezifischen Beispielkonfigurationen in Kapitel 4.8.7 Seite 169.)

<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.

(Siehe auch Parameter *layerclass* in Kapitel 4.8.4.1 Seite 158)



sowie

die Kapitel mit den datenbanktyp-spezifischen Beispielkonfigurationen in Kapitel 4.8.7 Seite 169.)

Soll der JDBC-Treiber in einer Webanwendung zur Verfügung stehen, ist die Datei `web.xml` erforderlich (vgl. Kapitel 4.8.2.2 Seite 144):

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app id="JDBC_PostgreSQL_9_1"
  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 Webanwendungen eines Tomcat-Web-Servers verwendet werden, wird ebenfalls eine `module.xml`-Datei benötigt. Eine exemplarische `module.xml` könnte so aussehen:

```
<module>
  <name>JDBC_Derby_10</name>
  <version>10.8.2.2.1</version>
  <description>JDBC Driver for Derby 10.8 databases</description>
  <vendor>Apache Software Foundation</vendor>

  <resources>
    <resource scope="module">lib/derbyclient.jar</resource>
  </resources>

  <components>
    <web-app>
      <name>WebApp_Derby_10</name>
      <description>Provides the JDBC Driver in a web
application.</description>
      <web-xml>web.xml</web-xml>
      <web-resources>
        <resource scope="module"
          name="derby"
          version="10.8.2.2"
          minVersion="10.8.1"
          maxVersion="10.8.9999">lib/derbyclient.jar</resource>
      </web-resources>
    </web-app>
  </components>

  <configuration>
    <DRIVER>org.apache.derby.jdbc.ClientDriver</DRIVER>
  </configuration>

  <layerclass>de.espirit.or.impl.derby.DerbyLayer</layerclass>
```



```
</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="JDBC_Derby_10"  
  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 4.8.3.4.1 Seite 154).



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.



4.8.2.3 Verzeichnisstruktur eines JDBC-Treiber-Moduls

Wird der Treiber nur für den FirstSpirit-Server verwendet, müssen die Dateien in folgender Verzeichnisstruktur abgelegt werden (vgl. Kapitel 4.8.2.1 Seite 143 und Kapitel 4.8.2.2 Seite 144):

[-] jdbc_postgresql_9_1		
lib		
postgresql-9.1-902.jdbc4.jar	Executable Jar File	549 KB
META-INF		
module.xml	XML-Dokument	1 KB

Abbildung 4-5: Verzeichnisstruktur Server

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

[-] jdbc_postgresql_9_1		
web.xml	XML-Dokument	1 KB
lib		
postgresql-9.1-902.jdbc4.jar	Executable Jar File	549 KB
META-INF		
module.xml	XML-Dokument	1 KB

Abbildung 4-6: Verzeichnisstruktur Server und Webanwendung

Um nun ein gültiges FirstSpirit-Modul zu erhalten, muss aus den Inhalten des übergeordneten Ordners ("jdbc_postgresql_9_1") 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 4-6 der Ordnername als Dateiname übernommen, sollte die Modul-Datei schließlich `jdbc_postgresql_9_1.fsm` heißen.

Alternativ kann die Modul-Datei über folgenden Befehl erstellt werden:

```
jar cvf jdbc_postgresql_9_1.fsm -C jdbc_postgresql_9_1 .
```

Das Programm "jar" ist Bestandteil jedes JDK und je nach Betriebssystem und Installation beispielsweise unter

`c:\programme\jdk[Versionsnummer]\bin\jar.exe` oder
`/opt/jdk[Versionsnummer]/bin/jar` zu finden.



4.8.3 Installation und Konfiguration des JDBC-Treiber-Moduls

4.8.3.1 Installation des JDBC-Treiber-Moduls

Wurde das JDBC-Treiber-Modul erfolgreich wie in Kapitel 4.8.2 Seite 143 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 (siehe auch Kapitel 7.3.11 Seite 254). 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: JDBC_PostgreSQL_9_1, siehe Kapitel 4.8.2.2 Seite 144, Tag <name>) und Version (Tag <version>) angezeigt:

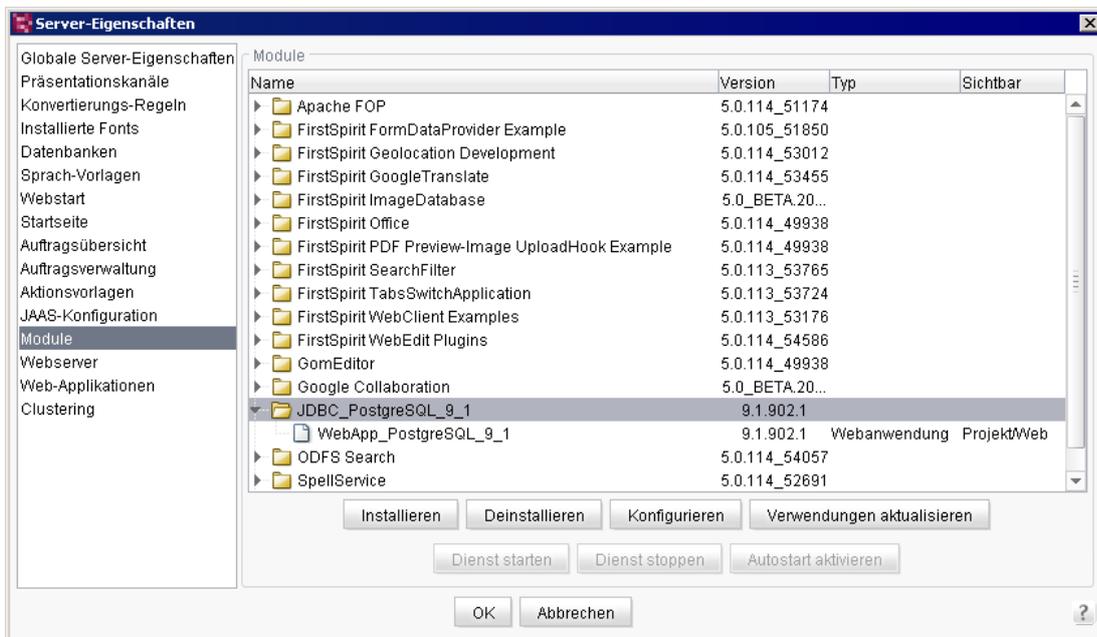


Abbildung 4-7: Server-Eigenschaften – JDBC-Treiber als Modul

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

Hier braucht keine weitere Konfiguration vorgenommen werden.



4.8.3.2 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 (siehe auch Kapitel 7.3.5 Seite 241):

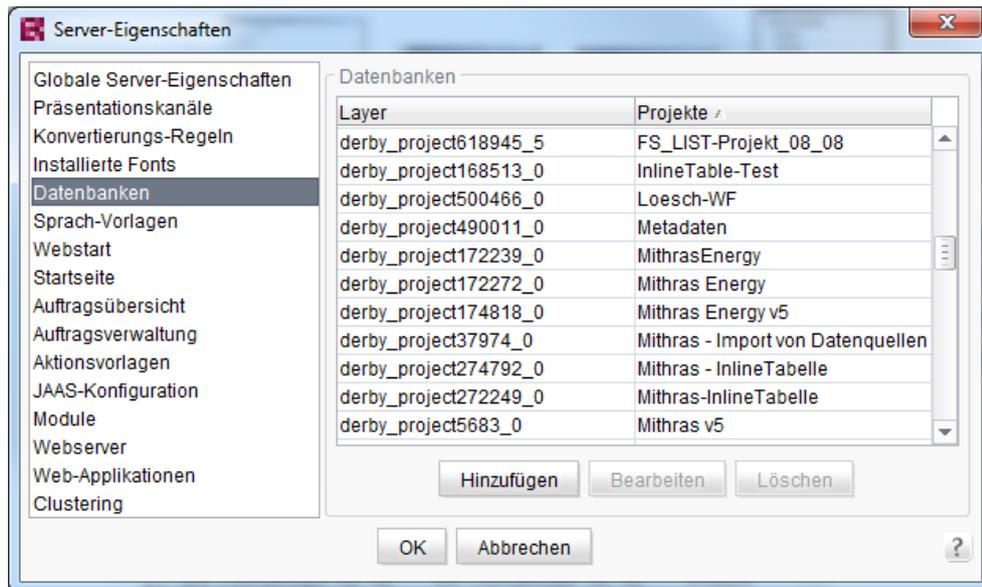


Abbildung 4-8: Server-Eigenschaften – Datenbanken

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 4-9: Datenbank-Konfiguration

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

```
module=JDBC_PostgreSQL_9_1
```



Abbildung 4-10: Datenbank-Konfiguration mit `module`-Parameter

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



4.8.3.3 Bei Verwendung in Webanwendungen

Soll die Datenbank in einer Webanwendung verwendet werden, muss das Modul in der gewünschten Web-Komponente hinzugefügt werden. Dies wird in den Projekt-Eigenschaften im Bereich "Web-Komponenten" vorgenommen (siehe dazu Kapitel 7.4.18 Seite 327). 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 4.8.2.2 Seite 144):

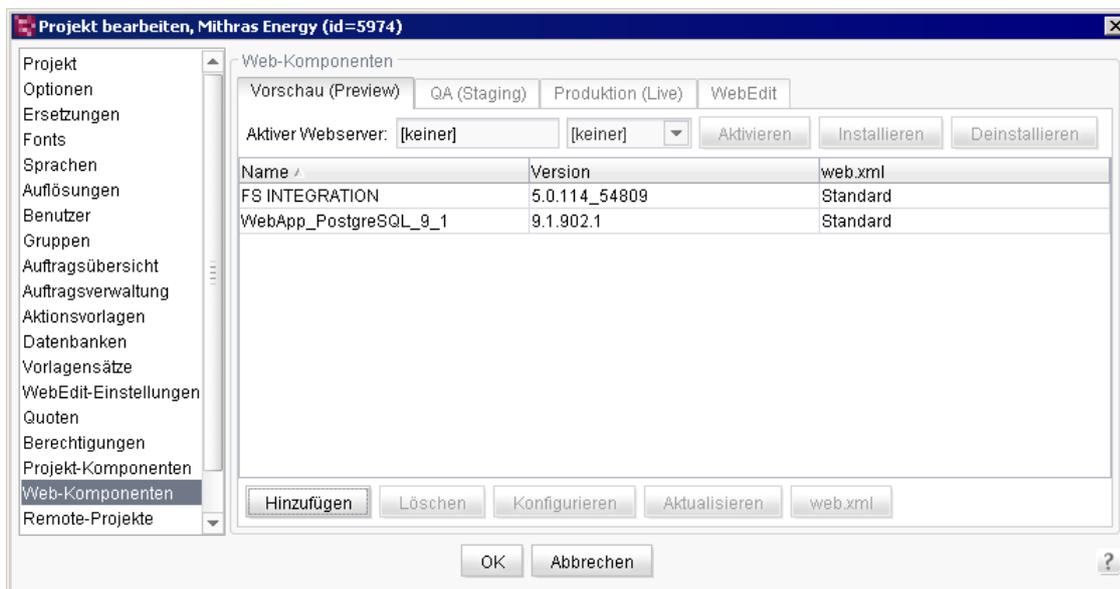


Abbildung 4-11: Projekt-Eigenschaften – JDBC-Treiber als Web-Komponente

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



4.8.3.4 Bei Verwendung der Derby-Datenbank in Webanwendungen

Wird eine Derby-Datenbank in Webanwendungen (z. B. im FirstSpirit-Modul `DynamicDatabaseAccess`) verwendet, muss das JDBC-Treiber-Modul ebenfalls, wie in Kapitel 4.8.3.3 Seite 153 beschrieben, der Webanwendung hinzugefügt werden.



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

4.8.3.4.1 Beispiel: Modul "FirstSpirit DynamicDatabaseAccess"

Bei der Verwendung des Moduls `FirstSpirit DynamicDatabaseAccess` 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 4.8.3.2 Seite 151) 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 `DynamicDatabaseAccess` muss die Konfiguration aus dem JDBC-Treiber-Modul nach diesen Änderungen manuell aktualisiert werden, wenn in der Datenbank-Verbindung die Option



"Benutzerspezifisch" aktiviert ist:

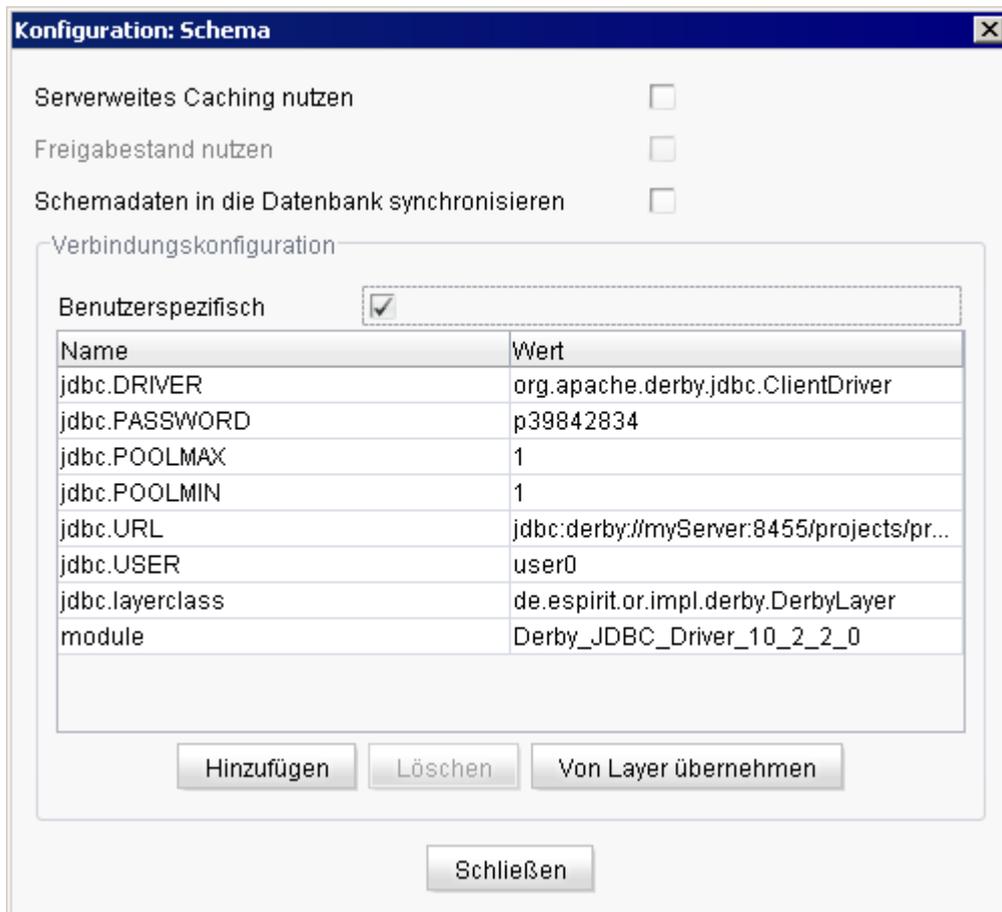


Abbildung 4-12: FirstSpirit DynamicDatabaseAccess – Datenbank-Konfiguration

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 4.8.3.2 Seite 151).

4.8.3.4.2 Individuelle Modul-Implementierung

Die Anpassungen, die für das Beispiel "FirstSpirit DynamicDatabaseAccess" (siehe Kapitel 4.8.3.2 Seite 151) vorgenommen werden müssen, müssen in jedem Modul, das individuell programmiert wurde und mit der Derby-Datenbank zusammenarbeitet, vorgenommen werden.



4.8.4 Konfiguration von Datenquellen

Die Konfiguration der Datenbankbindung des FirstSpirit-Servers erfolgt über die Server- und Projektkonfiguration (siehe Kapitel 7.3.5 Seite 241) oder direkt über den Eintrag "DATABASES" in der Konfigurationsdatei `fs-database.conf` (siehe Kapitel 4.3.3 Seite 76). Empfohlen wird das Bearbeiten der Datenbankkonfiguration über die Server- und Projektkonfiguration (siehe Kapitel 7.3.5 Seite 241), da hier alle Änderungen automatisch in die `fs-database.conf` geschrieben und auf dem Server aktualisiert werden. Außerdem ist es möglich, die konfigurierte Datenbankbindung zu testen (siehe Abbildung 7-36).

Die FirstSpirit Datenbankbindung kann für verschiedene Zwecke genutzt werden:

1. Anlegen von "internen Datenquellen"
2. Einbindung von externen Datenbanken ("externe Datenquellen")

Voraussetzung für die Nutzung von Datenquellen in FirstSpirit:

- Datenbank als externe oder interne Datenquelle:
Lizenz-Key mit dem Feature-Code "content" (`fs-license.conf`, Kap. 4.3.5 Seite 94)
- Datenbank-Konfiguration über die Server- und Projektkonfiguration (empfohlen) oder direkt über die Datei
`firstspirit5/conf/fs-database.conf`

Für einen FirstSpirit-Server können beliebig viele Datenquellen definiert werden, die über die Administratoren-Oberfläche den Projekten individuell zugeordnet bzw. beim Importieren eines Projektes ausgewählt werden können.

Für die direkte Konfiguration über die Datei `fs-database.conf` gilt:

- Es können mehrere Datenbanken kommasepariert hintereinander aufgelistet werden.
- Die Namen der externen Datenquellen können beliebig gewählt werden, wobei die folgenden Konventionen eingehalten werden müssen.
Erlaubte Zeichen: A-Z, a-z, 0-9, _, -

Beispiel:

```
DATABASES=content1, content2, extern1, extern2
```



Für jedes Element der in "DATABASES" angegebenen Datenbanken müssen zusätzlich folgende Parameter in der Konfiguration definiert werden (Schema):

```
<database>.jdbc.DRIVER=<JDBC-Driver-Class>
<database>.jdbc.URL=<JDBC-Connection-url>
<database>.jdbc.SCHEMA=<dbName>
<database>.jdbc.USER= <db_login>
<database>.jdbc.PASSWORD=<db_passwort>
<database>.jdbc.layerclass=<FirstSpirit-DB-Layer-Klasse>
```

Konfigurationsbeispiel (für DATABASE=content1):

```
content1.jdbc.DRIVER=com.mysql.jdbc.Driver
content1.jdbc.URL=jdbc:mysql://localhost:3306/mydb
content1.jdbc.SCHEMA=mydb
content1.jdbc.USER=cms
content1.jdbc.PASSWORD=cms
content1.jdbc.layerclass=de.espirit.or.impl.mysql.MySQLLayer
```

Beschreibung der Pflicht-Parameter siehe Kapitel 4.8.4.1 Seite 158.

Optional können folgende Parameter verwendet werden:

```
content1.jdbc.SCHEMA=dbName
content1.jdbc.CATALOG=dbCatalogname
content1.jdbc.POOLMIN=10
content1.jdbc.POOLMAX=15
content1.jdbc.POOLCYCLE=120
content1.jdbc.POOLTIMEOUT=240
content1.jdbc.CONNECTIONTIMEOUT=3600
content1.jdbc.CONNECTIONRETRY=3
content1.jdbc.CONNECTIONRETRYCYCLE=500
content1.jdbc.MAXSTRINGLENGTH=4000
```

Beschreibung der optionalen Parameter siehe Kapitel 4.8.4.2 Seite 159.



4.8.4.1 Beschreibung der Pflichtparameter

`<database>.jdbc.DRIVER`: Enthält den vollständigen Klassennamen des verwendeten JDBC-Treibers, der für die Definition einer Content-Datenbank zwingend notwendig ist (siehe Kapitel 4.8.1 Seite 142). Dabei muss sichergestellt werden, dass die angegebene Klasse auch vom FirstSpirit-Server geladen werden kann. Dazu muss das JAR-File, das den JDBC-Treiber enthält, in das Verzeichnis `shared/lib` des FirstSpirit-Servers abgelegt werden. Bei allen Änderungen in diesem Verzeichnis ist ein Neustart des Servers erforderlich. Es wird die Einbindung von JDBC-Treiberdateien als FirstSpirit-Modul empfohlen (siehe auch Kapitel 4.8.2 Seite 143).

```
content1.jdbc.DRIVER=com.mysql.jdbc.Driver
```

`<database>.jdbc.URL`: Enthält die Angabe des JDBC-URL zu einem Datenbank-Server und einer dort vorhandenen Datenbank, beispielsweise:

```
content1.jdbc.URL=jdbc:mysql://myServer:3306/mydb
```

Hier wird eine MySQL-Datenbank "mydb" auf dem Datenbank-Server "myServer" angesprochen. Das Aussehen der JDBC-Connection-URLs ist von Datenbank zu Datenbank unterschiedlich und muss der jeweiligen Dokumentation zur Datenbank entnommen werden (siehe Kapitel 4.8.7 Seite 169).

`<database>.jdbc.USER`: Gültiger Login-Name eines Datenbank-Users. Mit diesem Account baut der FirstSpirit-Server zur Laufzeit eine Verbindung zur Datenbank auf.

```
content1.jdbc.USER=db2admin
```

`<database>.jdbc.PASSWORD`: Gültiges Passwort zum Login unter `<database>.jdbc.USER`.

```
content1.jdbc.PASSWORD=admin
```

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

```
content1.jdbc.layerclass=de.espirit.or.impl.mysql.MySQLLayer
```





Der Parameter `layerclass` darf nicht leer sein, da es sonst zu Fehlern bei der Konfiguration der Datenbankanbindung kommt (siehe Kapitel 7.3.5 Seite 241).

Folgende Layer-Klassen gehören zum Standardlieferumfang von FirstSpirit:

- `de.espirit.or.impl.db2.DB2Layer`
- `de.espirit.or.impl.derby.DerbyLayer`
- `de.espirit.or.impl.mssql.MSSQL2005Layer`
- `de.espirit.or.impl.mssql.MSSQL2000Layer`
- `de.espirit.or.impl.mysql.MySQLLayer`
- `de.espirit.or.impl.oracle.OracleLayer`
- `de.espirit.or.impl.postgres.PostgreSQLLayer`



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

4.8.4.2 Beschreibung der optionalen Parameter

`<database>.jdbc.SCHEMA`: Dieser Parameter definiert das von FirstSpirit zu verwendende Schema auf dem DBMS (Datenbank-Management-System). Ein Schema wird häufig auch als "Datenbank" bezeichnet. Unter Oracle entspricht es einem normalen Benutzerkonto, bei anderen DBMS, z.B. PostgreSQL, kann ein normales Benutzerkonto auch mehrere Schemata umfassen.

Sofern dieser Parameter definiert ist, handelt es sich um einen **Standard-Layer**. In einem FirstSpirit-Projekt, dem ausschließlich Standard-Layer zugeordnet sind, kann ein FirstSpirit-Benutzer keine neuen zusätzlichen Schemata anlegen. Nur der FirstSpirit-Administrator kann dem Projekt weitere Layer hinzufügen.

Um das Anlegen weiterer Schemata auch FirstSpirit-Benutzern zu ermöglichen, wird ein sogenannter **DBA-Layer** benötigt, der bei den meisten DBMS aber DBA-Rechte (DBA = Database Administrator) benötigt. Bei einem DBA-Layer ist der `SCHEMA`-Parameter nicht eingetragen. Über einen DBA-Layer kann der FirstSpirit-Benutzer dann selbständig neue Standard-Layer erzeugen.





Vor FirstSpirit Version 4.2 wurden folgende Bezeichnungen verwendet:
Multiproject-Layer (sic): entspricht seit FirstSpirit Version 4.2 dem Begriff "Standard-Layer"
Singleproject-Layer (sic): entspricht seit FirstSpirit Version 4.2 dem Begriff "DBA-Layer"

Weitere Details zu den Unterschieden von Standard-Layer und DBA-Layer sowie deren Vor- und Nachteilen entnehmen Sie bitte dem "FirstSpirit-Handbuch für Entwickler (Grundlagen)", Kapitel "Datenquellen".

Beispiel:

```
database.jdbc.SCHEMA=warendatenbank
```

<database>.jdbc.POOLMAX: FirstSpirit nutzt für jede Verwendung, z. B. einen Schema-Knoten in einem Projekt, eine deployte Web-Applikation (z. B. FirstSpirit DynamicDatabaseAccess) usw., eine eigene Pool-Instanz.

POOLMAX definiert die Anzahl der ungenutzten DB-Verbindungen, die maximal im Pool verbleiben. Es gibt keinen Parameter, um die Anzahl der maximal parallel offenen Verbindungen zu einer Datenbank zu begrenzen.

```
content1.jdbc.POOLMAX=15
```

Ist kein Wert angegeben, wird die Anzahl der DB-Verbindungen auf den Wert von POOLMIN + 5 begrenzt.

<database>.jdbc.POOLMIN: POOLMIN definiert die minimale Anzahl der vorgehaltenen DB-Verbindungen pro Pool.

```
content1.jdbc.POOLMIN=10
```

Ist kein Wert angegeben, werden 5 DB-Verbindungen pro Pool vorgehalten.

<database>.jdbc.POOLCYCLE: Mit dem Parameter POOLCYCLE wird das Zeitintervall (in Sekunden) definiert, mit dem FirstSpirit abgelaufene DB-Verbindung aus dem Pool entfernt. Eine DB-Verbindung wird als abgelaufen betrachtet, wenn entweder POOLTIMEOUT oder CONNECTIONTIMEOUT abgelaufen ist. Wird kein Wert angegeben, wird der kleinste von FirstSpirit akzeptierte Wert – 90 Sekunden – gesetzt.



```
content1.jdbc.POOLCYCLE=120
```

<database>.jdbc.POOLTIMEOUT: Mit dem Parameter POOLTIMEOUT wird das Zeitintervall (in Sekunden) definiert, die der FirstSpirit-Server eine DB-Verbindung nutzen kann. Falls er die Verbindung nach Ablauf dieser Zeit nicht freigegeben hat, wird sie automatisch geschlossen. Wird kein Wert angegeben, wird standardmäßig der Wert "180" gesetzt.

```
content1.jdbc.POOLTIMEOUT=240
```

<database>.jdbc.CONNECTIONTIMEOUT: Mit dem Parameter CONNECTIONTIMEOUT wird das Zeitintervall (in Sekunden) definiert, nachdem eine DB-Verbindung vom FirstSpirit-Server als veraltet betrachtet und geschlossen wird. Wird kein Wert angegeben, wird standardmäßig ein Timeout von 30 Minuten (Wert 1800) verwendet. Bei einem Wert <= 0 wird der Timeout deaktiviert. Dieser Wert muss immer kleiner sein als das Idle-Timeout des Datenbank-Servers, das normalerweise einige Stunden beträgt.
Beispiel, um den Timeout auf 15 Minuten einzustellen:

```
content1.jdbc.CONNECTIONTIMEOUT=900
```

<database>.jdbc.CONNECTIONRETRY: Beim Ausführen einer SQL-Abfrage, versucht FirstSpirit eine Verbindung aus dem Connection-Pool zu benutzen. Falls keine freien Verbindungen vorhanden sind, wird versucht eine neue Verbindung zur Datenbank aufzubauen. Dabei können Anfragen abgewiesen werden (z. B. aufgrund der Datenbank-Konfiguration). Mit dem Parameter CONNECTIONRETRY wird die Anzahl der Verbindungsversuche zur Datenbank definiert. Wird die Anzahl überschritten, bricht ein fehlgeschlagener Verbindungsversuch mit einer Fehlermeldung ab. Wird kein Wert angegeben, wird standardmäßig der Wert "5" gesetzt.

```
content1.jdbc.CONNECTIONRETRY=3
```

<database>.jdbc.CONNECTIONRETRYCYCLE: Nach einem fehlgeschlagenen Versuch, eine Verbindung zur Datenbank aufzubauen, wartet der ORMMapper die in CONNECTIONRETRYCYCLE angegebene Zeit (in ms), bevor er einen neuen Versuch startet. Wird kein Wert angegeben, wird standardmäßig der Wert "300" gesetzt.

```
content1.jdbc.CONNECTIONRETRYCYCLE=500
```

<database>.jdbc.MAXSTRINGLENGTH: Der Parameter MAXSTRINGLENGTH legt die maximale Anzahl der Zeichen einer VARCHAR-Spalte beim



Erstellen einer neuen DB-Tabelle fest. Wenn für ein String-Attribut ein größerer Wert angegeben wird, als über den Parameter MAXSTRINGLENGTH definiert, wird dieses String-Attribut als BLOB oder CLOB in der Datenbank gespeichert. (Bei DB2 hängt dieser Wert von der Größe der verwendeten "Page Size" des verwendeten Tabellenbereichs ab.) Ist hier kein Wert gesetzt, wird abhängig von der verwendeten Datenbank ein Standardwert gesetzt:

- Derby: 32672
- MSSQL-Server: 4000
- Oracle: 2000
- PostgreSQL: 255
- Andere: 1024

```
content1.jdbc.MAXSTRINGLENGTH=4000
```

<database>.jdbc.JNDI: Läuft der ORMMapper innerhalb eines Web-Containers oder innerhalb eines Applikation-Servers, kann eine Datenbankverbindung über einen DataSource geholt werden. Damit nutzt der ORMMapper die Pooling-Kapazität des Web-Containers. Der Parameter JNDI legt den JNDI-Namen des benutzten DataSource fest.

```
database.jdbc.JNDI=java:comp/env/jdbc/ORMapper
```

<database>.jdbc.isolation: Durch parallel laufende Transaktionen, die Änderungen an Daten der Datenbank vornehmen, können undefinierte Zustände entstehen. Um zu vermeiden, dass eine laufende Transaktion durch eine parallel ablaufende Transaktion (durch eine Änderung der verwendeten Daten) in einen undefinierten Zustand versetzt wird, sollte eine Transaktionsisolation konfiguriert werden. Mithilfe des Parameters können unterschiedliche Isolationslevel²¹ konfiguriert werden. Generell unterstützt werden:

`READ_COMMITTED`: Niedrigstes Isolationslevel.

`REPEATABLE_READ`: Mittleres Isolationslevel.

`SERIALIZABLE`: Höchstes Isolationslevel (Standardwert).

Hinweis: Nicht jedes Datenbankmanagementsystem unterstützt alle Isolationslevel. So unterstützt beispielsweise eine OracleDB 11 nur die beiden Isolationslevel `READ_COMMITTED` und `SERIALIZABLE`,

²¹ Beschreibung siehe ANSI Standard bzw. Dokumentation zur verwendeten Datenbank



nicht aber `REPEATABLE_READ`. Im Einzelfall sollte die Dokumentation der entsprechenden Datenbank zu Rate gezogen werden.

```
database.jdbc.isolation=READ_COMMITTED
```

`module`: Wird ein FirstSpirit-Modul als JDBC-Treiber verwendet, muss über diesen Parameter der Name des JDBC-Treiber-Moduls angegeben werden. Zur Erstellung und Verwendung von JDBC-Treiber-Modulen siehe auch Kapitel 4.8.2 Seite 143.

```
database.module=JDBC_PostgreSQL_9_1
```

4.8.4.3 Beschreibung der Oracle-spezifischen Parameter

`<database>.jdbc.oracle.TABLESPACE`: Betrifft nur DBA-Layer-Konfigurationen. Über diesen Parameter muss ein Tablespace angegeben werden, z. B.

```
jdbc.oracle.TABLESPACE=USERS
```

4.8.4.4 Beschreibung der MS-SQL spezifischen Parameter

`<database>.jdbc.CATALOG`: Die Metadaten der Datenbanken werden in Namensräume gruppiert. Diese haben eine Baumstruktur mit den CATALOG Namen als Root-Knoten. Dieser Parameter schränkt den Namensraum für die Metadaten ein, mit welchen der ORMMapper arbeitet.

```
database.jdbc.CATALOG=ormapper
```



4.8.5 Notwendige Rechte für Datenbank-Benutzerkonten

Abhängig von der verwendeten Datenbank sind bestimmte Rechte notwendig, die in den folgenden Kapiteln beschrieben werden.

4.8.5.1 Oracle-Datenbanken

Für DBA-Layer:

```
CREATE USER <dbuser> IDENTIFIED BY "<password>";
GRANT DBA TO <dbuser>;
```

Für Standard-Layer:

```
CREATE USER <dbuser> IDENTIFIED BY "<password>";
GRANT CONNECT TO <dbuser>;
GRANT RESOURCE TO <dbuser>;
```

4.8.5.2 MySQL-Datenbanken

Für DBA-Layer:

```
# mysqladmin --default-character-set=utf8 create database <dbname>
# mysql
mysql> CREATE USER <dbuser> IDENTIFIED BY "<password>";
mysql> GRANT ALL PRIVILEGES ON *.* TO <dbuser>;
mysql> GRANT GRANT OPTION ON *.* TO <dbuser>;
```

Für Standard-Layer:

```
# mysqladmin --default-character-set=utf8 create database <dbname>
# mysql
mysql> CREATE USER <dbuser> IDENTIFIED BY "<password>";
mysql> GRANT ALL PRIVILEGES ON <dbname>.* TO <dbuser>;
```

Die Angabe der Zeichenkodierung UTF-8 ist erst ab MySQL Version 5 sinnvoll.

Die InnoDB Storage-Engine muss auf dem MySQL-Server aktiviert sein!

Empfehlenswerte MySQL-Server-Parameter für Produktionssysteme:

```
[mysqld]
set-variable=max_allowed_packet=4M
key_buffer_size=20M
sort_buffer_size=1M
query_cache_size=14M
innodb_buffer_pool_size=128M
```



4.8.5.3 PostgreSQL

Für DBA-Layer:

```
createdb -E UTF8 myDBname "my DB description text"  
createuser -D -A -P -E myDBuser  
psql -d myDBname -c "grant create on database myDBname to myDBuser;"
```

Für Standard-Layer:

```
createdb -E UTF8 myDBname "my DB description text"  
createuser -D -A -P -E myDBuser
```

Alle Abfragen bei `createuser` können mit "Nein" beantwortet werden, da weder bei DBA-Layer noch Standard-Layer erweiterte Benutzerrechte, außer denen über `grant create` für DBA-Layer zugewiesenen, notwendig sind.

In der Datei `/etc/postgres/pg_hba.conf` muss anschließend noch die Passwort-Authentifizierung (Typ MD5) für den angegebenen Benutzer auf die verwendete Datenbank eingetragen werden. Um diese Änderung dem Datenbank-Server bekannt zu machen, ist folgender Aufruf notwendig:

```
pg_ctl reload
```

4.8.5.4 IBM DB2

Erstellen der Datenbank (DBA-Layer und Standard-Layer):

```
db2 create database myDB using codeset utf-8 territory us pagesize 32 k  
  
db2 update db cfg for myDB using applheapsz 1024  
  
db2 connect to myDB  
  
db2 create schema myUser
```

Die letzte Zeile ist nur notwendig, wenn das Standard-Schema "myUser" der Datenbank "myDB" noch nicht vorhanden ist.

"myUser" ist der JDBC-Benutzername, der bei DB2 dem Instanznamen entspricht, also beispielsweise "db2inst1". Bei Bedarf kann aber auch ein anderer Schema-Name verwendet werden.

Folgende Berechtigungen sind notwendig für das DB2-Benutzerkonto:

Für DBA-Layer: DBADM

Für Standard-Layer: CONNECT, CREATETAB, BINDADD, IMPLICITSCHEMA



4.8.6 Hinweise und Einschränkungen zu den einzelnen Datenbanksystemen

Dieses Kapitel enthält Hinweise und Einschränkungen zum Einsatz der verschiedenen Datenbanksysteme, z. B. zur Verwendung von speziellen Treiber-Versionen, zur Konfiguration, der Unicode-Unterstützung, Einschränkungen hinsichtlich von Funktionen usw. Weitere Hinweise zu den einzelnen Datenbanksystemen finden sich auch im jeweiligen Unterkapitel in Kapitel 4.8.7 Seite 169.

4.8.6.1 Allgemeine Hinweise und Einschränkungen

Es wird empfohlen, die jeweils zur eingesetzten Datenbankversion passende JDBC-Treiber-Version zu verwenden, soweit nicht anders angegeben.

Einschränkungen

Bei einigen Datenbanken existieren Einschränkungen hinsichtlich der maximalen Länge von Bezeichnungen (speziell Spaltennamen) bzw. der Datenbank-Zeilenlänge. Bei der Erstellung von Content-Datenstrukturen sollten daher folgende Punkte beachtet werden:

1. Alle Texteingabefelder (o.ä.) sollen nur so groß wie unbedingt notwendig erzeugt werden.
2. Alle Spaltennamen sollten möglichst kurz gewählt werden
3. Sprachabhängige Eingabefelder sollten nur dann verwendet werden, wenn dies auch wirklich benötigt wird.
4. Nicht jede Datenbank ist in der Lage, Unicode-Zeichen im UTF-8 Format zu speichern. Wenn Sie vorhaben, multilinguale Projekte mit Unicode-Zeichen zu erstellen, so achten Sie darauf, eine unicodefähige und entsprechend konfigurierte Datenbank zu benutzen.
5. Die Spaltenanzahl sollte möglichst gering gehalten werden.



4.8.6.2 MySQL

Unicode: Unicode-Unterstützung ab MySQL-Version 5.

Weitere Einschränkungen für MySQL-Datenbanken (V 4.x - 5.1):

Große Tabellen können nicht gespeichert werden:

- <http://dev.mysql.com/doc/refman/5.1/en/innodb-restrictions.html>
- <http://bugs.mysql.com/bug.php?id=30295>

"The maximum row length, except for VARBINARY, VARCHAR, BLOB and TEXT columns, is slightly less than half of a database page. That is, the maximum row length is about 8000 bytes... InnoDB stores the first 768 bytes of a VARBINARY, VARCHAR, BLOB, or TEXT column in the row, and the rest into separate pages."

Das bedeutet: Eine Tabelle mit 11 Spalten vom Typ TEXT oder VARCHAR (>730) ist für MySQL bereits zu gross. Diese Einschränkung trifft auf die folgenden Beispiele zu:

- 4 Sprachen mit 2 DOM-Eingabekomponenten und einer sprachabhängigen Spalte (länger als ca. 230 Zeichen) oder
- 2 Sprachen je 5 DOM-Eingabekomponenten plus 1 sprachunabhängige Spalte (länger als 320 Zeichen) oder
- 1 Sprache mit 11 Spalten (je länger als 730 Zeichen)

4.8.6.3 Oracle



Das Speichern von Datenbank-Schemata und Änderungen an diesen können bei der Verwendung von Oracle-Datenbanken einige Zeit in Anspruch nehmen.

Unicode: Bei der Installation einer Oracle-Datenbank sollte die UNICODE Unterstützung aktiviert werden, um alle internationalen Zeichen darstellen zu können. Beim Anlegen der Oracle-Instanz müssen dazu beim Create-Database-Statement die folgenden Parameter gesetzt werden:

```
NLS_CHARACTERSET: UTF8  
NLS_NCHAR_CHARACTERSET: AL32UTF8
```

Die Verwendung von UTF-16 ist ebenfalls möglich, führt aber bei einigen selten verwendeten Sonderzeichen zu Problemen, da die Umsetzungstabelle von Oracle



für diese Kodierung anscheinend Lücken aufweist.

Treiber: 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.

4.8.6.4 IBM DB2

Unicode: Bei der Erstellung einer DB2-Datenbank sollte die UNICODE Unterstützung aktiviert werden.

Löschen von Spalten: Bei der Verwendung von DB2 ist es nicht möglich, über den JDBC-Treiber Spalten zu löschen. Die Spalten können zwar im Datenbankschema des FirstSpirit-JavaClients entfernt werden, bleiben aber in der Datenbank erhalten.

Sonstige Anmerkungen: Die Größe des Heaps von DB2 ist standardmäßig zu klein konfiguriert (128 x 4KB) und sollte mindestens auf eine Größe von 1024x4kB konfiguriert werden. Empfohlen wird die Ausführung des folgenden Statements auf der DB2-Konsole:

```
db2 update db cfg for myDB using applheapsz 1024
```



4.8.7 Beispiele für die Anbindung verschiedener Datenbanksysteme

Die Unterschiede bei der Anbindung der verschiedenen DBMS werden hier im Detail anhand von Beispielen gezeigt. Falls ein DBMS noch vor einer FirstSpirit-Installation vorbereitet werden soll und Unklarheiten gegenüber dem Datenbank-Administrator bezüglich der Zugangsparameter oder notwendigen Treiber-Dateien bestehen, hilft meistens, die Datenbank-Anbindung mittels eines externen JDBC-Clients zu testen. Dafür bietet sich z. B. der DB-Visualizer von <http://www.minq.se/products/dbvis/> an.

Die folgenden Konfigurationsbeispiele zeigen die Verwendung als Standard-Layer. Bei Verwendung als DBA-Layer wird die Zeile `jdbc.SCHEMA=...` weggelassen.

4.8.7.1 Konfigurationsbeispiel: MySQL

Treiber: `mysql-connector-java-x.x-bin.jar`

```
jdbc.DRIVER=com.mysql.jdbc.Driver
jdbc.URL=jdbc:mysql://localhost:3306/dbname?useUnicode=true&characterEncoding=UTF8
jdbc.USER=cms
jdbc.PASSWORD=cmspw
jdbc.layerclass=de.espirit.or.impl.mysql.MySQLLayer
jdbc.SCHEMA=dbname
```

Die Angabe von `?useUnicode=true&characterEncoding=UTF8` bei `jdbc.URL` ist nur bei Verwendung einer UTF-8-kodierten Datenbank (ab MySQL 5) notwendig. Bei Verwendung der Standard-MySQL-Kodierung (latin1, ISO-8859-1) ist die Angabe dieser Parameter nicht notwendig.

Mit FirstSpirit muss auch für MySQL 5.0 der Connector/J 5.1 verwendet werden, mindestens in Version 5.1.10: <http://dev.mysql.com/downloads/connector/j/5.1.html>.

4.8.7.2 Konfigurationsbeispiel: MS-SQL-Server

Treiber: `sqljdbc-x.x.jar`

```
jdbc.CATALOG=testDB
jdbc.DRIVER=com.microsoft.sqlserver.jdbc.SQLServerDriver
jdbc.PASSWORD=testpasswort
jdbc.URL=jdbc:microsoft:sqlserver://myserver:1433;DATABASENAME=testDB;selectMethod=cursor
jdbc.USER=testuser
jdbc.layerclass=de.espirit.or.impl.mssql.MSSQL2000Layer
```

bzw.



```
jdbc.CATALOG=testDB
jdbc.DRIVER=com.microsoft.sqlserver.jdbc.SQLServerDriver
jdbc.PASSWORD=testpasswort
jdbc.URL=jdbc:microsoft:sqlserver://myserver:1433;DATABASENAME=testDB;selectMethod=cursor
jdbc.USER=testuser
jdbc.layerclass=de.espirit.or.impl.mssql.MSSQL2005Layer
```

Zur Beschreibung der MS-SQL spezifischen Parameter siehe Kapitel 4.8.4.4 Seite 163.

4.8.7.3 Konfigurationsbeispiel: Oracle

Treiber: ojdbc14_x.x.jar

Layerparameter:

```
jdbc.DRIVER=oracle.jdbc.OracleDriver
jdbc.URL=jdbc:oracle:thin:@myserver:1521:ORCL
jdbc.USER=cms
jdbc.PASSWORD=cmspw
jdbc.layerclass=de.espirit.or.impl.oracle.OracleLayer
jdbc.SCHEMA=cms
```

Bei URL wird als letzter Parameter (im Beispiel ORCL) der Instanz-Name des Oracle-Servers eingetragen und nicht der Schema-Name. Der bei `jdbc.SCHEMA` angegebene Schema-Name entspricht dem bei `jdbc.USER` angegebenen Benutzernamen.

4.8.7.4 Konfigurationsbeispiel: PostgreSQL

Treiber: postgresql-x.x.jdbc3.jar

```
jdbc.DRIVER=org.postgresql.Driver
jdbc.URL=jdbc:postgresql://myServer:5432/myDB
jdbc.USER=cms
jdbc.PASSWORD=cmspw
jdbc.layerclass=de.espirit.or.impl.postgres.PostgreSQLLayer
jdbc.SCHEMA=public
```

Bei Verwendung als Standard-Layer muss bei PostgreSQL als Parameter `jdbc.SCHEMA` der Wert `public` eingetragen werden und nicht der Datenbankname.



4.8.7.5 Konfigurationsbeispiel: DB2

Layerparameter:

Zur Konfiguration einer IBM DB2 Datenbank muss ein JDBC Type 4 Treiber eingesetzt werden:

```
jdbc.DRIVER=com.ibm.db2.jcc.DB2Driver
jdbc.layerclass=de.espirit.or.impl.db2.DB2Layer
jdbc.URL=jdbc:db2://myServer:50000/myDB
jdbc.USER=myUser
jdbc.PASSWORD=myPassword
jdbc.SCHEMA=myDB
```

Treiber: db2java.zip (muss exakt zum verwendeten DB2-Server passen)

Port: DB2-Java-Connector (wird bereitgestellt über: db2jstrt PORTNUMMER).

Bei Verwendung als Standard-Layer wird bei DB2 als Parameter `jdbc.SCHEMA` derselbe Wert wie bei `jdbc.USER` eingetragen, falls das Standardschema der angegebenen Datenbank (hier "myDB") verwendet werden soll. Optional kann ein anderes Schema verwendet werden. In beiden Fällen muss das Schema zuvor ausserhalb von FirstSpirit über den SQL-Befehl "create schema myUser;" erstellt werden.

Treiber:

- seit DB 9.5: db2jcc4.jar, db2jcc_license_cu.jar
- vor DB 9.5: db2jcc.jar, db2jcc_license_cu.jar

Die JAR-Dateien liegen auf dem DB2-Server im Verzeichnis `db2inst1/sqlllib/java`. Um Inkompatibilitäten zu vermeiden, sollte immer die vom DB2-Server mitgelieferte Treiberversion verwendet werden.

Port: DB2-Connector (db2jstrt ist für Type 4 nicht notwendig). Die Portnummer kann unter Unix aus der Datei `/etc/services` bei Eintrag `db2_db2inst` abgelesen werden und ist standardmässig auf 50000 eingestellt.



4.8.7.6 Konfigurationsbeispiel: Interne Apache Derby-Datenbank

Für Testsysteme enthält der FirstSpirit-Server bereits ein einfaches relationales Datenbanksystem (Apache Derby). Normalerweise legt FirstSpirit alle Daten im Dateisystem (Berkeley-Datenbank) ab und nur – je nach Projektanforderung – einige wenige in relationalen Datenbanken. Standardmäßig wird beim Anlegen eines neuen Projekts diese Standard-Datenbank für das Projekt aktiviert (siehe Kapitel 7.2.3.1 Seite 209) und schreibender Zugriff auf die Datenbank für dieses Projekt gesetzt (siehe Kapitel 7.4.12 Seite 312). Nachträglich kann eine Derby-Datenbank in den Server-Eigenschaften angelegt (siehe Kapitel 7.3.5 Seite 241) und in den Projekt-Eigenschaften dem Projekt hinzugefügt werden (siehe Kapitel 7.4.12 Seite 312).

Um die Derby-Datenbank von externen Prozessen aus zu benutzen, z. B. Webanwendung mit FirstSpirit-Modul `DynamicDatabaseAccess` im externen Application-Server, muss zunächst der JDBC-Konnektor für Netzverbindungen aktiviert werden (siehe Kapitel 4.3.1.13 Seite 56). Als Verbindungsparameter in der Webanwendung werden die Parameter aus den Datenbank-Einstellungen des jeweiligen Projektes zunächst kopiert, z. B.

```
jdbc.URL=jdbc:derby:projects/project_12345/derby;create=true
jdbc.DRIVER=org.apache.derby.jdbc.EmbeddedDriver
jdbc.USER=testuser
jdbc.PASSWORD=testpassword
jdbc.POOLMAX=1
jdbc.POOLMIN=1
jdbc.layerclass=de.espirit.or.impl.derby.DerbyLayer
```

Anschliessend in den Verbindungsparametern für die Webanwendung die Zeilen `jdbc.URL` und `jdbc.DRIVER` durch folgende ersetzen, dabei statt "fs5server" den Hostnamen des FirstSpirit-Servers eintragen und die Projekt-ID 12345 durch die tatsächliche ersetzen:

```
jdbc.URL=jdbc:derby://fs5server:1527/projects/project_12345/derby
jdbc.DRIVER=org.apache.derby.jdbc.ClientDriver
```

Der JDBC-Treiber zur Einbindung in die Webanwendung kann als Datei `derbyclient.jar` unter <http://db.apache.org/derby/> heruntergeladen werden, um dann nach `WEB-INF/lib` oder in ein globales `classpath`-Verzeichnis des Application-Servers kopiert zu werden. Die jeweils in FirstSpirit aktive Version der Derby-Datenbank kann in der Log-Datei `firstspirit5/log/fs-database.log` abgelesen werden. Die aktuell in FirstSpirit verwendet Derby-Version ist 10.8.2.2.

Um die JDBC-Konfiguration für die einzelnen Webanwendungen automatisch zu erstellen, kann man diese geänderte Datenbank-Konfiguration auch in der Layer-



Einstellungen des FirstSpirit-Servers direkt vornehmen. Dann muss zusätzlich `derbyclient.jar` nach `firstspirit5/shared/lib` kopiert werden.

Falls in der externen Webanwendung auf genau 1 Schema zugegriffen werden soll, gegebenenfalls `jdbc.URL` um den Parameter `DATABASENAME` erweitern. `{SCHEMA-ID}` und `{PROJECT-ID}` dabei jeweils dem betreffenden FirstSpirit-Projekt entnehmen.

```
jdbc.URL=jdbc:derby://fs5server:1527/projects/project_12345/derby;
DATABASENAME=P{SCHEMA-ID}_{PROJECT-ID}
```

Es wird empfohlen, JDBC-Treiber statt in `firstspirit5/shared/lib` und manuell in `WEB-INF/lib` als FirstSpirit-Modul einzubinden, so dass sie automatisch in alle FirstSpirit-Webanwendungen integriert werden. Dann muss zusätzlich der Parameter `module=JDBC-Modulname` angegeben werden. Zur Erstellung und Verwendung von JDBC-Treiber-Modulen siehe auch Kapitel 4.8.2 Seite 143.

4.8.8 Vorgehen bei der Anbindung externer Datenbanken

1. Server- und Projektkonfiguration: In den **Server-Eigenschaften** (Menüeintrag "Server" / "Eigenschaften" / "Datenbanken") wird eine neue Datenbankanbindung konfiguriert (siehe Kapitel 7.3.5 Seite 241). Für eine externe MySQL-Datenbank wird beispielsweise folgender Eintrag konfiguriert:

```
jdbc.DRIVER=com.mysql.jdbc.Driver
jdbc.URL=jdbc:mysql://dbserver:3306/mydb
jdbc.USER=cms
jdbc.PASSWORD=cms
jdbc.SCHEMA=mydb
jdbc.layerclass=de.espirit.or.impl.mysql.MySQLLayer
```

2. Server- und Projektkonfiguration: In den **Projekteigenschaften** (Menüeintrag "Datenbanken") müssen für jedes Projekt, das die neu konfigurierte Datenbank (siehe 1.) verwenden soll, die Checkboxen "Ausgewählt", "Kein Schema Sync" und "Schreibgeschützt" für die entsprechende Datenbank aktiviert werden (siehe Kapitel 7.4.12 Seite 312) (vgl. Abbildung 4-13 "Konfiguration einer internen Derby-Datenbank und einer externen Datenbank").

Datenbanken			
Name	Ausgewählt	Schreibgeschützt	Kein Schema Sync
derby_project98514_0	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
externe_Datenbank	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Abbildung 4-13: Konfiguration einer internen (Derby) und einer externen Datenbank



Die Struktur und die Inhalte der externen Datenbank dürfen nicht verändert werden. Im Gegensatz zu internen Datenbanken ist für externe Datenbanken nur ein lesender aber kein schreibender Zugriff möglich.

3. Innerhalb der **Vorlagen-Verwaltung** des Projekts, für das die Datenbank "Ausgewählt" wurde (siehe 2.), wird nun das Kontextmenü auf dem Ordner "Datenbank-Schema" aufgerufen. Über den Eintrag "Schema aus Datenbank erzeugen" wird nun die entsprechende Datenbank ausgewählt und damit ein Datenbank-Schema auf Basis der externen Datenbank für dieses Projekt erzeugt (siehe "FirstSpirit Dokumentation für Entwickler").
Abhängig von der Anzahl der Tabellen in der Datenbank kann die Darstellung dieser Tabellen im Schema (Vorlagen-Verwaltung) ein wenig Zeit in Anspruch nehmen.
Sollte das Schema nicht automatisch entsperrt werden, beispielsweise weil in der Datenbank fehlerhafte Tabellen entdeckt wurden, muss das Projekt neu geöffnet werden, ohne das Schema zuvor manuell zu entsperren.
4. Im neuen Schema können nun die gewünschten Tabellenvorlagen und Tabellenabfragen angelegt werden (siehe "FirstSpirit Dokumentation für Entwickler").



4.9 Ausrollprozess für native Applikationen

4.9.1 Ausrollprozess (Server)

Im Rahmen der Funktionalität "Integrierte Vorschau" wird eine Mozilla-Engine in FirstSpirit verwendet. Die dafür benötigten nativen Systemkomponenten müssen zunächst auf die Arbeitsplatzsysteme der Redakteure verteilt werden.

Die Datei `fs-server.jar` enthält dazu eine aktuelle Version aller plattformabhängigen, nativen Bestandteile der Browserintegration (Client-Applikationen). Die Client-Applikationen werden beim Serverstart standardmäßig in das Verzeichnis `~FirstSpirit5\data\clientapp` entpackt. Das Verzeichnis für den Ausrollprozess der Client-Applikations-Dateien kann über den Parameter `CLIENTAPP_PATH` in der `fs-server.conf` konfiguriert werden (siehe Kapitel 4.3.1.2 Seite 35).

Jede Client-Applikation liegt anschließend in einem individuellen Verzeichnis unterhalb von `~FirstSpirit5\data\clientapp` (Standardeinstellung), beispielsweise die Mozilla-Engine in Windows-Betriebssystemen unter:

```
~FirstSpirit5\data\clientapp\jxbrowser\windows\lib
```

Eine Applikation besteht aus betriebssystemabhängigen Dateien und gemeinsamen Dateien. Die gemeinsamen Dateien befinden sich im Verzeichnis "common". Die betriebssystemabhängigen Dateien befinden sich in separaten Betriebssystem-spezifischen Verzeichnissen, beispielsweise `clientapp\jxbrowser\windows` oder `clientapp\jxbrowser\osx`.

Server-Versionsverwaltung: Gibt es im jeweiligen Applikations-Verzeichnis eine Datei `version.txt`, steht die Applikation unter Server-Versionsverwaltung und wird, falls notwendig, bei jedem Server-Start aktualisiert.

Wird die `version.txt` gelöscht, wird die Applikation nicht mehr vom Server aktualisiert und muss manuell verwaltet werden.

Um manuelle Änderungen vornehmen zu können, z. B. um Plug-ins zu installieren oder vorhandene problematische Plug-ins zu deaktivieren, müssen folgende Schritte durchgeführt werden:

1. FirstSpirit-Server stoppen
2. Datei `version.txt`



(unter `FirstSpirit5\data\clientapp\jxbrowser`) entfernen

3. Gewünschte Änderungen unterhalb des Verzeichnisses `\jxbrowser` vornehmen, z. B. Dateien unterhalb von `~FirstSpirit5\data\clientapp\jxbrowser\...\plugins` hinzufügen oder entfernen
4. FirstSpirit-Server starten

Beim nächsten Start eines JavaClients werden die Änderungen dort automatisch übernommen.

Falls später die Änderungen entfernt werden sollen, also wenn die mit FirstSpirit ausgelieferte Version der Applikation wieder verwendet werden soll:

1. FirstSpirit-Server stoppen
2. Verzeichnis `~FirstSpirit5\data\clientapp\jxbrowser` entfernen
3. FirstSpirit-Server starten

Beim nächsten Start eines JavaClients wird wieder die Standard-Version der Applikation verwendet.

4.9.2 Ausrollprozess (Arbeitsplatzrechner)

Die Verzeichnisse, die beim Ausrollen der nativen Systemkomponenten auf dem Server angelegt werden (siehe Kapitel 4.9.1 Seite 175), werden als exakte Kopie im User-Home-Verzeichnis des Arbeitsplatzrechners der Redakteure angelegt. Dazu werden automatisch alle Komponenten identifiziert, die zum aktuellen Betriebssystem des Arbeitsplatzrechners passen. Diese Komponenten werden auf dem Arbeitsplatzrechner in das User-Home-Verzeichnis des Redakteurs `~\Dokumente und Einstellungen\<USER>\.firstspirit_<FirstSpirit-Major-Version>.<FirstSpirit-Minor-Version><FirstSpirit-Release-Version>` ausgerollt. Alle zugehörigen Client-Applikationen liegen anschließend unterhalb dieses Verzeichnisses, beispielsweise die Mozilla-Engine in Windows-Betriebssystemen unter:

```
C:\Dokumente und  
Einstellungen\<USER>\.firstspirit_5.0R1\jxbrowser\xulrunner
```

Das Verzeichnis, in das die Client-Applikationen ausgerollt werden, kann über die Parameter `CLIENT_HOME_DIR` oder `CLIENT_HOME_DIR_WINDOWS` in der Datei `fs-server.conf` oder in den Verbindungseinstellungen definiert werden (siehe Kapitel 4.3.1.2 Seite 35 bzw. Kapitel 6.3.4.1 Seite 192 und Kapitel 4.9.5 Seite 179).



Die Auswertungsreihenfolge ist dabei folgende:

1. Zunächst werden betriebssystemspezifische Pfadangaben, die in den **Verbindungseinstellungen** gesetzt sind, ausgewertet (z. B. `CLIENT_HOME_DIR_WINDOWS`).
2. Dann werden Pfadangaben, die in den **Verbindungseinstellungen** über den Parameter `CLIENT_HOME_DIR` gesetzt sind, ausgewertet.
3. Dann werden betriebssystemspezifische Pfadangaben, die in der `fs-server.conf` definiert sind, ausgewertet (z. B. `CLIENT_HOME_DIR_WINDOWS`).
4. Dann werden Pfadangaben, die in der Datei `fs-server.conf` über den Parameter `CLIENT_HOME_DIR` gesetzt sind, ausgewertet.
5. Ist der Parameter weder in den Verbindungseinstellungen noch in der `fs-server.conf` gesetzt, wird standardmäßig das betriebssystemspezifische User-Home-Verzeichnis verwendet.

Die Angaben, die serverweit für alle Benutzer mithilfe der `fs-server.conf` getroffen werden, können damit benutzerspezifisch überschrieben werden.



Wird ein Verzeichnis angegeben, auf das der jeweilige Benutzer über keine Zugriffsrechte verfügt, wird eine entsprechende Exception ausgegeben. Die jeweilige Client-Applikation wird dann nicht ausgerollt und kann nicht verwendet werden. Siehe dazu auch Kapitel 4.9.5 Seite 179.



4.9.3 Aktualisierung der nativen Systemkomponenten

Eine weitere Zielsetzung des Ausrollprozesses für native Applikationen ist die zentrale Konfiguration und Aktualisierung der einzelnen Komponenten. Beispielsweise können Proxy-Einstellungen oder Plug-ins für die Mozilla-Engine zentral verwaltet und auf die einzelnen Arbeitsplatzrechner verteilt werden.

Die Verzeichnissynchronisation erfolgt über Hash-Werte, die pro Client-Applikation den kompletten (Betriebssystem-spezifischen) Teilbaum eindeutig beschreiben. Dieser Wert wird gecached und in einer CRC.TXT-Datei gespeichert, die im jeweiligen Applikations-Verzeichnis liegt. Diese Datei gibt es auf Client und Server.

Serverseitige Änderungen: Bei jeder Konfigurationsänderung auf dem Server wird ein neuer Hash-Wert berechnet, der dem Client ermöglicht zu erkennen, dass eine Änderung vorgenommen wurde:

- Wird nach einer Änderung der Server neu gestartet, werden die Hash-Werte automatisch neu berechnet. Beim Aufbau einer Verbindung zu einem Client-Rechner wird anschließend der Hash-Wert mit den Dateien des Arbeitsplatzrechners abgeglichen. Bei einer Abweichung erfolgt ein Dateisystem-Transfer der geänderten Client-Applikationen vom Server auf den betroffenen Arbeitsplatzrechner. Dabei werden nur geänderte Dateien übertragen.
- Bei manuellen Änderungen der Client-Applikations-Dateien auf dem Server muss der Hash-Wert manuell entfernt werden. Dazu kann die CRC.TXT-Datei im Betriebssystem-spezifischen Client-Applikations-Verzeichnis des Servers gelöscht werden, beispielsweise in Windows-Betriebssystemen unter:

```
~FirstSpirit5\data\clientapp\jxbrowser\windows\crc.txt
```

Beim Aufbau einer Verbindung zu einem Client-Rechner erkennt der Server, dass die Datei fehlt und berechnet automatisch einen neuer Hash-Wert. Dieser Hash-Wert wird mit den Dateien des Arbeitsplatzrechners abgeglichen und anschließend wird die Aktualisierung gestartet.

Clientseitige Änderungen: Die Aktualisierung der nativen Systemkomponenten erfolgt nur in eine Richtung, vom Server auf die Client-Rechner. Das bedeutet, bei einer manuellen Änderung der clientseitigen Client-Applikations-Dateien wird der clientseitige Hash-Wert nicht neu berechnet. Beim nächsten Neustart des Clients bleiben die manuellen Änderungen damit erhalten und werden nicht mit den serverseitigen Dateien synchronisiert (da beide Hash-Werte identisch sind).

Um versehentliche Veränderung der clientseitigen Client-Applikations-Dateien auf den Stand des Servers zurückzusetzen, muss zunächst die CRC.TXT-Datei im betriebssystem-spezifischen Client-Applikations-Verzeichnis des



Arbeitsplatzrechners gelöscht werden, beispielsweise in Windows-Betriebssystemen unter:

```
C:\Dokumente und  
Einstellungen\
```

Beim Aufbau einer Verbindung zu einem Server-Rechner erkennt der Server nun, dass die Datei fehlt und startet automatisch den Dateisystem-Transfer der geänderten Client-Applikationen vom Server auf den betroffenen Arbeitsplatzrechner.

4.9.4 Überschreiben von Dateien beim Ausrollprozess unterbinden

Bei jedem Update der nativen Systemkomponenten auf dem Server (über die Datei `fs-server.jar`) und jeder darauf erfolgenden Aktualisierung der Arbeitsplatzrechner können bestehende Konfigurationseinstellungen überschrieben werden. Ist das Überschreiben nicht erwünscht, muss für diese Dateien ein Dateisystem-Schreibschutz gesetzt werden. Das bedeutet, alle clientseitig oder serverseitig geänderten Dateien können mit einem Schreibschutz versehen werden und sind dann bei einer Aktualisierung vor Überschreiben geschützt. Bei der Aktualisierung der Dateien wird in diesem Fall eine Warnung als Hinweis für den Benutzer geloggt.

4.9.5 Anforderungen an das Ausrollverzeichnis

Folgende Anforderungen müssen für das Verzeichnis erfüllt sein, in das die Komponenten der Browserintegration ausgerollt werden, damit die Integrierte Vorschau korrekt funktioniert:

1. Auf folgendem Verzeichnis benötigt der Benutzer Schreibrechte:
`.../.firstspirit/jxbrowser/jxProfile`
2. Auf folgenden Verzeichnissen benötigt der Benutzer das Recht "Datei Ausführen":
`.../.firstspirit/jxbrowser/lib`
`.../.firstspirit/jxbrowser/xulrunner`
3. Das User-Home-Verzeichnis (das Verzeichnis oberhalb von `.firstspirit`), sowie der gesamte Dateipfad bis zu `.firstspirit` (einschließlich), darf keine Sonderzeichen enthalten.

Ansonsten kann es zu Fehlermeldungen kommen, z. B.

```
Could not intialize browser!
```

und der Integrierte Browser wird nicht geöffnet.



5 Konfiguration der FirstSpirit Webanwendungen



Die gleichzeitige Anmeldung über einen Webbrowser (z. B. in mehreren Fenstern oder Tabs) auf mehreren FirstSpirit-Servern mit demselben Hostnamen (z. B. myServer:8200 und myServer:8400) wird nicht unterstützt.

5.1 Konfiguration der FirstSpirit Startseite (fs5root)

Die Default-Startseite des FirstSpirit-Servers befindet sich unter:

```
<cms_basedir>\web\fs5root\index.jsp
```

Zum Aufrufen der Startseite ist ein gültiger Login notwendig (siehe Kapitel 6 Seite 185). Die Startseite kann genutzt werden, um den Anwendern einen einfachen Zugriff auf die FirstSpirit-Clients zu ermöglichen (siehe Kapitel 6.3 Seite 187).

Die FirstSpirit Startseite kann über die FirstSpirit Server- und Projektkonfiguration konfiguriert und angepasst werden (siehe Kapitel 7.3.7 Seite 249).

Die Schnellstart-Einträge können ebenfalls über die FirstSpirit Server- und Projektkonfiguration konfiguriert werden (siehe Kapitel 7.3.8 Seite 250).

Außerdem besteht die Möglichkeit, die Verbindungseinstellungen benutzerspezifisch zu konfigurieren (siehe Kapitel 6.3.4.1 Seite 192).

5.2 Konfiguration des WebClients

Der WebClient bzw. WebEdit wurde als Ergänzung zum Redaktionssystem FirstSpirit-JavaClient entwickelt. Der WebClient-Modus stellt eine browserbasierte Oberfläche für die schnelle und unkomplizierte Pflege redaktioneller Inhalte zur Verfügung. Dabei können Autoren, die vielfältigen Funktionen der FirstSpirit Redaktionsumgebung sofort nutzen, da entgegen der Installation des FirstSpirit-JavaClients für den WebClient keine Java-Umgebung (JRE) benötigt wird. Der WebClient arbeitet aus technischer Sicht rein auf Basis von HTML und JavaScript.





Die Konfiguration des WebClients erfolgt u.a. über die Konfigurationsdatei `fs-server.conf` im Bereich "Web Applications" (siehe Kapitel 4.3.1.7 Seite 43) und "WEBedit configuration" (siehe Kapitel 4.3.1.14 Seite 57). Vor dem produktiven Einsatz des WebClients sollte die Konfiguration gemäß dieser Kapitel an die jeweiligen Erfordernisse angepasst werden.

Der WebClient steht auch als projektlokal, konfigurierbare Webanwendung zur Verfügung. Das bedeutet, für jedes Projekt kann über die Server- und Projektkonfiguration eine gesonderte WebClient-Instanz installiert, konfiguriert und aktiviert werden (siehe Kapitel 5.2.2 Seite 182).

Für den Einsatz des WebClients in FirstSpirit-Projekten müssen bestimmte Voraussetzungen erfüllt sein (siehe Kapitel 5.2.1 Seite 181 und Kapitel 5.2.3 Seite 184).

Alle Konfigurationsdateien sind für den Einsatz mit internem Jetty optimiert und sollten direkt nach der Installation von FirstSpirit lauffähig sein. Die Konfiguration sollte nur dann geändert werden, wenn anstelle des standardmäßig konfigurierten, internen Jetty ein externer Application-Server eingesetzt werden soll (siehe Kapitel 4.6 Seite 131).

Zu Funktionen, Einschränkungen und Erweiterungsmöglichkeiten des WebClients siehe auch FirstSpirit Online Dokumentation, Bereich "Plugin-Entwicklung" / "WebClient-Erweiterungen" (EN).

5.2.1 Projektvoraussetzungen für den Einsatz des WebClients

Folgende Randbedingungen müssen gegeben sein, um den WebClient-Modus in FirstSpirit-Projekten einsetzen zu können. Diese Bedingungen sollten vor dem Einsatz geprüft werden:

1. In FirstSpirit Version 5.0 werden nicht mehr alle Eingabekomponenten unterstützt, die im JavaClient zur Verfügung stehen. Teilweise weichen die Funktionen der Eingabekomponenten im WebClient auch von denen im JavaClient ab. Daher sollte vor Verwendung des WebClients genau geprüft werden, ob die im Projekt benötigten Eingabekomponenten vom WebClient unterstützt werden. Auch kundenspezifische Komponenten müssen genau geprüft werden.



2. Browser-Kompatibilität:
 - Welcher Browser wird eingesetzt?
 - Welche Sicherheits-Einstellungen werden im Unternehmen verwendet?
Läuft der WebClient in dieser Konfiguration?
3. Proxy-/Firewall-Konfiguration: Wird zwangsweise ein Proxy oder eine Firewall verwendet? Läuft der WebClient in dieser Konfiguration (speziell Refresh-Problem)?
4. Gesicherter Zugriff: kann der http/https-Port des FirstSpirit-Servers für den WebClient-Anwenderkreis gefahrlos (ggf. auch nach außen) freigeschaltet werden?
5. Bildschirmauflösung: mind.: 1024x768 Pixel

5.2.2 Der WebClient als projektlokale Anwendung

Der WebClient steht auch als projektlokal, konfigurierbare Webanwendung zur Verfügung. Das bedeutet über die Server- und Projektkonfiguration kann für jedes Projekt eine gesonderte WebClient-Instanz installiert, konfiguriert und aktiviert werden.

Im Bereich "Web-Komponenten" ist dazu der Bereich "WebEdit" vorhanden. Über die Registerkarte kann eine WebClient-Instanz für das gewählte Projekt konfiguriert werden (siehe Kapitel 7.4.18 Seite 327).

Der WebClient-Instanz können weitere Web-Komponenten hinzugefügt werden (z. B. FirstSpirit DynamicDatabaseAccess, FirstSpirit DynamicPersonalization). Diese Web-Komponenten können einzeln für diesen Webbereich ("WebEdit") konfiguriert werden. Dabei ist sowohl die Standard-Konfiguration über den Button "Konfiguration" als auch das manuelle Editieren der Datei "web.xml" möglich.



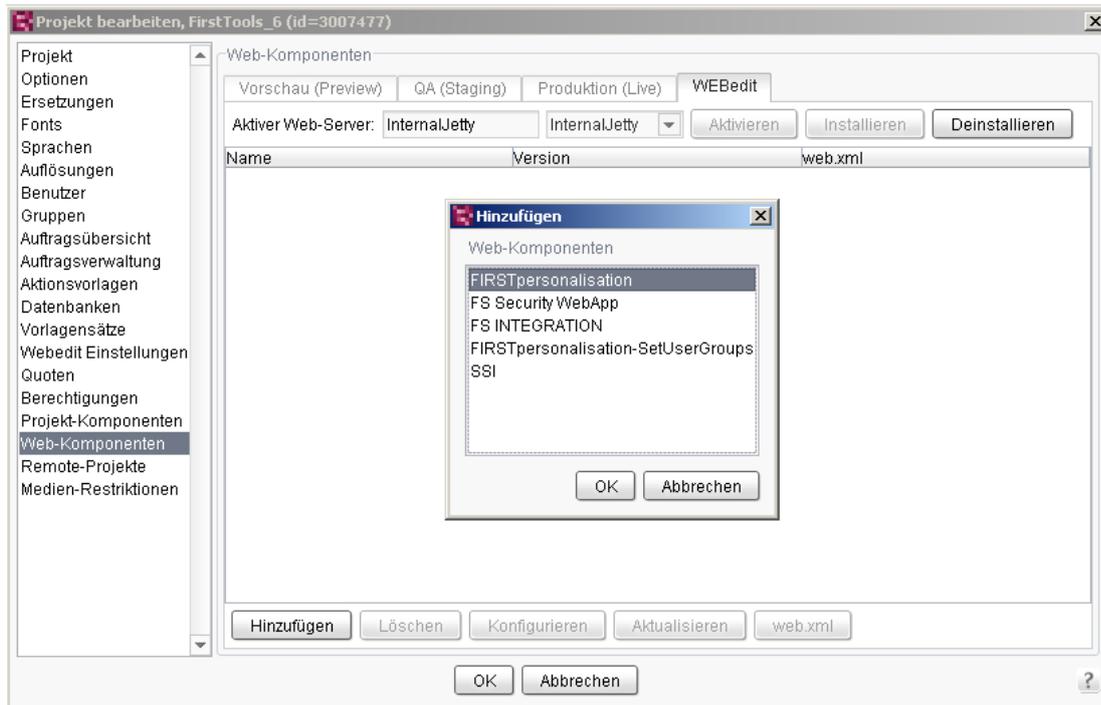


Abbildung 5-1: Der WebClient als projektlokale Webanwendung

Die projektlokale Konfiguration ermöglicht es, eine "erweiterte" WebClient-Instanz zu konfigurieren und auf dem gewünschten Webserver zu installieren und zu aktivieren. Dabei werden alle hinzugefügten Web-Komponenten berücksichtigt. Beim Installieren auf dem gewünschten Webserver werden alle erforderlichen Komponenten installiert und eine web.xml erzeugt, die alle bisher konfigurierten Dateien (web.xmls der einzelnen Komponenten) zu einer Datei zusammenfasst.

Damit kann nun für einzelne Projekte der WebClient mit einer projektspezifischen Personalisierungskonfiguration kombiniert werden.

Bei einer Aktualisierung auf dem Server, beispielsweise mit einer neuen WebClient-Version, ist das Aktualisieren aller projektlokalen WebClient-Instanzen möglich.



5.2.3 Browserkonfiguration für den Einsatz des WebClients

Beim Arbeiten mit dem Webbrowser Mozilla Firefox können Probleme beim Kopieren, Ausschneiden und Einfügen von Texten in die Eingabekomponente DOM-Editor (CMS_INPUT_DOM) auftreten. (Beispielsweise können Inhalte nicht mehr gespeichert oder mehrfach eingefügt werden.) Dieses Verhalten ist keine Fehlfunktion des WebClients, sondern eine Sicherheitseinstellung des Browsers. Firefox unterbindet aus sicherheitstechnischen Gründen (Standardeinstellung) das Einfügen bzw. Ändern von Inhalten aus der Zwischenablage mit JavaScript. Eine Aufbereitung solcher Inhalte ist jedoch für den DOM-Editor notwendig.

Die Funktion kann über die entsprechende Konfiguration der Browsereinstellungen (in der Datei "user.js") aktiviert werden.



Diese Einstellung sollte aus Sicherheitsgründen NICHT global (für alle URLs) vorgenommen werden, sondern nur für die benötigten URLs.

```
user_pref("capability.policy.allowclipboard.Clipboard.cutcopy", "allAccess");
user_pref("capability.policy.allowclipboard.Clipboard.paste", "allAccess");
user_pref("capability.policy.allowclipboard.sites", "http://aServer:port");
user_pref("capability.policy.policynames", "allowclipboard");
```

Für `capability.policy.allowclipboard.sites` können, durch Leerzeichen getrennt, auch mehrere Sites angegeben werden:

```
user_pref("capability.policy.allowclipboard.sites", "http://aServer:10000
http://aServer:11000");
```

Bei Problemen mit der Browserkonfiguration wenden Sie sich bitte an den Systemadministrator.



6 FirstSpirit Startseite

Der initiale Zugang zum FirstSpirit-Server erfolgt normalerweise über das Internet. Entsprechend den Einstellungen bei der Installation wird die Standardverbindung zum FirstSpirit-Server aufgebaut (siehe FirstSpirit Installationsanleitung).

Abhängig von der Konfiguration des Anmeldevorgangs kann die Anmeldung automatisch erfolgen (Kapitel 6.1 Seite 185) oder durch Angabe von Benutzername und Passwort (Kapitel 6.2 Seite 186). Die Konfiguration des Anmeldevorgangs erfolgt über die Konfigurationsdatei `fs-jaas.conf` (Parameter siehe Kapitel 4.3.4 Seite 77).



Die gleichzeitige Anmeldung über einen Webbrowser (z. B. in mehreren Fenstern oder Tabs) auf mehreren FirstSpirit-Servern mit demselben Hostnamen (z. B. `myServer:8200` und `myServer:8400`) wird nicht unterstützt.

6.1 Automatische Anmeldung über Single Sign-on

Befindet sich auf dem Server ein SSO-fähiges Login-Modul kann der Benutzer automatisch (z. B. mit seinem Windows-Login) auf dem FirstSpirit-Server authentifiziert werden. Dazu muss der Parameter `jaas.default` für SSO konfiguriert sein (Konfiguration JAAS²² siehe Kapitel 4.3.1.6 Seite 42). Die entsprechende Konfiguration kann über die FirstSpirit Server- und Projektkonfiguration vorgenommen werden (siehe Kapitel 7.3.10 Seite 253).

Bei Aufrufen der Startseite wird vom System überprüft, ob eine automatische Anmeldung möglich ist. Wenn sich auf dem Server ein SSO-fähiges Login-Modul befindet, wird der Benutzer automatisch unter seinem Windows-Login auf dem FirstSpirit-Server angemeldet und gelangt direkt zur Startseite (Kapitel 6.3 Seite 187). Ist der Benutzer noch nicht unter seinem Windows-Login auf dem Server registriert, dann wird er als externer Benutzer neu angelegt.

²² Java Authentication and Authorization Service
(Informationen: <http://www.oracle.com/technetwork/java/javase/jaas/index.html>)



6.2 Anmeldung mit Benutzername und Passwort

Schlägt die automatische Anmeldung fehl (oder ist die Anmeldung über SSO nicht konfiguriert), erscheint eine Anmeldeseite. Über das Anmeldefenster kann sich ein Benutzer am FirstSpirit-Server anmelden. Diese Anmeldung ist für alle Anwendungen auf dem Server gültig und bleibt eine gewisse Zeit auch für inaktive Benutzer erhalten.

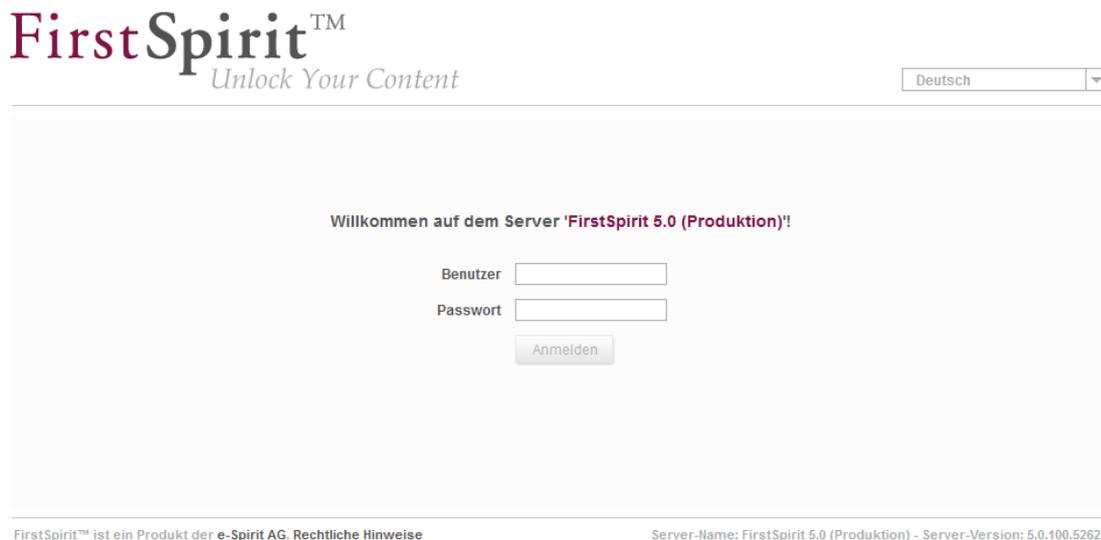


Abbildung 6-1: Anmeldung

Auf der Seite befinden sich Informationen über den Namen und die Version des FirstSpirit-Servers:

Server-Version: Die Version wird automatisch vom Server geliefert.

Servername: Name des FirstSpirit-Servers. Ist in der Konfigurationsdatei `fs-server.conf` ein symbolischer Name für den Server definiert, wird dieser auf der Startseite angezeigt (siehe Kapitel 4.3.1.1 Seite 32). Ist kein symbolischer Name definiert, dann wird der Hostname aus dem Zugriffspfad angezeigt, also beim Aufruf über <http://www.myServer.de:4050> beispielsweise der Servername "myServer".

 Über diese Auswahlbox kann eine Sprache für das weitere Arbeiten mit FirstSpirit festgelegt werden.

Benutzer: In diesem Feld wird der Benutzername eingetragen, unter dem der Benutzer auf dem FirstSpirit-Server angemeldet werden soll.

Passwort: In diesem Feld wird das Kennwort des Benutzers eingetragen.

Durch einen Klick auf die Schaltfläche "Anmelden" erfolgt die Anmeldung unter dem eingetragenen Benutzernamen.



6.3 FirstSpirit-Startseite

Nach der Anmeldung (automatisch oder manuell) öffnet sich die FirstSpirit-Startseite. Die Startseite ist in unterschiedliche Bereiche unterteilt, deren Sichtbarkeit von den Rechten des jeweiligen Benutzers abhängen:

- Bereich Schnellstart (siehe Kapitel 6.3.1)
- Bereich Clients (siehe Kapitel 6.3.2)
- Bereich Administration (siehe Kapitel 6.3.3)
- Bereich Benutzer (siehe Kapitel 6.3.4)

Zum Starten der FirstSpirit Server- und Projektkonfiguration und des FirstSpirit-JavaClients wird das Oracle Java Runtime Environment (JRE) benötigt, darin ist Java Web Start enthalten (Das JRE wird i.d.R. automatisch bei der Installation des JDKs installiert).

Genauere Informationen zu der benötigten Version des JRE entnehmen Sie bitte dem "Technischen Datenblatt".



Abbildung 6-2: FirstSpirit-Startseite

Auf der Seite befinden sich Informationen über den Namen und die Version des FirstSpirit-Servers. Außerdem wird angezeigt, welcher Benutzer gerade auf dem Server angemeldet ist.



Über diese Auswahlbox kann eine Sprache für das weitere Arbeiten mit FirstSpirit festgelegt werden.

(Zur Konfiguration der Webanwendung FirstSpirit Startseite siehe Kapitel 5, Seite



180 ff.)

6.3.1 Schnellstart

Im linken Bereich der Seite befinden sich die Schnellstart Einträge, die direkt mit einem Projekt verknüpft sind. Über diese Einträge wird automatisch der für den Eintrag konfigurierte Client (JavaClient oder WebClient) gestartet und das ausgewählte Projekt geöffnet. In dieser Liste werden nur die Projekte angezeigt, für die der angemeldete Benutzer das Recht hat, sie zu öffnen (Zur Konfiguration der Schnellstarteinträge siehe Kapitel 7.3.8 Seite 250).



6.3.2 Clientstart

Im mittleren Bereich der Seite befinden sich die Einträge zum Starten der FirstSpirit-Clients.

- **JavaClient (Redaktionsumgebung):** Mit einem Klick auf diesen Eintrag wird das FirstSpirit Redaktionssystem gestartet. Der Redakteur kann das gewünschte Projekt auswählen. Eine Verbindung zum Server wird automatisch hergestellt (siehe Kapitel 6.4).

Beim ersten Starten eines JavaClients kann folgender Bestätigungsdialog angezeigt werden:



Damit der Bestätigungsdialog nicht erneut angezeigt wird, ist die Option "Für Anwendungen dieses Anbieters nicht mehr anzeigen" zu aktivieren.

- **WebClient (Autorenumgebung):** Mit einem Klick auf diesen Eintrag wird innerhalb des Browsers die FirstSpirit Autorenumgebung gestartet. Die FirstSpirit Autorenumgebung bietet einen, gegenüber dem JavaClient, eingeschränkten Funktionsumfang zum redaktionellen Arbeiten mit FirstSpirit an (siehe Kapitel 6.4).



6.3.3 Administration

Der rechte Bereich der Seite ist noch einmal aufgeteilt. Im oberen Teil befinden sich die Einträge für Server- und/oder Projektadministratoren.

- **Server-Monitoring:** Durch einen Klick auf diesen Eintrag öffnet sich das Server-Monitoring. Eine ausführliche Beschreibung dazu befindet sich in Kapitel 8 (Seite 429).
- **Server- und Projektkonfiguration:** Durch einen Klick auf diesen Eintrag öffnet sich die FirstSpirit Server- und Projektkonfiguration. Eine ausführliche Beschreibung dazu befindet sich in Kapitel 7 (Seite 204).
Beim ersten Starten der FirstSpirit Server- und Projektkonfiguration kann folgender Bestätigungsdialog angezeigt werden:



Damit der Bestätigungsdialog nicht erneut angezeigt wird, ist die Option "Für Anwendungen dieses Anbieters nicht mehr anzeigen" zu aktivieren.



*Dieser Bereich ist nur für Server- und Projektadministratoren sichtbar.
(Wird ein Projekt deaktiviert, werden die Einträge für die Projektadministratoren des Projekts ausgeblendet.)*



6.3.4 Benutzer

Im unteren Teil des rechten Bereichs befinden sich die Einträge für die Benutzereinstellungen des aktuell angemeldeten Benutzers:

- **Verbindungseinstellungen:** Hier können die Verbindungseinstellungen des aktuell angemeldeten Benutzers geändert werden (siehe Kapitel 6.3.4.1 Seite 192).
- **Benutzer wechseln:** In einigen Fällen kann es gewünscht sein, sich unter einem anderen Benutzernamen am FirstSpirit-Server zu authentifizieren, beispielsweise um sich als Serveradministrator anzumelden (siehe Kapitel 6.3.4.3 Seite 197).
- **Passwort ändern:** Hier kann das Passwort des aktuell angemeldeten Benutzers geändert werden (siehe Kapitel 6.3.4.2 Seite 196).
- **Abmelden:** Ein Klick auf diesen Eintrag beendet die aktuelle FirstSpirit Sitzung für den angemeldeten Benutzer (siehe Kapitel 6.3.4.4 Seite 197).



6.3.4.1 Verbindungseinstellungen konfigurieren



☑ Einstellungen verwenden

Modus

HTTPS Protokoll verwenden

Host

Port

Speicher

Kompression

Verschlüsselung

Servlet-Zone

Optionale Parameter

Abbildung 6-3: Verbindungseinstellungen konfigurieren

An dieser Stelle werden die Verbindungseinstellungen des aktuell angemeldeten Benutzers für den Start des JavaClients und der Anwendung zur Server- und Projektkonfiguration angezeigt. Die hier konfigurierten Werte überschreiben die serverweit gültigen Webstart-Einstellungen für diesen Benutzer (vgl. Kapitel 7.3.7 Seite 249). Die Einstellungen sollten nur zu Testzwecken geändert werden.

Modus: In der Klappliste kann der Verbindungsmodus für die Standardkommunikation zwischen FirstSpirit-Clients und -Server für den aktuell eingeloggten Benutzer eingestellt werden:

- HTTP: normale Internetverbindung (Standardeinstellung)
- Socket: direkter Verbindungsmodus.

HTTPS Protokoll verwenden: Parameter, über den definiert wird, ob die Kommunikation im Verbindungsmodus HTTP über das sichere HTTPS-Protokoll übertragen wird.

Host: Servername oder IP-Adresse des FirstSpirit-Servers, mit dem sich der Client beim Webstart verbinden soll.

Port: Portnummer des FirstSpirit-Servers.

Speicher: Hier wird der Speicherplatz (in MB) angegeben, der für die virtuelle Maschine des Clients zur Verfügung gestellt werden soll.



Kompression: Kompression für die Kommunikation zwischen FirstSpirit-Clients und -Server für den aktuell eingeloggten Benutzer:

- Keine: keine Kompression bei der Übertragung von Daten zwischen Client und Server.
- Deflate: Verwendung des Deflate-Algorithmus mit Standard-Kompression bei der Übertragung von Daten zwischen Client und Server.
- Deflate Speed: Verwendung des Deflate-Algorithmus mit schnellster Kompression bei der Übertragung von Daten zwischen Client und Server.
- Deflate Best: Verwendung des Deflate-Algorithmus mit bester Kompression bei der Übertragung von Daten zwischen Client und Server.
- Snappy: Standardmäßige Voreinstellung für neu installierte FirstSpirit-Server. Der Kompressionsmodus "Snappy" steht für die Betriebssysteme (32 und 64 Bit) Microsoft Windows, Linux und Mac OS zur Verfügung. Als Fallback (z. B. auf nicht unterstützten Betriebssystemen) wird der Kompressionsmodus "Deflate Speed" verwendet.

Verschlüsselung: Verschlüsselung für die Kommunikation zwischen FirstSpirit-Clients und -Server für den aktuell eingeloggten Benutzer:

- Keine: keine Verschlüsselung bei der Übertragung von Daten zwischen Client und Server.
- TLS²³: Verwendung des TLS-Protokolls für die Übertragung von Daten zwischen Client und Server.
- DH ARC4: Verwendung des Verschlüsselungs-Algorithmus DH ARC4 für die Übertragung von Daten zwischen Client und Server.

Servlet-Zone: Angabe der Servlet-Zone (Die Servlet-Zone entspricht dem Mapping der Root-Applikations-URL (vgl. Parameter `WEBAPP_ROOT_URL` Kapitel 4.3.1.7 Seite 43)).

Optionale Parameter: In diesem Feld können optionale Parameter für die Webstart-Konfiguration gespeichert werden. Die Parameter können in der folgenden Form Semikolon-separiert hintereinander angegeben werden: `PARAMETER1=WERT1; PARAMETER2=WERT2`. Die optionalen Parameter entsprechen teilweise den Einstellungsmöglichkeiten, die innerhalb der Eingabefelder des Dialogs eingetragen werden können (vgl. z. B. Encryption, Compression):

- `compression`: Parameter für die Kompression (bei der Kommunikation zwischen FirstSpirit-Clients und -Server).

²³ Transport Layer Security



Mögliche Werte: 0 (keine), 1 (Deflate), 2 (Deflate Speed), 3 (Deflate Best)

Beispiel: `compression=3`

- `encryption`: Parameter für die Verschlüsselung (Kommunikation zwischen FirstSpirit-Clients und -Server).

Mögliche Werte: 0 (keine), 1 (TLS), 2 (DH ARC4)

Beispiel: `encryption=1`

- `login`: Angabe des Login-Moduls (plain, sso). Beispiel: `login=plain`
- `autologin`: Parameter für die Angabe von Benutzer-Login und -Passwort. Beide Werte werden im Klartext durch ":" getrennt übergeben. Dabei kann das Passwort ausgelassen werden, der Doppelpunkt muss jedoch immer übergeben werden.
- `host`: Angabe des Hostnamen (Vorauswahl des Servers).
- `port`: Angabe der Portnummer (Ganzzahl).
- `mode`: Angabe des Verbindungsmodus (`http`, `socket`)
- `httpproxy`: Parameter für die Angabe des Proxys, der im Verbindungsmodus HTTP verwendet werden soll. Ist dieser Parameter angegeben, wird ausschließlich dieser Proxy benutzt. Ist der Parameter nicht angegeben, wird versucht die Java-Webstart Proxy-Konfiguration auszuwerten. Diese Auswertung kann durch Angabe des Parameters `nohttpproxy=1` unterbunden werden.
- `httpsproxy`: Parameter für die Angabe des Proxys, der im Verbindungsmodus Socket verwendet werden soll. Der Proxy wird in diesem Fall verwendet, um die TCP-Verbindungen zu tunneln.
- `nohttpproxy`: Parameter, um die Auswertung der Java Web Start Proxy-Konfiguration zu verhindern (vgl. `httpproxy`).
Beispiel: `nohttpproxy=1`
- `usehttps`: Parameter, über den definiert wird, ob die Kommunikation im Verbindungsmodus HTTP über das sichere HTTPS-Protokoll übertragen wird (Wert=1) oder nicht (Wert=0).
Beispiel: `usehttps=1`
- `proxybypass`: Parameter, über den definiert wird, welche Hosts den Proxy umgehen dürfen. Im HTTP-Modus läuft die Kommunikation (für diese Hosts) dann nicht über einen Proxy. Mehrere Hosts können als Semikolon-separierte Liste übergeben werden. Bei Angabe eines (oder mehrerer) Hostnamen dürfen alle Hosts den Proxy umgehen, die mit dem angegebenen Hostnamen beginnen (siehe Bsp. `myServer_1`, `myServer_2`,...).
Beispiel: `proxybypass=myServer;localhost`
- `inlinebrowser.httpproxy`: Parameter zur Konfiguration des Inline-Browsers (für die integrierte Vorschau) für die Kommunikation über einen HTTP-Proxy (analog zur Proxy-Konfiguration des FirstSpirit JavaClient). Dazu müssen Servername oder IP-Adresse des Proxys und Port angegeben werden. Eine über



diesen Parameter vorgenommene HTTP-Proxy-Konfiguration hebt die lokalen Konfigurationseinstellungen des Browsers temporär auf, sie wird aber nicht dort gespeichert.

Beispiel: `inlinebrowser.httpproxy=myServer:8888`

- `CLIENT_HOME_DIR`: Parameter zur Angabe eines Verzeichnisses im Dateisystem, in das Client-Applikationen abgelegt werden sollen (siehe Kapitel 4.3.1.2 Seite 35 und Kapitel 4.9.2 Seite 176).
- `disableExpensiveCSOperations`: Über diesen Parameter können folgende Operationen für Datenquellen deaktiviert werden, die bei Datenquellen mit vielen Datensätzen (beispielsweise 1 Mio. Datensätze) speicher- und zeitintensiv sein können:
 - "Einfache Suche" im Suchdialog einer Datenquelle
 - "Erweiterte Suche" im Suchdialog einer Datenquelle
 - "Volltext Suche" im Suchdialog einer Datenquelle
 - Operation "alle Datensätze anzeigen" in der Datenquellenübersicht und im Auswahldialog der Eingabekomponenten (z. B. FS_DATASET)
 - Sortierungsfunktion der Spalten in der Datenquellenübersicht und im Auswahldialog der Eingabekomponenten (z. B. FS_DATASET)

Um diese Operationen für **alle** Datenquellen zu deaktivieren, ist der Parameter ohne Wert anzugeben:

```
disableExpensiveCSOperations=
```

Sollen die Operationen nur für **einzelne** Datenquellen deaktiviert werden, so sind die eindeutigen Bezeichner der gewünschten Datenquellen – durch Kommata getrennt – anzugeben:

```
disableExpensiveCSOperations=datenquelle1,datenquelle2
```

Die Deaktivierung einzelner Datenquellen hat keine Auswirkung bei der Verwendung der (veralteten) Eingabekomponenten `CMS_INPUT_CONTENTLIST`, `CMS_INPUT_OBJECTCHOOSE` und `CMS_INPUT_TABLIST`. Sollen die Einschränkungen auch für diese Eingabekomponenten gelten, so müssen diese auf alle Datenquellen erweitert werden.

Mit einem Klick auf den Button **Speichern** werden die geänderten Verbindungseinstellungen für den aktuell eingeloggten Benutzer gespeichert. Damit die Einstellungen aktiviert werden, muss die Checkbox "Einstellungen verwenden" aktiviert werden. Auf der Startseite wird dann die Information "Verbindungseinstellungen SIND AKTIVIERT" angezeigt.



6.3.4.2 Passwort ändern

Durch einen Klick auf diesen Eintrag kann ein Benutzer sein Passwort zur Anmeldung auf den FirstSpirit-Server ändern.



Passwort

Bitte geben sie ein neues Passwort für den Benutzer 'Admin' ein:

Neues Passwort

Passwort wiederholen

Ändern Abbrechen

Abbildung 6-4: Passwort ändern

Passwort: In diesem Feld muss noch einmal das aktuelle Passwort angegeben werden.

Neues Passwort: In diesem Feld wird das neue Passwort eingetragen.

Passwort wiederholen: Das neue Passwort wird in diesem Feld erneut eingetragen, um versehentliche Schreibfehler bei der Änderung des Passworts auszuschließen.

Durch einen Klick auf die Schaltfläche "Ändern" wird das neue Passwort für den angemeldeten Benutzer übernommen.

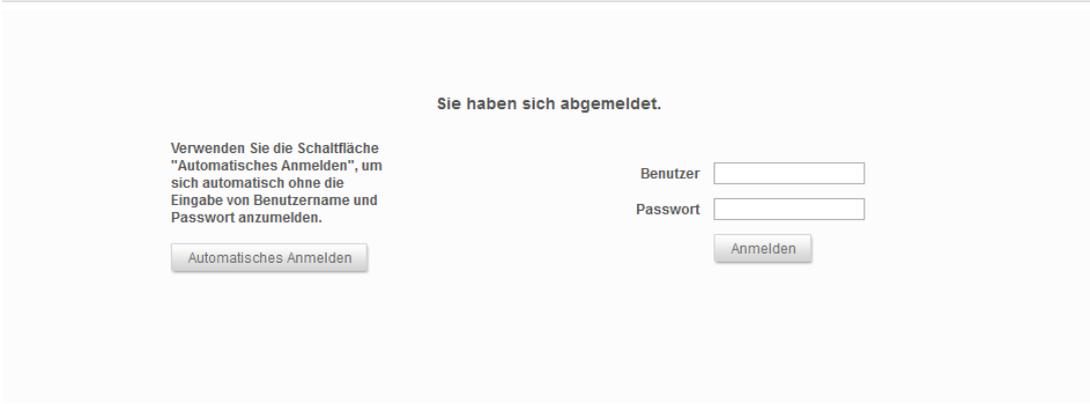


Dieser Eintrag steht nur Benutzern zur Verfügung, die manuell auf dem Server angelegt wurden, und nicht für externe Benutzer, die durch eine automatische SSO-Anmeldung angelegt wurden.



6.3.4.3 Benutzer wechseln

Durch einen Klick auf diesen Eintrag kann ein anderer Benutzer auf dem Server angemeldet werden. Es öffnet sich wieder die Anmeldeseite (siehe Kapitel 6.2 ab Seite 186), die nun um die Möglichkeit der Automatischen Anmeldung erweitert ist.



The screenshot shows the FirstSpirit login interface. At the top left is the logo 'FirstSpirit™' with the tagline 'Unlock Your Content'. To the right is a language dropdown menu set to 'Deutsch'. The main content area displays the message 'Sie haben sich abgemeldet.' (You have logged out). Below this, there is a text block explaining the 'Automatic Login' button: 'Verwenden Sie die Schaltfläche "Automatisches Anmelden", um sich automatisch ohne die Eingabe von Benutzername und Passwort anzumelden.' (Use the 'Automatic Login' button to log in automatically without entering username and password). To the right of this text are input fields for 'Benutzer' (Username) and 'Passwort' (Password), and an 'Anmelden' (Login) button. At the bottom left, there is a button labeled 'Automatisches Anmelden' (Automatic Login). At the bottom of the page, there is a footer with the text 'FirstSpirit™ ist ein Produkt der e-Spirit AG. Rechtliche Hinweise' and 'Server-Name: localhost - Server-Version: 5.0.100.52700'.

Abbildung 6-5: Java Web Start – Benutzer wechseln

Befindet sich auf dem Server ein SSO-fähiges Login-Modul, dann besteht nun auf der rechten Bildschirmseite die Möglichkeit, sich unter dem Windows-Login automatisch auf dem Server anzumelden. Durch einen Klick auf die Schaltfläche "Automatisches Anmelden" erfolgt die Anmeldung unter dem Windows-Login.

6.3.4.4 Abmelden

Durch einen Klick auf diesen Eintrag kann sich der angemeldete Benutzer von dem Server abmelden. Es öffnet sich dann wieder die auf Abbildung 6-5 zu sehende Anmeldeseite (siehe auch Kapitel 6.2 ab Seite 186).



6.4 Starten der Anwendungen

6.4.1 JavaClient

Für den Start des JavaClients wird ein Webbrowser mit "Java Web Start"²⁴ benötigt. Über Java Web Start werden Softwareaktualisierungen der FirstSpirit Produkte automatisch beim Start an die Client-Systeme weitergegeben²⁵. Hierfür sind unter anderem die benötigten Berechtigungskonfigurationen (z. B. Dateierstellungsrechte) auf System- oder Benutzer-Ebene vorzunehmen

Zum Starten des Clients genügt ein Klick auf den entsprechenden Eintrag auf der FirstSpirit-Startseite (siehe Kapitel 6.3 Seite 187).

Beim Starten des JavaClients öffnet sich nach dem Verbindungsaufbau ein Projektauswahldialog mit einer Liste der zur Verfügung stehenden Projekte für den angemeldeten Benutzer. Initial ist nach der Installation nur das Beispielprojekt vorhanden, wenn dieses bei der Installation des FirstSpirit-Servers nicht ausgewählt wurde.



Abbildung 6-6: Projekt auswählen

Mit einem Doppelklick auf den Eintrag oder mit einem Klick auf die Schaltfläche "OK", wird das ausgewählte Projekt geladen.

Sie können nun damit beginnen, sich mit dem FirstSpirit Client vertraut zu machen.

²⁴ Weitere Informationen: <http://www.oracle.com/technetwork/java/javase/javawebstart/index.html>

²⁵ Prinzip: http://de.wikipedia.org/wiki/Java_Web_Start



6.4.2 WebClient

Der WebClient benötigt keine Java-Umgebung und kann direkt über den Webbrowser aufgerufen werden. Zum Starten des Clients genügt ein Klick auf den entsprechenden Eintrag auf der FirstSpirit-Startseite (siehe Kapitel 6.3 Seite 187).

Beim Starten des WebClients öffnet sich nach dem Verbindungsaufbau ein Projektauswahldialog mit einer Liste der für den angemeldeten Benutzer zur Verfügung stehenden Projekte. Um in dieser Liste zu erscheinen, muss ein Projekt für die Bearbeitung im WebClient aktiviert werden (siehe Kapitel 7.4.14 Seite 316). Initial ist nur das Beispielprojekt vorhanden, wenn dieses bei der Installation des FirstSpirit-Servers nicht abgewählt wurde.

6.4.3 Server-Monitoring

Das Server-Monitoring benötigt keine Java-Umgebung und kann direkt über den Webbrowser aufgerufen werden. Zum Starten des Clients genügt ein Klick auf den entsprechenden Eintrag im Bereich Administration auf der FirstSpirit-Startseite (siehe Kapitel 6.3 Seite 187). Hier kann beispielsweise die Lizenzdatei einfach installiert werden (siehe Kapitel 8.6.1.2 Seite 451).

6.4.4 Server- und Projektkonfiguration

Für den Start der Server- und Projektkonfiguration wird ein Webbrowser mit "Java Web Start"²⁶ benötigt. Über Java Web Start werden Softwareaktualisierungen der FirstSpirit Produkte automatisch beim Start an die Client-Systeme weitergegeben²⁷. Hierfür sind unter anderem die benötigten Berechtigungskonfigurationen (z. B. Dateierstellungsrechte) auf System- oder Benutzer-Ebene vorzunehmen.

Zum Starten des Clients genügt ein Klick auf den entsprechenden Eintrag im Bereich Administration auf der FirstSpirit-Startseite (siehe Kapitel 6.3 Seite 187).

²⁶ Weitere Informationen: <http://www.oracle.com/technetwork/java/javase/javawebstart/index.html>

²⁷ Prinzip: http://de.wikipedia.org/wiki/Java_Web_Start



6.5 FirstSpirit-Client als JAVA-Anwendung starten

Erfolgt der Zugang zum FirstSpirit-Server nicht über das Internet, sondern beispielsweise über eine Kommandozeile, können im folgenden Verbindungsdialog Parameter zur Kommunikation zwischen JavaClient und FirstSpirit-Server, ähnlich wie bei die Verbindungseinstellungen in Kapitel 6.3.4.1 ab Seite 192, konfiguriert werden.

6.5.1 Socket-Modus

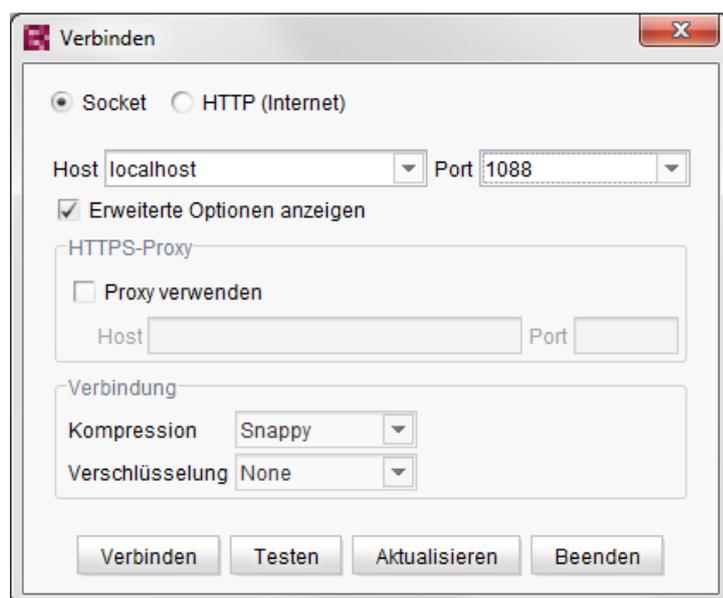


Abbildung 6-7: Verbindungsdialog für den JavaClient – Socket-Modus

Host: Servername oder IP-Adresse des FirstSpirit-Servers, mit dem sich der Client beim Web-Start verbinden soll.

Port: Portnummer des FirstSpirit-Servers.

Wird die Checkbox "Erweiterte Optionen anzeigen" aktiviert, können folgende Parameter eingestellt werden:

Proxy verwenden: Ist diese Checkbox aktiviert, kann über die darunterliegenden Felder "Host" und "Port" ein Proxy-Server definiert werden, der zur Kommunikation zwischen JavaClient und FirstSpirit-Server verwendet werden soll.



Kompression: Kompression für die Kommunikation zwischen FirstSpirit-Clients und -Server für den aktuell eingeloggtten Benutzer:

- **None:** keine Kompression bei der Übertragung von Daten zwischen Client und Server.
- **Deflate:** Verwendung des Deflate-Algorithmus mit Standard-Kompression bei der Übertragung von Daten zwischen Client und Server.
- **Deflate_Speed:** Verwendung des Deflate-Algorithmus mit schnellster Kompression bei der Übertragung von Daten zwischen Client und Server.
- **Deflate_Best:** Verwendung des Deflate-Algorithmus mit bester Kompression bei der Übertragung von Daten zwischen Client und Server.
- **Snappy:** Standardmäßige Voreinstellung für neu installierte FirstSpirit-Server. Der Kompressionsmodus "Snappy" steht für die Betriebssysteme (32 und 64 Bit) Microsoft Windows, Linux und Mac OS zur Verfügung. Als Fallback (z. B. auf nicht unterstützten Betriebssystemen) wird der Kompressionsmodus "Deflate Speed" verwendet.

Verschlüsselung: Verschlüsselung für die Kommunikation zwischen FirstSpirit-Clients und -Server für den aktuell eingeloggtten Benutzer:

- **None:** keine Verschlüsselung bei der Übertragung von Daten zwischen Client und Server.
- **TLS:** Verwendung des TLS-Protokolls für die Übertragung von Daten zwischen Client und Server.
- **DH_Arc4:** Verwendung des Verschlüsselungs-Algorithmus DH ARC4 für die Übertragung von Daten zwischen Client und Server

Eine Beschreibung der Einstellungen befindet sich in Kapitel 6.3.4.1 Seite 192.



6.5.2 HTTP (Internet)-Modus

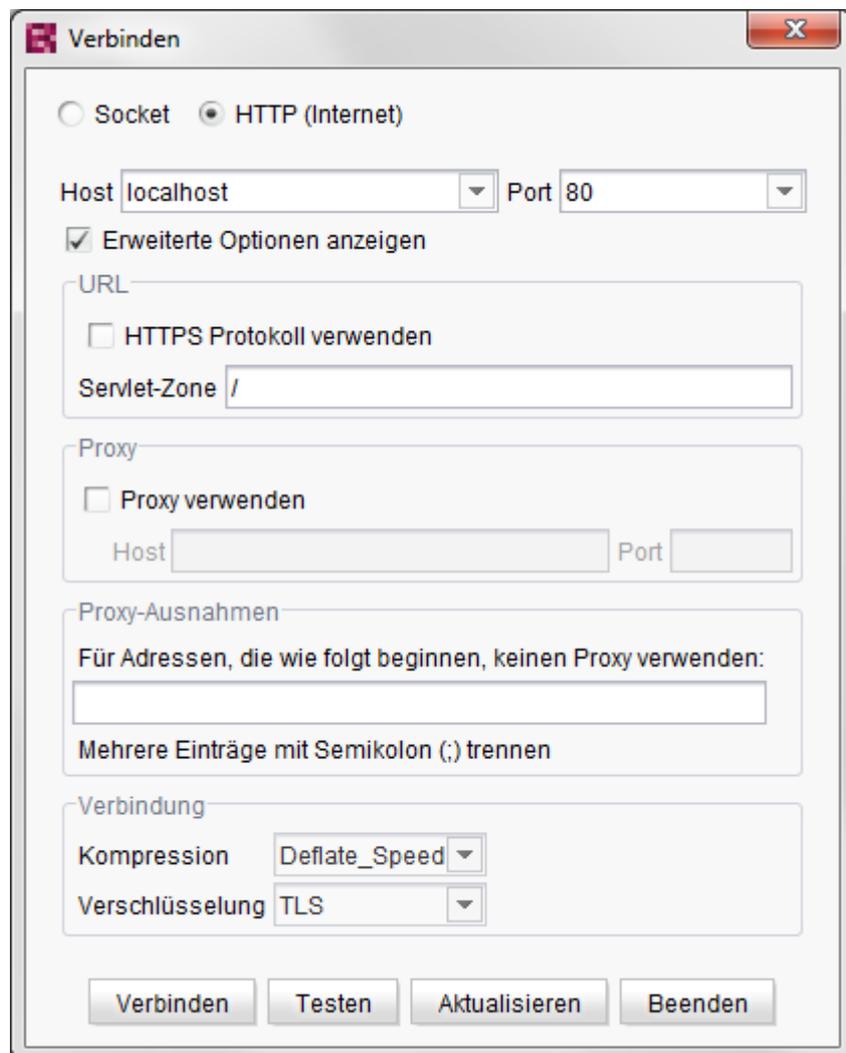


Abbildung 6-8: Verbindungsdialog für den JavaClient – HTTP-Modus

Für den HTTP-Modus kann ein anderer Hostname und eine andere Portnummer als für den Socket-Modus angegeben werden. Folgende Parameter können darüber hinaus konfiguriert werden:

HTTPS Protokoll verwenden: Ist diese Checkbox aktiviert, wird die Kommunikation zwischen Client und Server über das HTTPS-Protokoll verschlüsselt.



Servlet-Zone: Pfadangabe zum Servlet-Verzeichnis. Der Pfad muss immer mit einem "/" beginnen.

Für Adressen, die wie folgt beginnen, keinen Proxy verwenden: Angabe von Domains, die direkt, also nicht über eine Proxy-Verbindung, aufgerufen werden sollen, z. B. Adressen des eigenen Firmennetzwerks. Mehrere Adressen müssen mit Semikolon getrennt werden.

In beiden Fällen (Socket und HTTP) kann über die Schaltfläche "Testen" die Verbindung zwischen JavaClient und FirstSpirit-Server mit den vorgenommenen Einstellungen getestet werden. Ist eine Verbindung nicht möglich, muss die Konfiguration verändert werden. War der Test erfolgreich, kann die Verbindung schließlich über die Schaltfläche "Verbinden" hergestellt werden.

Eine Beschreibung der Einstellungen befindet sich in Kapitel 6.3.4.1 Seite 192.



7 FirstSpirit Server- und Projektkonfiguration

Die FirstSpirit Server- und Projektkonfiguration ist eine Java-Anwendung mit einer komfortablen Swing-Benutzungs Oberfläche, die den FirstSpirit-Administrator bei allgemeinen, administrativen Aufgaben rund um FirstSpirit unterstützt. Über die Oberfläche können neue FirstSpirit-Projekte angelegt und konfiguriert werden. Neben den allgemeinen Aufgaben stehen dort weiterreichende Funktionen zur Verfügung. Beispielsweise kann die Server- und Projektkonfiguration zur Benutzerdefinition bzw. zur Anbindung bereits vorhandener Identity-Management-Systeme, wie z. B. LDAP oder Active-Directory, verwendet werden. Analog zum JavaClient wird die Server- und Projektkonfiguration über Java-Web-Start gestartet und aktualisiert.

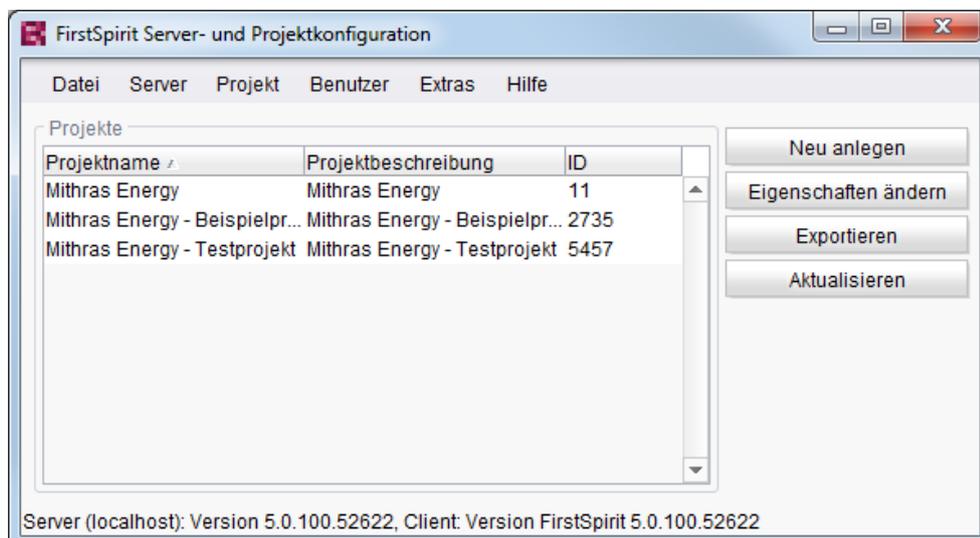


Abbildung 7-1: Server- und Projektkonfiguration



7.1 Server- und Projektadministratoren

Die Aufgabenteilung in FirstSpirit betrifft auch die Administratoren, die gemäß ihrer Rechte unterschieden werden. Jedes Projekt, das auf dem Server angelegt wird, benötigt einen oder mehrere Projektadministratoren.

Projektadministratoren können:

- die Eigenschaften ihres Projekts verändern
- ihr Projekt exportieren
- ihr Projekt aufräumen

Der **Serveradministrator** dagegen kann:

- neue Projekte anlegen / importieren / exportieren / löschen
- Benutzer anlegen
- die Einstellungen aller Projekte verändern
- Projektadministratoren definieren
- Editor- und Funktions-Komponenten installieren und deinstallieren
- spezielle Operationen des Serverbetriebs ausführen

Im **JavaClient** haben Serveradministratoren in neu angelegten Projekten standardmäßig alle Rechte, Projektadministratoren haben ab FirstSpirit Version 5.0R2 ebenfalls standardmäßig alle Rechte, zuvor hatten sie nur eingeschränkte Rechte.

7.2 Elemente der Menüleiste

Im Folgenden werden die einzelnen Einträge in der Menüleiste der Server- und Projektkonfiguration beschrieben.

7.2.1 Datei

7.2.1.1 Beenden

Diese Funktion schließt die Server- und Projektkonfiguration.



7.2.2 Server

7.2.2.1 Aufräumen

Mit dieser Funktion können u.a. Backup- und Log-Dateien des Servers bereinigt werden. Über den Dialog "Server aufräumen" können die folgenden Aufräumaktionen ausgeführt werden:

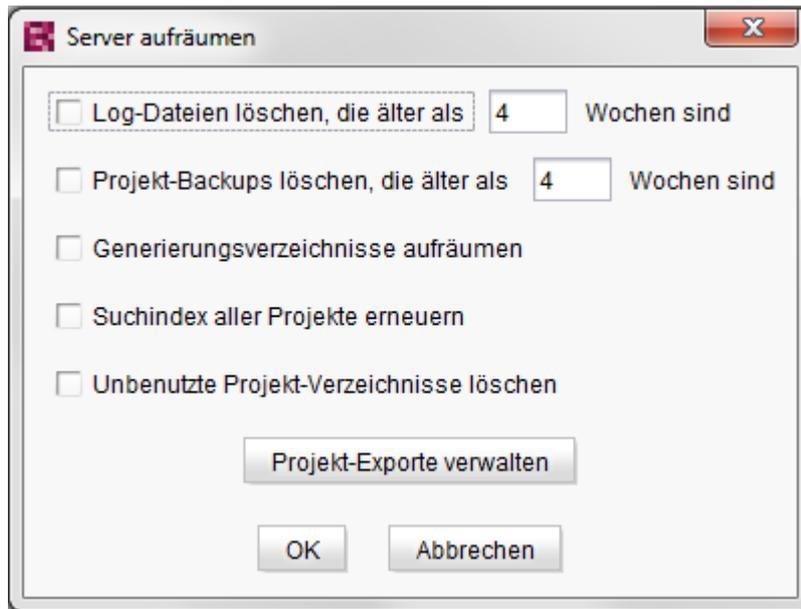


Abbildung 7-2: Server aufräumen

Log-Dateien löschen: Entfernt alle Dateien aus dem Log-Verzeichnis des Servers, die älter als die angegebene Anzahl von Wochen sind.

Projekt-Backups löschen: Entfernt alle (automatisch angelegten) Backups von Projekten, die älter als die angegebene Anzahl von Wochen sind. Betroffen sind alle komprimierten Exportdateien, die sich im Unterverzeichnis "backup" direkt im Wurzelverzeichnis des FirstSpirit-Servers befinden. Diese werden innerhalb der Auftragsplanung angelegt (siehe Kapitel 7.5.10.3 Seite 391). (Dabei ist es möglich, das Verzeichnis auf eine andere Festplatte auszulagern (Parameter `BACKUP_PATH`, siehe Kapitel 4.3.1.15 Seite 58).)



Automatisch angelegte Backups folgen einem bestimmten Namensschema, z.B. '20130417_103642_projectname.tar.gz' – Backups, die dieses Schema nicht befolgen, werden durch diese Aktion nicht entfernt.





Im Export-Verzeichnis für Projekte werden über die Aktivierung der Funktion "Projekt-Backups löschen" keine Projekte gelöscht.

Enterprise-Backups: Neben den automatisch angelegten Projektsicherungen entfernt die Aktion auch alle Enterprise-Backup-Dateien. Beim „Enterprise Backup“ handelt es sich um eine lizenzabhängige Funktion, siehe dazu FirstSpirit Modul-Dokumentation "EnterpriseBackup".

Für Enterprise-Backups gilt folgende Regel: Für ein vollständige funktionierendes Backup, wird immer zwingend (mindestens) der letzte Snapshot benötigt. Aufbauend auf einem Snapshot können zusätzlich differenzielle und inkrementelle Backups erstellt werden. Liegen hier alle Dateien (Snapshots, differentielle und inkrementelle Backups) im aufzuräumenden Zeitraum (also älter als x Wochen), so werden restlos alle Dateien gelöscht. Liegt dagegen mindestens das letzte inkrementelle oder differentielle Backup in dem Zeitraum, der nicht gelöscht werden soll, so bleiben alle differentiellen und inkrementellen Backups einschließlich des dazugehörigen Snapshots erhalten, damit eine Wiederherstellung dieses Backup-Satzes möglich ist. Alle Snapshots, differentielle und inkrementelle Backups, die vor dem letzten Snapshot liegen, werden dagegen gelöscht.

Generierungsverzeichnisse aufräumen: Durch Verwendung dieser Funktionalität werden die nicht mehr benötigten Daten sowohl von gelöschten als auch von deaktivierten Projekten gelöscht. Hierbei handelt es sich um die Verzeichnisse und deren Inhalt, die bei Generierungen und Deployments angelegt wurden:

- Aufräumen der Verzeichnisse "project_projectID" im Unterverzeichnis "web/fs5staging" des FirstSpirit-Servers.
- Aufräumen der Verzeichnisse "project_projectID_partial_deployment" im Unterverzeichnis "web/fs5staging" des FirstSpirit-Servers.
- Aufräumen der Verzeichnisse, in dem die Previews des Projekts gespeichert wurden (Verzeichnis "project_projectID" im Verzeichnis "preview_cache" des Unterverzeichnisses "web/fs5staging"). Das Vorschau-Verzeichnis sowie die zugehörigen Pfade sind über die Konfigurationsdatei fs-server.conf konfigurierbar (siehe Kapitel 4.3.1.8 Seite 47).

Suchindex aller Projekte erneuern: FirstSpirit ermöglicht eine Suche (Volltextsuche über den Lucene-Suchindex) über die Auswahldialoge des JavaClients. Dazu muss der Suchindex für das Projekt berechnet werden. Wird diese Option aktiviert, wird die Indizierung für die Suche für alle Projekte auf dem



Server ausgeführt. Dabei wird nur der aktuelle Stand der Elemente im Projekt berücksichtigt. Historische Daten (z. B. Änderungen an einem Objekt innerhalb eines bestimmten Zeitraums) werden nicht berechnet. Die Neuberechnung des Suchindex kann abhängig von der Anzahl und Größe der Projekte sehr lange dauern. Eine Indizierung für einzelne Projekte oder Teilbereiche eines Projektes können über die FirstSpirit-Access-API ausgeführt werden (siehe Kapitel 7.5.10.5 Seite 392).

Für weitere Informationen siehe Kapitel 9.18 Seite 500.

Unbenutzte Projekt-Verzeichnisse löschen: In manchen Fällen können Projekte nicht vollständig gelöscht werden, z. B. wenn noch Zugriff durch das Betriebssystem auf Dateien in den Projekt-Verzeichnissen besteht. Auf diese Projekte kann über die Anwendung zur Server- und Projektkonfiguration nicht mehr zugegriffen werden. Die zugehörigen Projekt-Verzeichnisse können mit dieser Option endgültig gelöscht werden.

Projekt-Exporte verwalten: Mit einem Klick auf den Button öffnet sich der Dialog "Projekt-Exporte verwalten":

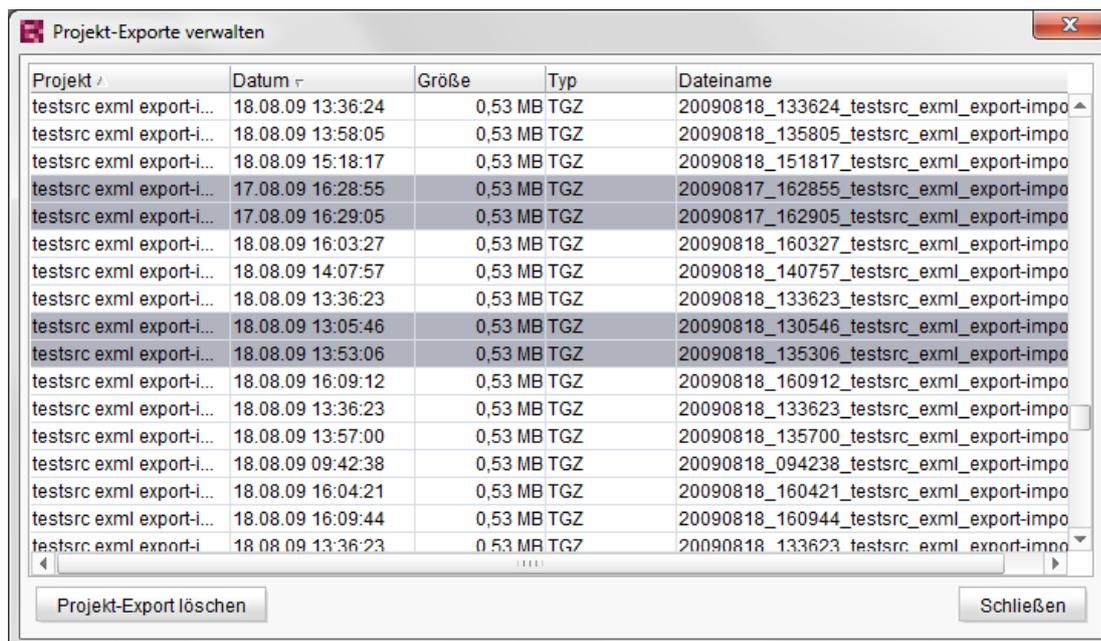


Abbildung 7-3: Projekt-Exporte verwalten

Über den Dialog werden tabellarisch alle vorhandenen Projekt-Exporte aufgelistet. Diese können in der Übersicht markiert und bei Bedarf über den Button "Projekt-Export löschen" gelöscht werden.



7.2.2.2 Beenden

Nach der Bestätigung einer Sicherheitsabfrage, ob der Server wirklich beendet werden soll, wird der Dialog der Server- und Projektkonfiguration über diese Funktion geschlossen und der Server heruntergefahren.



Ein Neustart ist über die Server- und Projektkonfiguration NICHT möglich!

7.2.2.3 Eigenschaften

Durch Aufrufen dieser Funktion öffnet sich ein Fenster, in dem ein ganze Reihe von Servereigenschaften vorgenommen werden können.

Eine genaue Beschreibung dieser Servereigenschaften befinden sich in Kapitel 7.3 ab Seite 231.

7.2.3 Projekt

7.2.3.1 Neu anlegen

Mit dieser Funktion lässt sich ein neues Projekt auf dem Server anlegen.

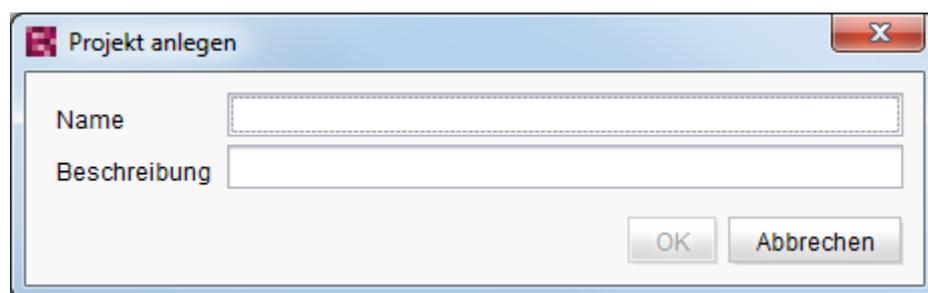


Abbildung 7-4: Projekt – Neu anlegen

Name: Der hier angegebene Name wird in der Wurzel des FirstSpirit-Client Baums aufgeführt. Der vergebene Projektname muss serverweit eindeutig sein.

Beschreibung: Die hier angegebene Beschreibung erscheint in der Projektauswahlliste nach dem Einloggen.

Nach der Eingabe des Projektnamens und der Projektbeschreibung wird im



nächsten Dialog die Projekt-Konfiguration festgelegt.



Abbildung 7-5: Projekt-Konfiguration

Durch das Aktivieren der Checkboxes können die jeweiligen Konfigurationen für das Projekt übernommen werden.

Standard Formatvorlagen: Die mit FirstSpirit ausgelieferten Standard Formatvorlagen werden in den Knoten Formatvorlagen (innerhalb der Vorlagen-Verwaltung des Projekts) importiert. Darüber ist es den Redakteuren möglich, beispielsweise in der Eingabekomponente DOM-Editor Text fett oder kursiv zu stellen.

Standard Arbeitsabläufe: Durch Aktivieren dieser Checkbox werden die mit FirstSpirit ausgelieferten Standard Arbeitsabläufe in den Knoten Arbeitsabläufe (innerhalb der Vorlagen-Verwaltung des Projekts) importiert. In FirstSpirit gibt es zwei integrierte Arbeitsabläufe:

1. "Aufgabe zuweisen": Arbeitsablauf zur allgemeinen Bearbeitung von Aufgaben im Projekt.
2. "Freigabe anfordern": Arbeitsablauf zur Freigabe von Objekten, die im Projekt neu angelegt oder verändert wurden.

Standard Datenbank: Durch Aktivieren dieser Checkbox wird die mit FirstSpirit ausgelieferte Standard Datenbank (Derby) für das Projekt aktiviert. Der Standardlayer kann im FirstSpirit-Client für ein Datenbank-Schema verwendet werden. Damit ist automatisch der schreibende Zugriff auf die Datenbank für dieses Projekt gesetzt (siehe Kapitel 7.4.12 Seite 312).





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

Mit dem Bestätigen der Konfiguration über den Button "OK" wird das neue Projekt sofort in die Projektliste der Server- und Projektkonfiguration aufgenommen und kann über die Schaltfläche "Eigenschaften ändern" (siehe Kapitel 7.3.13 ab Seite 269) ausführlicher bearbeitet werden.

Der Button "Neu anlegen" erfüllt die gleiche Aufgabe.

Dieser Menüpunkt steht nur Serveradministratoren zur Verfügung!

7.2.3.2 Importieren

Mithilfe des Projektimports können exportierte Projekte wieder auf den Server übertragen werden.

Dieser Menüpunkt steht nur Serveradministratoren zur Verfügung!

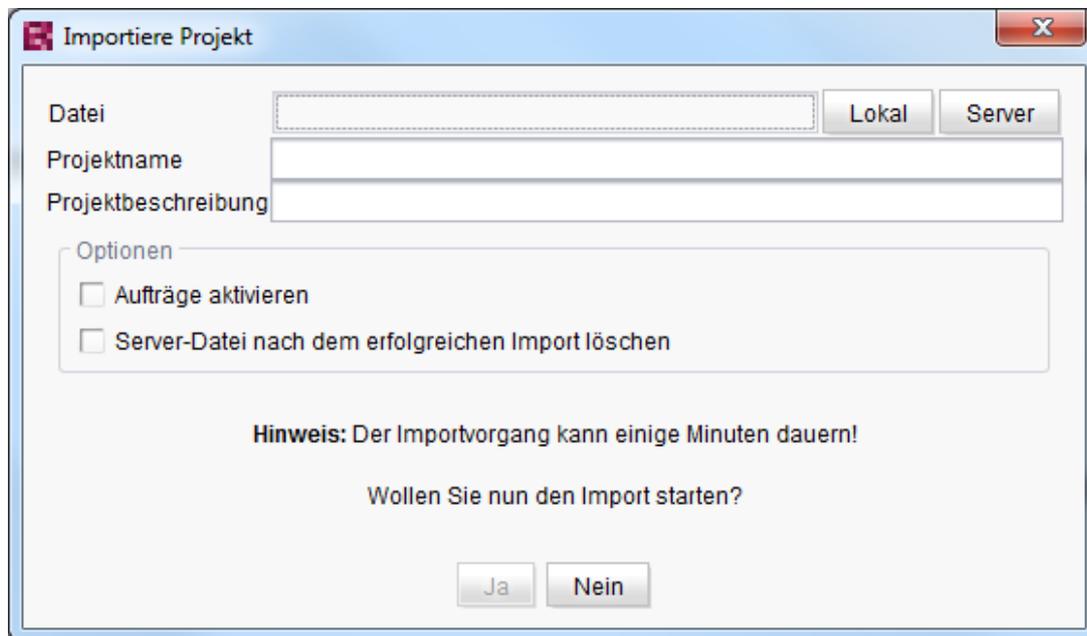


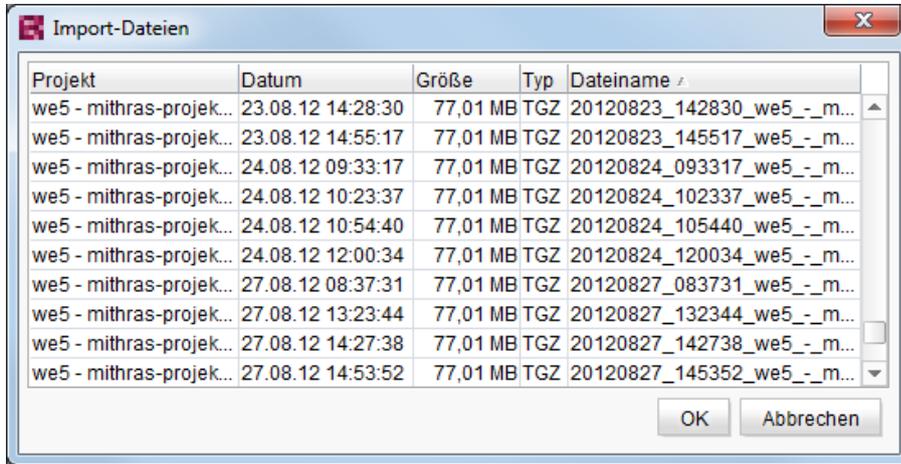
Abbildung 7-6: Projekt – Importieren

Datei: Über die beiden Buttons Lokal und Server kann eine komprimierte Projekt-Export-Datei ausgewählt werden. Über den Button **Lokal** kann das Dateisystem des



Rechners durchsucht werden.

Bei Betätigung des Buttons **Server** wird eine Liste der gepackten Dateien aus dem Exportverzeichnis des Servers angezeigt. Aus dieser kann das gewünschte Projekt ausgewählt werden (siehe Abbildung 7-7):



The screenshot shows a dialog box titled 'Import-Dateien' with a table of project files. The table has five columns: 'Projekt', 'Datum', 'Größe', 'Typ', and 'Dateiname'. There are 10 rows of data, each representing a project file. The 'Projekt' column contains truncated text 'we5 - mithras-projek...'. The 'Datum' column shows dates and times from 23.08.12 to 27.08.12. The 'Größe' column shows '77,01 MB' for all entries. The 'Typ' column shows 'TGZ' for all entries. The 'Dateiname' column shows unique identifiers for each file. At the bottom of the dialog, there are 'OK' and 'Abbrechen' buttons.

Projekt	Datum	Größe	Typ	Dateiname
we5 - mithras-projek...	23.08.12 14:28:30	77,01 MB	TGZ	20120823_142830_we5_-_m...
we5 - mithras-projek...	23.08.12 14:55:17	77,01 MB	TGZ	20120823_145517_we5_-_m...
we5 - mithras-projek...	24.08.12 09:33:17	77,01 MB	TGZ	20120824_093317_we5_-_m...
we5 - mithras-projek...	24.08.12 10:23:37	77,01 MB	TGZ	20120824_102337_we5_-_m...
we5 - mithras-projek...	24.08.12 10:54:40	77,01 MB	TGZ	20120824_105440_we5_-_m...
we5 - mithras-projek...	24.08.12 12:00:34	77,01 MB	TGZ	20120824_120034_we5_-_m...
we5 - mithras-projek...	27.08.12 08:37:31	77,01 MB	TGZ	20120827_083731_we5_-_m...
we5 - mithras-projek...	27.08.12 13:23:44	77,01 MB	TGZ	20120827_132344_we5_-_m...
we5 - mithras-projek...	27.08.12 14:27:38	77,01 MB	TGZ	20120827_142738_we5_-_m...
we5 - mithras-projek...	27.08.12 14:53:52	77,01 MB	TGZ	20120827_145352_we5_-_m...

Abbildung 7-7: Projekt Importieren – Liste der Import-Dateien

Projektname: In diesem Feld muss ein eindeutiger Name für das zu importierende Projekt eingetragen werden. Dieser Name wird auf der Projektwurzel des JavaClients angezeigt.

Projektbeschreibung: In diesem Feld kann eine Projektbeschreibung für das zu importierende Projekt eingetragen werden.

Optionen

Aufträge aktivieren

Ist diese Checkbox **aktiviert**, verbleiben alle Aufträge in dem Status, in dem sie zum Zeitpunkt des Projektexports waren. Das heißt, aktive Aufträge bleiben auch weiterhin aktiv und werden zum eingestellten Zeitpunkt ausgeführt. Diese Option sollte daher nur in Kenntnis der Einträge und mit Bedacht aktiviert werden, da es ansonsten beispielsweise zu unerwünschten Veröffentlichungen kommen kann. Ist die Checkbox **deaktiviert**, bleiben die Aufträge zu einem Projekt zwar erhalten, werden aber deaktiviert.



Server-Datei nach dem erfolgreichen Import löschen

Ist diese Checkbox **aktiviert**, wird die Exportdatei des Projekts, nach dem Importieren des Projekts, auf dem Server gelöscht. Sie steht dann in der Liste der Import-Dateien nicht mehr zur Auswahl zur Verfügung (siehe Abbildung 7-7).

Mit einem Klick auf "Ja" wird der Import gestartet.

Wurde das zu importierende Projekt auf einem FirstSpirit-Server einer höheren Version erstellt, wird nach dem Analysieren der Datei folgende Warnung angezeigt:

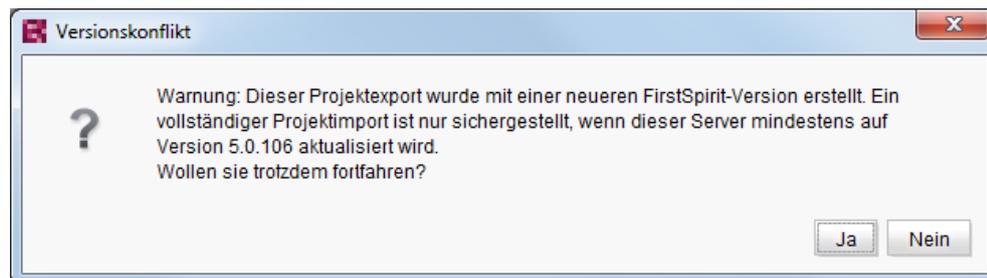
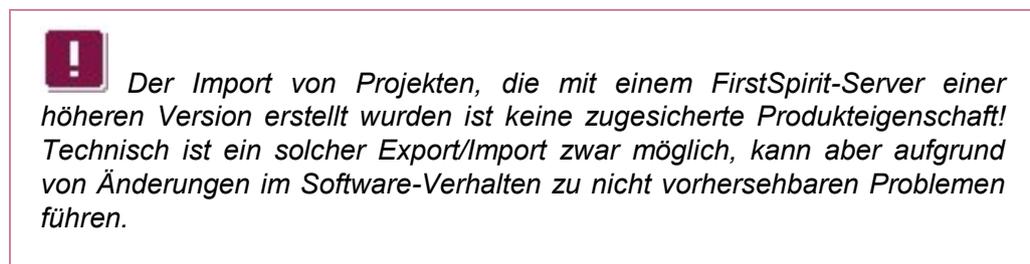


Abbildung 7-8: Versionskonflikt beim Import

Mit "Nein" wird der Import-Vorgang abgebrochen, mit "Ja" wird der Import weiter ausgeführt.



Werden innerhalb des Projekts Datenbank-Inhalte verwendet, so müssen diese Inhalte vor dem Projektimport auf einen neuen Datenbank-Layer gemappt werden. Beim Importieren eines Projekts muss jedem einzelnen Schema des Projekts ein neuer oder ein bereits auf dem Server vorhandener Datenbank-Layer zugewiesen werden (siehe Abbildung 7-9):



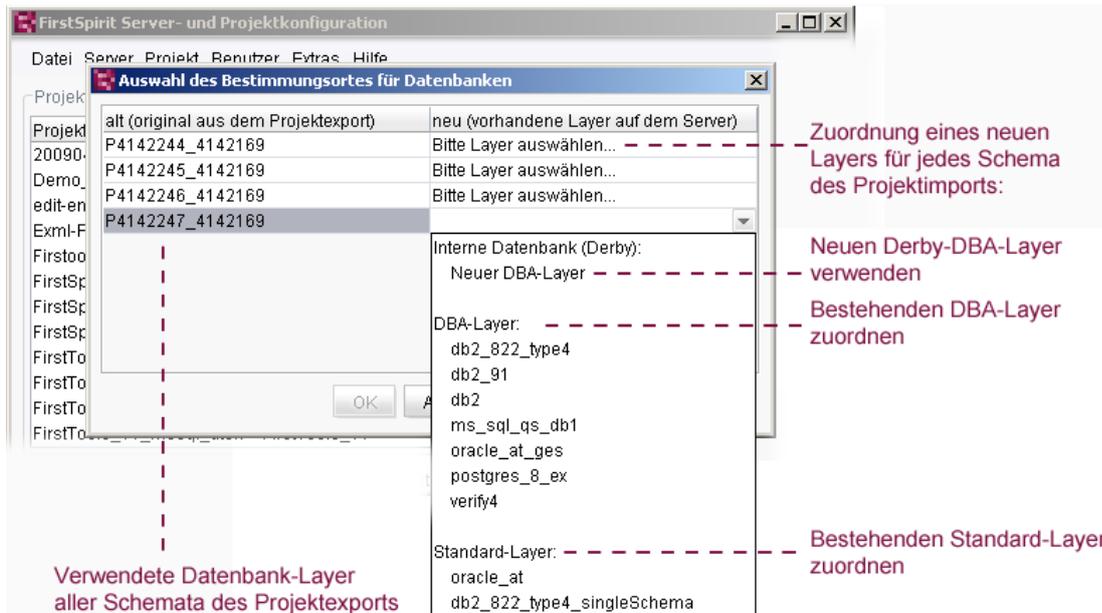


Abbildung 7-9: Auswahl des Bestimmungsortes für Datenbanken

Alle in der Exportdatei des Projekts vorhandenen Schemata werden im Dialog "Auswahl des Bestimmungsortes für Datenbanken" aufgelistet (linker Bereich). Für die neue Zuordnung eines Layers ist im rechten Bereich des Dialogs jeweils eine Auswahl zwischen den folgenden Möglichkeiten zu treffen:

- ein neuer Derby-DBA-Layer
- ein Standard-Layer
- ein DBA-Layer



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

Die Auswahl des Layer-Typen entscheidet darüber, ob FirstSpirit-Benutzer nach dem Import selbstständig neue Schemata im Projekt anlegen können (möglich bei DBA-Layern) oder nicht (wird bei Standard-Layern unterbunden) (weitere Beschreibung siehe Kapitel 4.8.4.2 Seite 159).



Eine nähere Beschreibung der unterschiedlichen Layer-Typen können Sie dem Entwicklerhandbuch entnehmen.

Der Importvorgang kann, abhängig von der Projektgröße, einige Minuten dauern.



Der Gesamtfortschritt sowie die einzelnen Aktionen werden im Importdialog dargestellt.

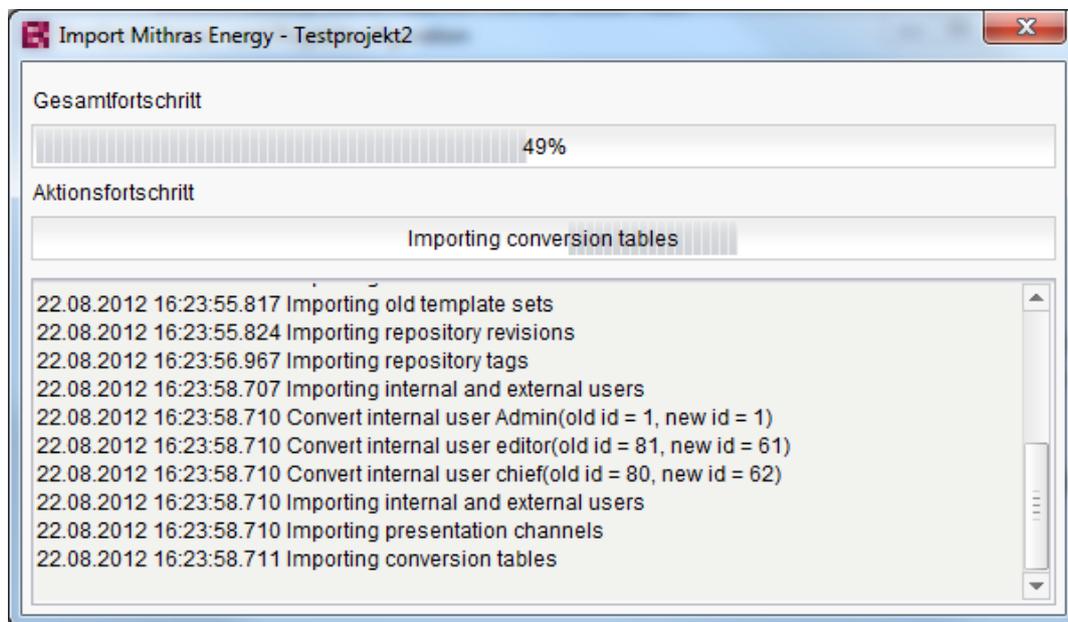


Abbildung 7-10: Projekt Importieren – Fortschrittsanzeige



Treten während eines Projektimports Fehler auf, wird der Import des Projekts nicht unterbrochen, sondern weiter ausgeführt (z. B. bei einem fehlgeschlagenen Datenbank-sync). Am Ende des Importvorgangs wird eine Fehlermeldung angezeigt. Die entsprechenden Exceptions werden im Import-Log angezeigt. Das Projekt ist nach dem Import zunächst deaktiviert.



7.2.3.3 Exportieren

Mithilfe dieser Funktion wird für ein einzelnes Projekt eine Exportdatei erstellt.

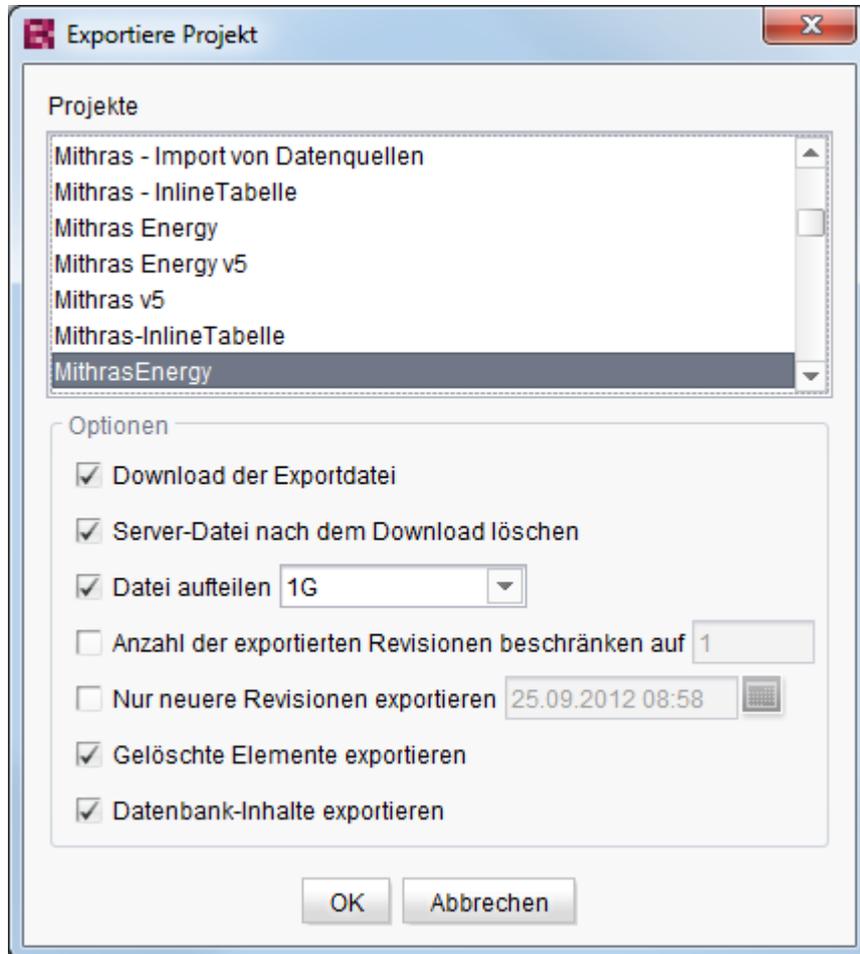


Abbildung 7-11: Projekt – Exportieren

Im oberen Bereich des Fensters sind alle Projekte aufgelistet, die sich aktuell auf dem Server befinden. Aus diesen Projekten kann das zu exportierende Projekt ausgewählt werden.

Download der Exportdatei: Ist diese Option aktiviert, wird die Exportdatei auch im Dateisystem des lokalen Rechners gespeichert.

Nach Bestätigung der Einstellungen durch **OK** öffnet sich ein Fenster, in dem der Zielort für die Exportdatei ausgewählt werden kann.

Server-Datei nach dem Download löschen: Diese Option lässt sich nur aktivieren, wenn ein lokaler Download der Exportdatei erfolgt. Ist diese Option aktiviert, wird die Exportdatei auf dem Server direkt nach dem Download auf den lokalen Rechner



wieder gelöscht.

Datei aufteilen: Mithilfe dieser Option kann die Exportdatei auf eine maximale Größe beschränkt werden. Es werden dann entsprechend viele Exportdateien erstellt. (Diese Option ist z. B. bei besonders großen Projekten erforderlich, um diese auf externe Datenträger kopieren zu können.)

Diese Option hat nur Einfluss auf den Download der Exportdatei. Beim Serverseitigen Export wird ein Projekt ab einer Größe von mehr als 1 GB immer in 1 GB große Dateien aufgeteilt. Es gibt keine Möglichkeit, die Größe der Aufteilung zu beeinflussen!

Anzahl der exportierten Revisionen beschränken auf: Hier kann die Anzahl der beim Exportvorgang angelegten Revisionen festgelegt werden. Bei einer Beschränkung der Revisionen auf 1 wird lediglich der aktuelle Stand des Projektes exportiert (siehe auch Kapitel 9.17 Seite 500).

Nur neuere Revisionen exportieren: Hier kann festgelegt werden, wie alt die zu exportierenden Revisionen maximal sein dürfen. Mit dem Einstellen des aktuellen Datums wird lediglich der aktuelle Stand des Projektes exportiert.

Gelöschte Elemente exportieren: Ist diese Option aktiviert, werden auch gelöschte Elemente in die Exportdatei übernommen. Ist die Option deaktiviert, bleiben alle gelöschten Elemente bei einem Export unberücksichtigt und stehen bei einem erneuten Import unter dem Kontextmenüpunkt "Gelöschte Elemente wiederherstellen" nicht mehr zur Verfügung.

Datenbank-Inhalte exportieren: Mithilfe dieser Option kann der Export von Datenbank-Inhalten gesteuert werden. Ist die Option aktiviert, werden alle Datenquellen mit ihren Inhalten exportiert. Ist die Option deaktiviert, werden nur die Datenquellen ohne ihre Inhalte übernommen. Es stehen dann nur die leeren Tabellen zur Verfügung.



*Das nachträgliche Entfernen des Attributs nullable=1 (von NULLABLE auf NOT NULLABLE) innerhalb der FirstSpirit-Schema-Konfiguration wird nicht in die Datenbank übernommen (auch nicht, wenn die Option Schema-Sync aktiviert ist). Dies kann zu Problemen beim Exportieren und Reimportieren eines Projektes führen. **Ab FirstSpirit 5.0R4** wird dieser Fall beim Projektexport überprüft. Enthält das zu exportierende Projekt bereits gepflegte Datenbank-Inhalte mit Wert NULL, obwohl die entsprechenden Spalte nachträglich als NOT NULLABLE gekennzeichnet wurde, wird der Projektexport (mit einer entsprechenden Fehlermeldung) nicht erfolgreich abgeschlossen.*



7.2.3.4 Deaktivieren

Soll ein Projekt vorübergehend aus der Projektliste entfernt (aber nicht vollständig gelöscht) werden, lässt es sich hier deaktivieren.

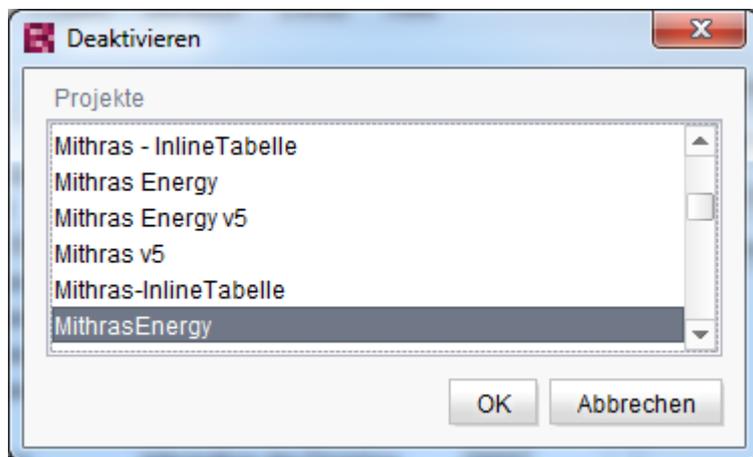


Abbildung 7-12: Projekt – Deaktivieren

Hierzu wird das zu deaktivierende Projekt aus der angezeigten Liste ausgewählt und die Auswahl durch einen Klick auf den "OK" bestätigt. Eine Mehrfachauswahl ist durch gleichzeitiges Drücken der SHIFT- bzw. der STRG-Taste möglich. Existieren noch geöffnete Sitzungen auf ein Projekt, wird vor dem Deaktivieren eine Warnung angezeigt. Das Projekt kann dennoch deaktiviert werden.

Das Deaktivieren eines Projektes ist außerdem zwingende Voraussetzung, um ein Projekt löschen zu können!

Dieser Menüpunkt steht nur Serveradministratoren zur Verfügung!

7.2.3.5 Reaktivieren

Hiermit lassen sich deaktivierte Projekte wieder zur Bearbeitung freigeben. Ein Auswahlfenster zeigt die aktuell deaktivierten Projekte.

Dieser Menüpunkt steht nur Serveradministratoren zur Verfügung!



7.2.3.6 Löschen

Mit dieser Funktion lassen sich Projekte endgültig vom Server löschen. Aus Sicherheitsgründen können allerdings nur Projekte gelöscht werden, die zuvor deaktiviert wurden. Eine Auswahlliste zeigt diese Projekte an.

In der Auswahlliste können auch mehrere Projekte gleichzeitig zum Löschen ausgewählt werden.

Vor dem Löschen eines Projekts erfolgt eine Sicherheitsabfrage.



Das Löschen eines Projektes bedeutet die vollständige Entfernung aller Projektdaten von der Festplatte des Servers. Dieser Vorgang kann nicht rückgängig gemacht werden.

Dieser Menüpunkt steht nur Serveradministratoren zur Verfügung!

7.2.3.7 Eigenschaften

Durch Aufrufen dieser Funktion erscheint ein Fenster, in dem als erstes ein Projekt ausgewählt werden muss.

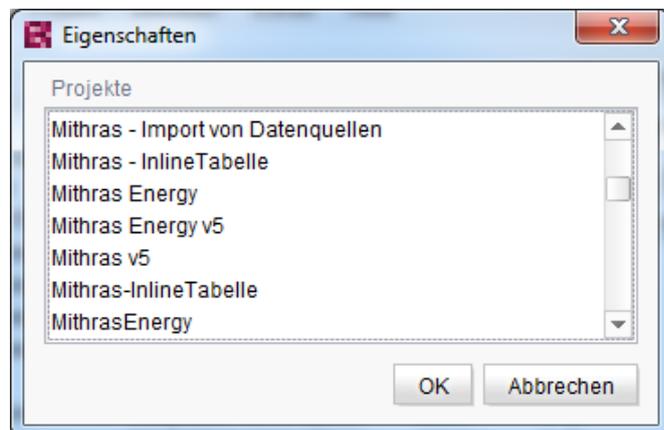


Abbildung 7-13: Projekt – Eigenschaften

Nach Auswahl des Projektes erscheint der Projekteinstellungen-Dialog. Näheres dazu siehe in den Projekteinstellungen (Kapitel 7.3.13 ab Seite 269).



7.2.3.8 Archiv

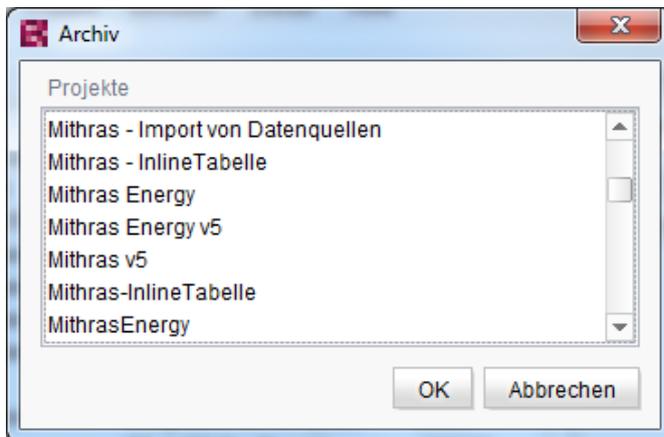


Abbildung 7-14: Projekt – Archiv

Über die Funktion "Archiv" können Daten aus Archivdateien, die während eines Archivierungsauftrags angelegt wurden (Kapitel 7.5.10.1 Seite 380), angezeigt werden:

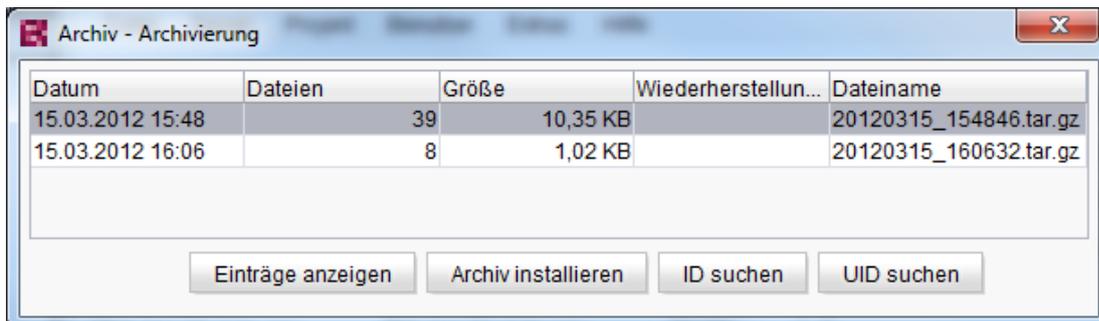


Abbildung 7-15: Archivliste

Nach der Auswahl eines Projekts aus Abbildung 7-14 werden alle vorhandenen Archivdateien mit Erstellungsdatum, der Anzahl der Dateien (archivierte Elemente), der Dateigröße des Archivs und dem Dateinamen angezeigt. Gelöschte Archivdateien werden grau dargestellt.

Einträge anzeigen: Über diesen Button können alle Einträge einer Archivdatei angezeigt werden. Sie werden nach Baumelementen, Datenbankeinträgen und Revisions-Informationen getrennt auf drei Registern angezeigt:

- Baumelemente (siehe Kapitel 7.2.3.8.1 Seite 221)
- Datenbankeinträge (siehe Kapitel 7.2.3.8.2 Seite 222)
- Revisionen (siehe Kapitel 7.2.3.8.3 Seite 223)



7.2.3.8.1 Archivierte Bauelemente

Pfad	ID	UID	Tag	Revision	Archiv	Typ
/formattemplates	416157	common_format_...	FORMATTEM...		2 20120315_154846.tar.gz	OLD_VERSION
/formattemplates/common_for...	416158	b	FORMATTEM...		2 20120315_154846.tar.gz	OLD_VERSION
/formattemplates/common_for...	416159	br	FORMATTEM...		2 20120315_154846.tar.gz	OLD_VERSION
/formattemplates/common_for...	416160	CMS_LINK	FORMATTEM...		2 20120315_154846.tar.gz	OLD_VERSION
/formattemplates/common_for...	416161	deleted	FORMATTEM...		2 20120315_154846.tar.gz	OLD_VERSION
/formattemplates/common_for...	416162	deleted_block	FORMATTEM...		2 20120315_154846.tar.gz	OLD_VERSION
/formattemplates/common_for...	416163	i	FORMATTEM...		2 20120315_154846.tar.gz	OLD_VERSION
/formattemplates/common_for...	416164	inserted	FORMATTEM...		2 20120315_154846.tar.gz	OLD_VERSION
/formattemplates/common_for...	416165	inserted_block	FORMATTEM...		2 20120315_154846.tar.gz	OLD_VERSION
/formattemplates/common_for...	416166	li	FORMATTEM...		2 20120315_154846.tar.gz	OLD_VERSION
/formattemplates/common_for...	416167	p	FORMATTEM...		2 20120315_154846.tar.gz	OLD_VERSION
/formattemplates/common_for...	416168	pre	FORMATTEM...		2 20120315_154846.tar.gz	OLD_VERSION
/formattemplates/common_for...	416169	table	FORMATTEM...		2 20120315_154846.tar.gz	OLD_VERSION
/formattemplates/common_for...	416170	td	FORMATTEM...		2 20120315_154846.tar.gz	OLD_VERSION
/formattemplates/common_for...	416171	tr	FORMATTEM...		2 20120315_154846.tar.gz	OLD_VERSION
/formattemplates/common_for...	416172	u	FORMATTEM...		2 20120315_154846.tar.gz	OLD_VERSION
/formattemplates/common_for...	416173	ul	FORMATTEM...		2 20120315_154846.tar.gz	OLD_VERSION
/formattemplates/common_for...	416174	default_inline_table	TABLEFORM...		2 20120315_154846.tar.gz	OLD_VERSION
/formattemplates/common_for...	416175	default_style_tem...	STYLETEMP...		2 20120315_154846.tar.gz	OLD_VERSION

24 Einträge

Abbildung 7-16: Archivierte Bauelemente

Für jedes archivierte Bauelement aus den FirstSpirit-Verwaltungen werden in dieser Liste folgende Daten angezeigt:

Pfad: Pfad zum Element, zu dem Daten archiviert wurden

ID / UID: ID bzw. Referenzname des archivierten Elements

Tag: Typ des archivierten Elementes (z. B. Medium, Template, Workflow)

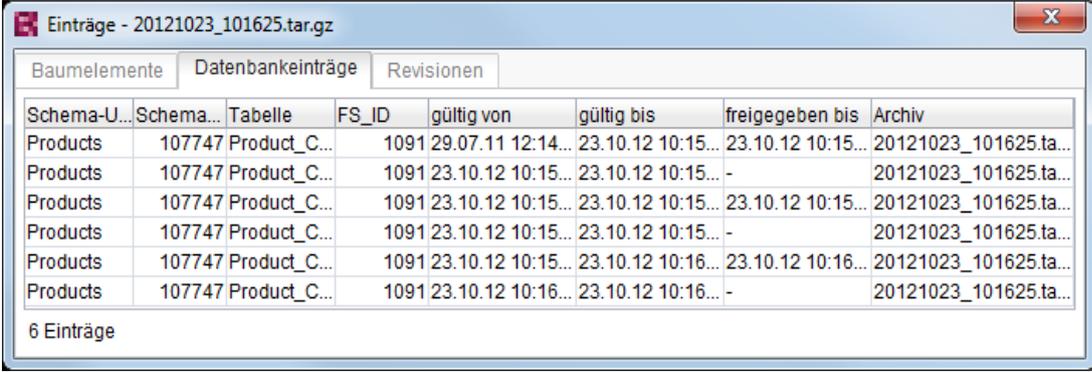
Revision: Revisionsnummer/n des archivierten Elements. Bis zu drei Revisionsnummern werden komplett angezeigt, bei mehr als drei Revisionen wird die erste und letzte Revision angezeigt und in Klammern dahinter die Anzahl der archivierten Revisionen.

Archiv: Name der Datei, in der das Element archiviert wurde.

Typ: Zeigt den Grund für die Archivierung an (OLD_VERSION bedeutet beispielsweise, dass ein älterer und damit nicht mehr benötigter Stand eines Objektes archiviert wurde, DELETED bedeutet, dass ein gelöscht Objekt archiviert wurde).



7.2.3.8.2 Archivierte Datenbankeinträge



Schema-U...	Schema...	Tabelle	FS_ID	gültig von	gültig bis	freigegeben bis	Archiv
Products	107747	Product_C...	1091	29.07.11 12:14...	23.10.12 10:15...	23.10.12 10:15...	20121023_101625.ta...
Products	107747	Product_C...	1091	23.10.12 10:15...	23.10.12 10:15...	-	20121023_101625.ta...
Products	107747	Product_C...	1091	23.10.12 10:15...	23.10.12 10:15...	23.10.12 10:15...	20121023_101625.ta...
Products	107747	Product_C...	1091	23.10.12 10:15...	23.10.12 10:15...	-	20121023_101625.ta...
Products	107747	Product_C...	1091	23.10.12 10:15...	23.10.12 10:16...	23.10.12 10:16...	20121023_101625.ta...
Products	107747	Product_C...	1091	23.10.12 10:16...	23.10.12 10:16...	-	20121023_101625.ta...

6 Einträge

Abbildung 7-17: Archivierte Informationen zu Datenbankeinträgen

Wurden durch den Archivierungsauftrag Datenbankeinträge archiviert (siehe Kapitel 7.5.10.1 Seite 380), werden sie in dieser Liste mit folgenden Angaben angezeigt:

Schema-ID / -UID: ID bzw. Referenzname des Schemas, aus dem die archivierten Informationen zum jeweiligen Datensatz stammen

Tabelle: Name der Tabelle, aus der die archivierten Informationen zum jeweiligen Datensatz stammen

FS_ID: ID des Datensatzes, zu dem Informationen archiviert wurden. Einträge mit derselben FS_ID deuten darauf hin, dass Änderungen an dem jeweiligen Datensatz vorgenommen wurden.

gültig von / gültig bis: analog zu Revisionen bei Bauelementen geben diese Daten den Zeitraum an, in dem eine unveränderte Version des Datensatzes vorgelegen hat. Wurden Änderungen an einem Datensatz vorgenommen, können die verschiedenen Versionen des Datensatzes anhand des Zeitraumes identifiziert werden.

freigegeben bis: gibt den Zeitpunkt an, bis zu dem der Datensatz freigegeben war

Archiv: Name der Datei, in der die Informationen zum Datenbankeintrag archiviert sind



7.2.3.8.3 Archivierte Revisionen

Revision	Datum	Bearbeiter	Kommentar	Status	Archiv
9	15.03.12 15:00:49	Admin (Admin), id=1	unlock	COMMITTED	20120315_154846.tar.gz
12	15.03.12 15:01:58	Admin (Admin), id=1	picture upload	COMMITTED	20120315_154846.tar.gz

2 Einträge

Abbildung 7-18: Archivierte Revisionen

Wurde im Archivierungsauftrag die Option "Systemdaten" im Bereich Objekte aktiviert (siehe Kapitel 7.5.10.1 Seite 380), werden in dieser Liste die archivierten Revisionen mit folgenden Angaben angezeigt:

Revision: Nummer der archivierten Revision

Datum: Datum der Revision

Bearbeiter: Name des Benutzers, der die Änderung vorgenommen hat

Kommentar: automatisch vergebener Kommentar

Status: gibt den Bearbeitungs-Status der Revision an (z. B. COMMITTED für abgeschlossene Revisionen, UNCOMMITTED für angelegte, aber nicht abgeschlossene Revisionen)

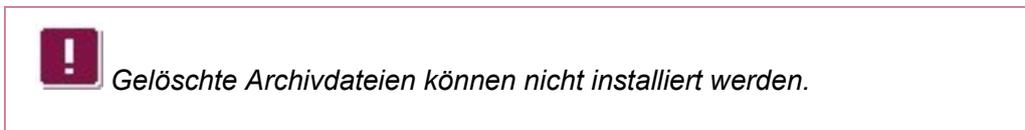
Archiv: Name der Datei, in dem die Revision archiviert wurde

Archiv installieren: Über diesen Button kann das ausgewählte Archiv mit allen Elementen wieder im Projekt installiert werden. Die Archivdatei wird durch eine Installation nicht gelöscht und kann zu einem späteren Zeitpunkt erneut installiert werden.

Sollen mehrere Archivdateien wieder installiert werden, muss dieses in der richtigen Reihenfolge geschehen, da ansonsten Lücken in der Versionshistorie entstehen können. Dazu sollten die Archivdateien in umgekehrter Reihenfolge, nach absteigendem Datum installiert werden, also die zuletzt erstellte Archivdatei zuerst, z. B. (vgl. Abbildung 7-15)



1. Installation der Archivdatei vom 15.03.2012, 16:06
2. Installation der Archivdatei vom 15.03.2012, 15:48



Wird die Reihenfolge nicht eingehalten, kann es zu "IllegalArgumentException"-Fehlermeldungen in der Form "unknown revision id: 18, latest: 46" kommen, wenn die Versionshistorie auf einem betroffenen Objekt aufgerufen wird.

ID suchen: Über diesen Button können die Archive nach IDs der archivierten Elemente durchsucht werden. Bei der Suche werden alle für das Projekt vorhandenen Archive durchsucht.

UID suchen: Über diesen Button können die Archive nach UIDs der archivierten Elemente durchsucht werden.

Bei der Suche werden alle für das Projekt vorhandenen Archive durchsucht. Die Ergebnisse werden in folgendem Fenster aufgelistet:

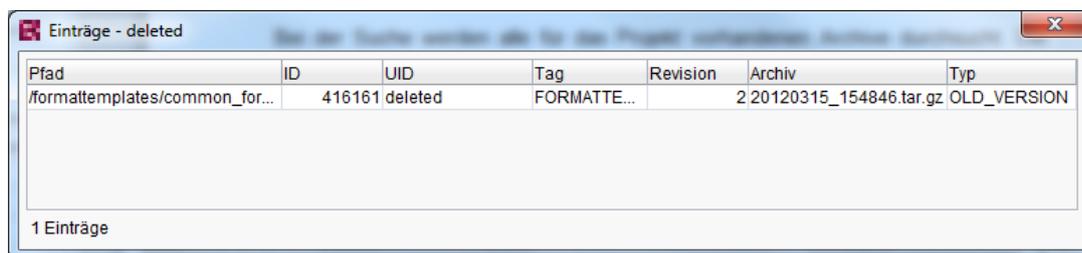


Abbildung 7-19: Ergebnisliste für UID-Suche nach "deleted"



7.2.4 Benutzer

7.2.4.1 Neu anlegen

Mithilfe dieser Funktion kann ein neuer Benutzer auf dem Server angelegt werden. Es erscheint ein Fenster, in dem die Benutzerdaten eingetragen werden können.

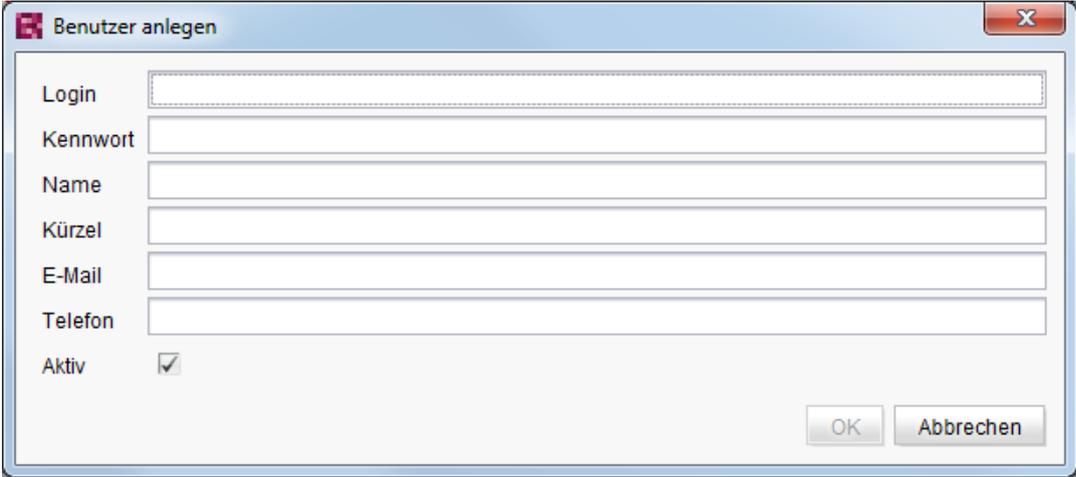


Abbildung 7-20: Benutzer – Neu anlegen

Login: Loginname des neuen Benutzers (Pflichtfeld).

Kennwort: Kennwort des neuen Benutzers (Pflichtfeld).

Die weiteren Informationen für den neuen Benutzer, nämlich der tatsächliche **Name**, ein **Kürzel**, **E-Mail** Adresse und **Telefonnummer** sind optional.

Sonderfall "externe Benutzer": Wird der Benutzer über ein Fremdsystem identifiziert, wird er automatisch beim ersten Anmeldevorgang als FirstSpirit-Benutzer angelegt. Die erforderlichen Benutzerattribute werden dabei aus dem Fremdsystem importiert (das Kennwort wird in FirstSpirit mit einem Zufallswert vorbelegt). Der Benutzer wird anschließend in der Liste der FirstSpirit-Benutzer angezeigt (siehe Abbildung 7-21).

Aktiv: Wird diese Option deaktiviert, wird der neu angelegte Benutzer direkt zwar im System angelegt, kann sich aber nicht authentifizieren (zum "Deaktivieren von Benutzern" siehe Kapitel 7.2.4.2 Seite 226).



7.2.4.2 Bearbeiten

Mithilfe dieser Funktion können die oben genannten Angaben zu einem Benutzer nachträglich geändert werden. Es wird eine sortierte Liste der eingetragenen Benutzer angezeigt. (Die Sortierung kann mit einem Klick auf eine beliebige Spaltenüberschrift neu sortiert werden.)

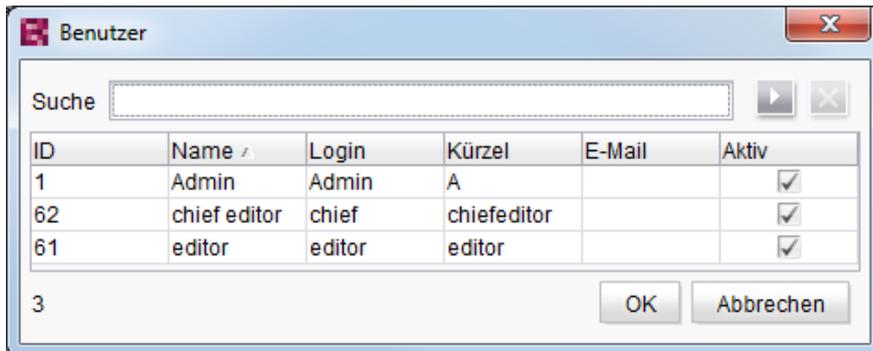


Abbildung 7-21: Benutzer bearbeiten

Suche: Mithilfe der Suchfunktion kann nach Worten oder Wortteilen in den Spalten Name und Login gesucht werden. Durch einen Klick auf den Button  wird die Suche gestartet, mit einem Klick auf den daneben liegenden Button kann die Suche gestoppt werden.

Nachdem ein Benutzer aus der Liste ausgewählt wurde, öffnet sich dann das Bearbeitungsfenster zum Ändern der Benutzerdaten.

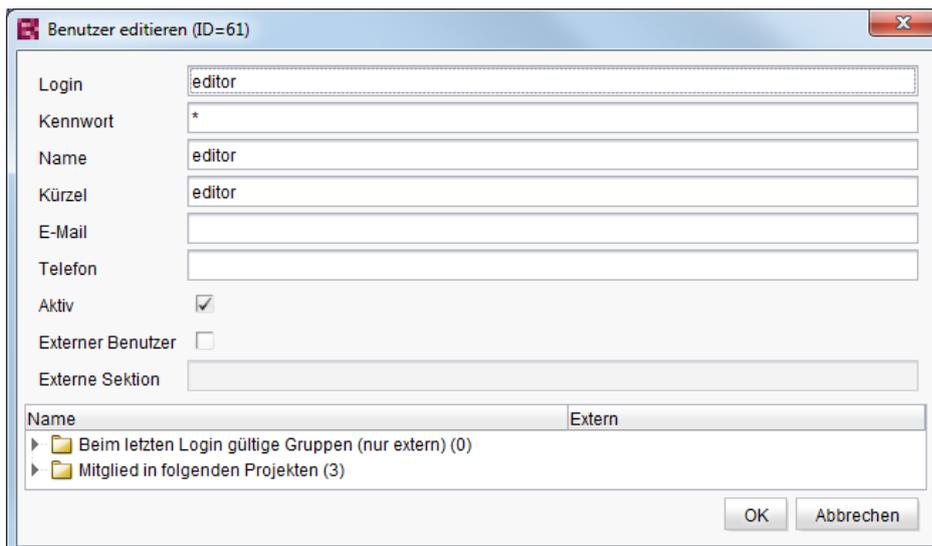


Abbildung 7-22: Benutzer editieren



Handelt es sich beim Benutzer um einen manuell angelegten FirstSpirit-Benutzer, können alle Angaben einschließlich Benutzername und Kennwort geändert werden. Der Benutzer wird intern über eine eindeutige Benutzer-ID identifiziert. Damit ist gewährleistet, dass Informationen zu einem Benutzer erhalten bleiben (beispielsweise die Zuordnung zu einem Projekt), wenn Benutzerattribute (z. B. der Name) geändert werden.

Handelt es sich beim Benutzer um einen automatisch angelegten FirstSpirit-Benutzer, der über ein Fremdsystem identifiziert wird, können die Benutzerattribute über FirstSpirit nicht geändert werden. Externe Benutzer werden durch die aktivierte Checkbox "Externer Benutzer" angezeigt.

Aktiv: Benutzer können deaktiviert werden, ohne sie endgültig aus FirstSpirit zu entfernen. Sie bleiben im System erhalten, können sich jedoch nicht mehr authentifizieren (auch bei Authentifizierung über Fremdsysteme, z. B. SSO). Innerhalb der FirstSpirit-Redaktionsumgebung und im Server-Monitoring werden die deaktivierten Benutzer grau dargestellt. Deaktivierte Benutzer können über diesen Dialog jederzeit wieder aktiviert werden. Alle Informationen zum Benutzer (z. B. die ursprüngliche ID des Benutzers) und alle Projektzuordnungen bleiben erhalten und können direkt wieder verwendet werden (im Gegensatz zum Löschen und erneuten Anlegen eines Benutzers).

Sonderfall "Externer Benutzer": Die Checkbox ist *aktiviert*, wenn es sich beim Benutzer um einen automatisch auf dem Server angelegten Benutzer aus einem Fremdsystem (z. B. aus dem LDAP) handelt (siehe Kapitel 7.4.7.2 Seite 302). Ist die Checkbox *deaktiviert*, wurde der Benutzer manuell angelegt (siehe Abbildung 7-20).

Externe Sektion: Hier wird die LDAP-Sektion angezeigt, auf der der Benutzer registriert ist. Erfolgt der Login über das WindowsLoginModule, wird an dieser Stelle die Domain als externe Sektion angezeigt.

Gruppenzugehörigkeit: Die Gruppenzugehörigkeit eines Benutzers wird im unteren Bereich des Dialogs angezeigt. Die Zuordnung erfolgt unterteilt nach Projekten und kann innerhalb des Dialogs nicht bearbeitet werden. Zum Ändern der Gruppenzugehörigkeit eines Benutzers siehe Kapitel 7.4.8.5 Seite 308. Diese Information ist auch über das FirstSpirit™ Server Monitoring verfügbar (siehe Kapitel 8.5 Seite 449).



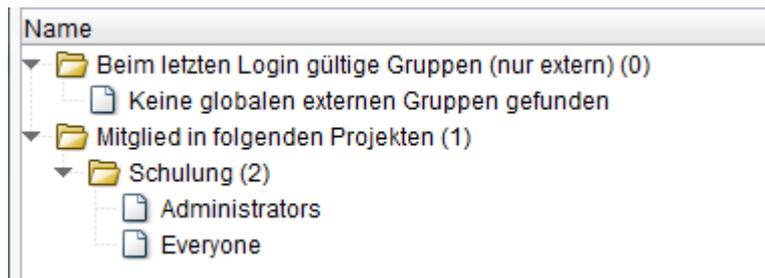


Abbildung 7-23: Gruppenzugehörigkeit eines Benutzers

Die weiteren Informationen für den neuen Benutzer, nämlich der tatsächliche **Name**, ein **Kürzel**, **E-Mail** Adresse und **Telefonnummer** sind optional. Bei externen Benutzern, werden diese Felder eventuell über Benutzerattribute automatisch gefüllt.

Die systeminterne Benutzer-ID wird automatisch vergeben und kann nicht geändert werden. Die Rechteverwaltung für alle Benutzer kann über das Kontextmenü des JavaClient vorgenommen werden.

Dieser Menüpunkt steht nur Serveradministratoren zur Verfügung!

7.2.4.3 Löschen

Mithilfe dieser Funktion kann ein Benutzer aus der Liste entfernt werden.

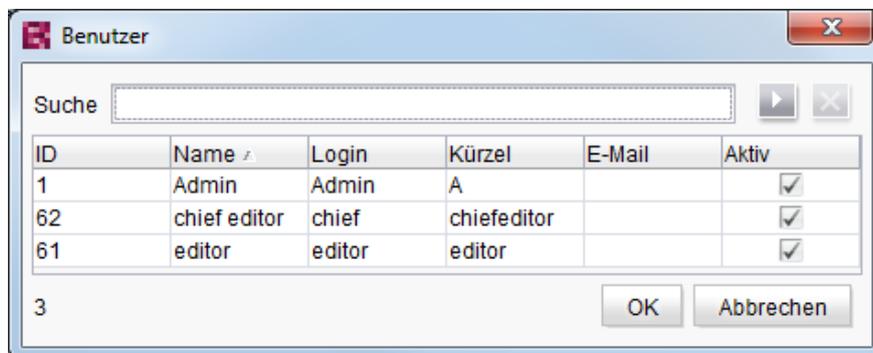


Abbildung 7-24: Benutzer löschen

Suche: Mithilfe der Suchfunktion kann nach Worten oder Wortteilen in den Spalten Name und Login gesucht werden. Durch einen Klick auf den Button  wird die Suche gestartet.

Nachdem ein Benutzer aus der Liste ausgewählt wurde, wird er nach einer Sicherheitsabfrage aus der Liste entfernt. Dabei lässt sich unterscheiden zwischen:

- a. manuell angelegter FirstSpirit-Benutzer:



Handelt es sich um einen manuell angelegten Benutzer, wird der Benutzer durch die Funktion "Benutzer löschen" automatisch aus allen FirstSpirit-Projekten entfernt und vom Server gelöscht.

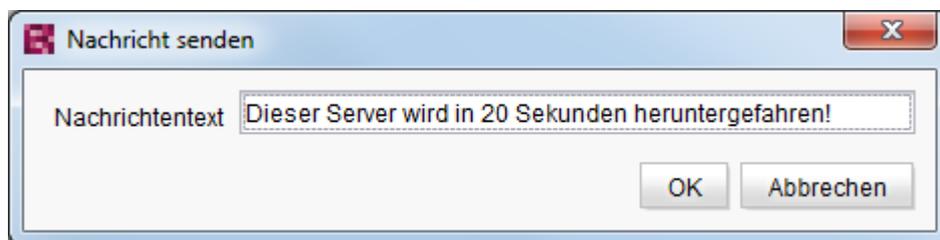
- b. Automatisch angelegter (externer) Benutzer:
Handelt es sich um einen externen Benutzer, wird der Benutzer durch die Funktion "Benutzer löschen" automatisch aus allen FirstSpirit-Projekten entfernt und vom Server gelöscht. Wurde der Benutzer im Fremdsystem nicht gelöscht, wird er bei einer erneuten Anmeldung jedoch wieder als neuer FirstSpirit-Benutzer mit einer neuen Benutzer-ID angelegt.

Dieser Menüpunkt steht nur Serveradministratoren zur Verfügung!

7.2.5 Extras

7.2.5.1 Nachricht senden

Hier kann ein Nachrichtentext editiert werden, der allen aktiven Clients am Server in Form eines PopUp-Fensters übermittelt wird.



Dies ist vor allem dann angebracht, wenn der Server herunter gefahren werden soll. Den Redakteuren bleibt somit Zeit, ihre Arbeit zu speichern, um Datenverluste zu vermeiden.



7.2.6 Hilfe

7.2.6.1 Über FirstSpirit

Mithilfe dieser Funktion ist es möglich, sich einen Überblick über einige Angaben der aktuell verwendeten FirstSpirit Version zu verschaffen:



7.2.6.2 Index

Durch Auswahl dieser Funktion öffnet sich das "Handbuch für Administratoren" (.pdf) im Browser. Dabei wird direkt das Kapitel "Server- und Projektkonfiguration" angezeigt.



7.3 Servereigenschaften

Über den Menüeintrag "Eigenschaften" unterhalb des Menüeintrags "Server" können die Eigenschaften des FirstSpirit-Servers konfiguriert werden.

7.3.1 Globale Server-Eigenschaften

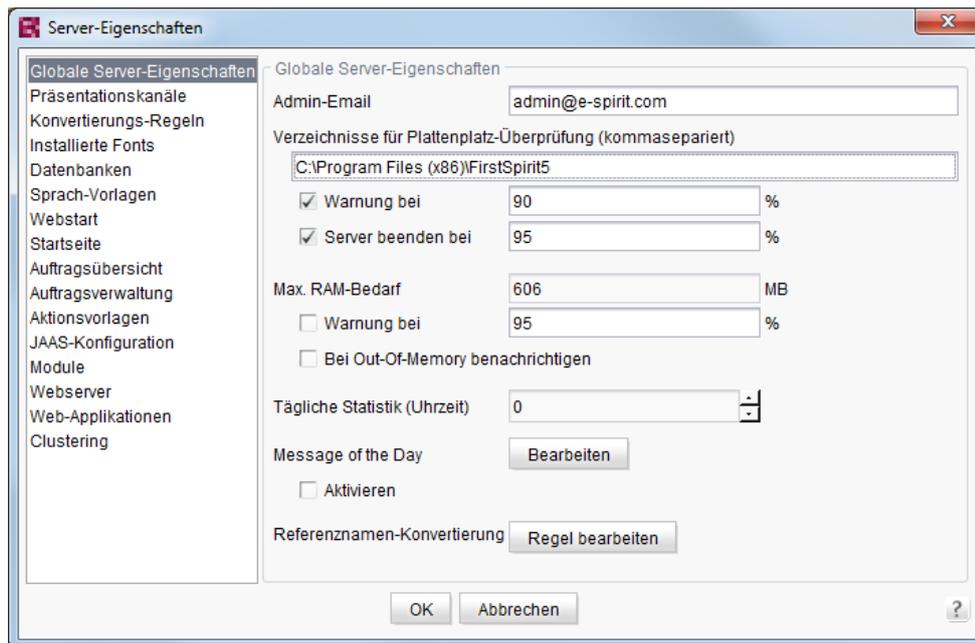


Abbildung 7-25: Server-Eigenschaften – Globale Server-Eigenschaften

Der erste Menüeintrag "Globale Server-Eigenschaften" kennt folgende Konfigurationsoptionen:

Admin-Email: In diesem Feld kann die E-Mail-Adresse des verantwortlichen Serveradministrators eingetragen werden. Kritische Fehler, welche die nachfolgend konfigurierten Felder betreffen, können per E-Mail an die hier angegebene E-Mail-Adresse gesendet werden. Wird keine E-Mail-Adresse angegeben, werden die entsprechenden Fehlermeldungen nur in der Log-Ausgabe des Servers ausgegeben.

Verzeichnisse für Plattenplatz-Überprüfung (kommasepariert): In dieses Feld können Verzeichnisse eingetragen werden, die hinsichtlich des noch zur Verfügung stehenden Festplattenplatzes überwacht werden sollen. Standardmäßig ist hier das Basis-Verzeichnis des FirstSpirit-Servers angegeben (siehe auch `fs-server.conf`, Parameter `hdd.directories`, Kapitel 4.3.1.15 Seite 58). 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. Der freie Speicherplatz in den angegebenen Verzeichnissen wird in einem Intervall von fünf Minuten ermittelt.

Warnung bei / Server beenden bei: In diesem Feld wird die Prozentzahl angegeben, ab der eine Warnungs-E-Mail an den Serveradministrator versendet bzw. der FirstSpirit-Server inklusive einer entsprechenden Benachrichtigung an den Serveradministrator heruntergefahren werden soll.

Betreff der Warnungs-E-Mail:

"FIRSTspirit server 'MYSERVER' disk space warning: /home/fs/firstspirit5, 32,01 GB free, 42,52 GB used".

Betreff der E-Mail beim Herunterfahren des FirstSpirit-Servers:

"FIRSTspirit server 'MYSERVER' disk space shutdown limit reached: /home/fs/firstspirit5, 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 für eine Warnung auf 90% und für das Herunterfahren des FirstSpirit-Servers auf 95% gesetzt. D. h. der Serveradministrator erhält eine Warnungs-E-Mail, wenn 90% des zur Verfügung stehenden Speicherplatzes belegt sind (siehe auch `fs-server.conf`, Parameter `hdd.limit`, Kapitel 4.3.1.15 Seite 58) und der Server wird herunter gefahren, wenn 95% des zur Verfügung stehenden Speicherplatzes belegt sind (siehe auch `fs-server.conf`, Parameter `hdd.shutdown`, Kapitel 4.3.1.15 Seite 58). Sind mehrere Verzeichnisse angegeben, wird eine E-Mail versendet oder der Server heruntergefahren, wenn in **einem** der Verzeichnisse die entsprechende Prozentzahl überschritten ist. Der für eine Warnung gewählte Wert sollte niedriger sein, als der Wert für "Server beenden". Es wird maximal alle 12 Stunden eine Warnungs-E-Mail versendet.

Über die jeweilige Checkbox lässt sich der Versand einer Warnungs-E-Mail (siehe auch `fs-server.conf`, Parameter `hdd.limit.active`, Kapitel 4.3.1.15 Seite 58) bzw. das Herunterfahren des Servers (siehe auch `fs-server.conf`, Parameter `hdd.shutdown.active`, Kapitel 4.3.1.15 Seite 58) aktivieren oder deaktivieren. Standardmäßig sind beide Checkboxen aktiviert.

Max. RAM-Bedarf: In diesem Feld kann der Wert für den maximalen RAM-Bedarf (in MB) für die VM des Server eingestellt werden. Dieser Wert wird beim Starten des Servers übergeben. Zusätzlich kann ein kritischer Schwellenwert (in %) definiert werden. Wird nebenstehende Checkbox aktiviert, wird eine Warnung in das Server-Log geschrieben, sobald der kritische Wert des maximal zur Verfügung stehenden



RAMs überschritten wurde (optional kann eine E-Mail an den Serveradministrator gesendet werden). Wird zusätzlich die Checkbox "Bei Out-Of-Memory benachrichtigen" aktiviert, wird eine E-Mail bei einer auftretenden OutOfMemory-Exception auf dem Server an den Serveradministrator geschickt.

Beispiel:

```
MEMORY:
used memory is 720.47 MB
HEAP STORAGE:
committed = 827719680
init = 838860800
max = 827719680
used = 755463032
NON_HEAP STORAGE
committed = 51019776
init = 8552448
max = 100663296
used = 50923776
```

In diesem Fall wird der Server mit: `-Xmx=900m` (HEAP STORAGE max + NON_HEAP STORAGE max) gestartet und in der Server- und Projektkonfiguration im Feld "Max. RAM-Bedarf" folgende Werte eingestellt:

- 90% (HEAP STORAGE used / HEAP STORAGE max=91% also mehr als 90%)

Tägliche Statistik (Uhrzeit): In diesem Feld kann eine Uhrzeit eingestellt werden, zu der eine tägliche auszuführende Statistik gestartet werden soll.

Message of the Day: Über den Button **Bearbeiten** kann für jede Projektsprache ein Text eingegeben werden, der beim Starten als Meldung erscheinen soll. Im FirstSpirit-JavaClient wird die Meldung beim Öffnen eines Projekts erst angezeigt, wenn die Checkbox "Aktivieren" aktiviert wurde.

Referenznamen-Konvertierung: Über diese Funktion kann serverweit ein Regelsatz definiert werden, nach dem ungültige Zeichen beim Anlegen neuer FirstSpirit-Objekte oder beim Ändern des Referenznamens umgeformt werden sollen. Durch einen Klick auf die Schaltfläche "Regel bearbeiten" öffnet sich folgendes Fenster:



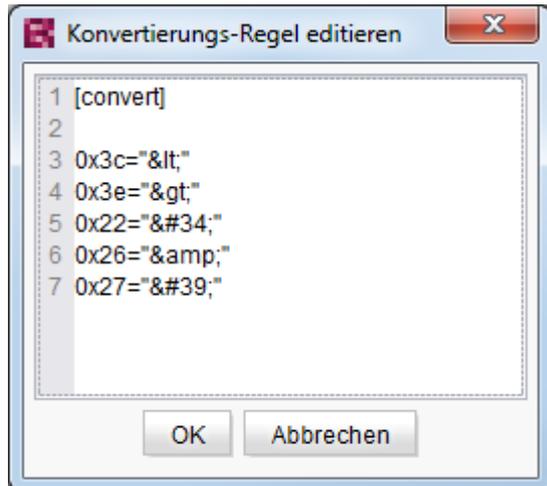


Abbildung 7-26: Referenznamen-Konvertierung

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 (Vokal in Kleinschrift + "e") und "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:...".*





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.



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.

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

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 neu gestartet werden.

7.3.2 Präsentationskanäle

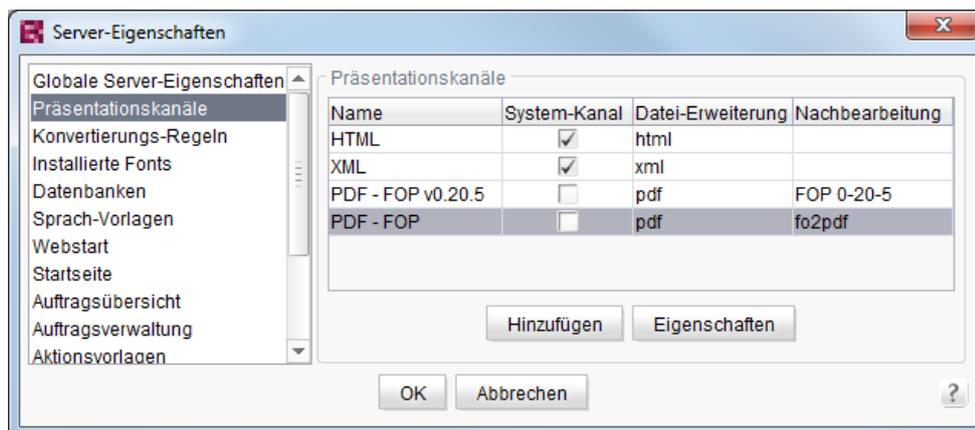


Abbildung 7-27: Server-Eigenschaften – Präsentationskanäle

In diesem Bereich werden die Präsentationskanäle des Servers definiert. Auf den hier definierten Präsentationskanälen beruhen die Vorlagensätze, die auf dem Server zur Verfügung stehen.



Hinzufügen: Beim Klick auf Hinzufügen öffnet sich ein Dialogfenster zur Definition eines neuen Präsentationskanals (siehe Abbildung 7-28).

Eigenschaften: Wird ein Präsentationskanal aus der Liste markiert, wird der Button "Eigenschaften" aktiv. Beim Klick auf den Button öffnet sich ein Dialogfenster zum Bearbeiten der Eigenschaften des Präsentationskanals (siehe Abbildung 7-28).

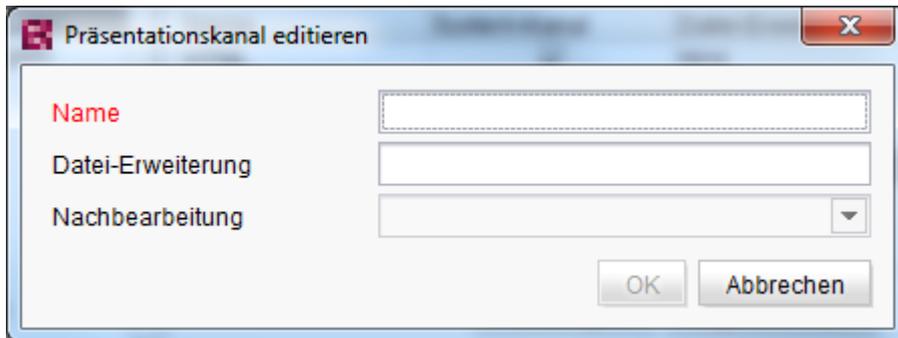


Abbildung 7-28: Server-Eigenschaften – Präsentationskanäle editieren

Name: Name des Präsentationskanals. Unter diesem Namen wird der Präsentationskanal im Projekt und innerhalb der Liste der Präsentationskanäle nach dem Speichern angezeigt.

System-Kanal: Die Checkbox zeigt an, ob es sich beim Präsentationskanal um einen System-Kanal handelt. Diese Präsentationskanäle können nicht bearbeitet oder gelöscht werden.

Datei-Erweiterung: In diesem Feld wird die Datei-Erweiterung des Präsentationskanals angegeben, beispielsweise "html" für den HTML-Präsentationskanal.

Nachbearbeitung: Über diese Auswahl lässt sich der FOP-Prozessor für die nachgelagerte Bearbeitung festlegen.

Zur Unterstützung von Apache FOP siehe Kapitel 7.8 Seite 421.



7.3.3 Konvertierungs-Regeln



Abbildung 7-29: Server-Eigenschaften – Konvertierungs-Regeln

In diesem Bereich werden die Konvertierungsregeln des Servers definiert. Die hier definierten Konvertierungsregeln können im Projekt innerhalb der Formatvorlagen für jeden Präsentationskanal aus einer Klappliste ausgewählt werden. Standardmäßig wird die Konvertierungsregel "Convert HTML" mit folgendem Inhalt mit ausgeliefert:

```
[convert]
0x3c="&lt;";
0x3e="&gt;";
0x22="&#34;";
0x26="&amp;";
0x27="&#39;";
```

Für jeden Ausgabekanal kann genau eine Konvertierungsregel festgelegt werden.



Konvertierungsregeln dienen dazu, eingegebene Zeichen zu konvertieren. Eine Konvertierungsregel besteht aus zwei Teilen:

- "convert" wird immer dann ausgewertet, wenn für eine Formatvorlage im Client eine Konvertierungsregel selektiert wurde.
- soll zusätzlich "quote" ausgewertet werden, muss der entsprechende Schalter in der Formatvorlage aktiviert werden.

Weiterführende Informationen zu Formatvorlagen siehe "FirstSpirit Handbuch für Entwickler".

Hinzufügen: Es öffnet sich ein Auswahldialog, in dem die lokale Dateistruktur des Rechners durchsucht werden kann. Eine neue Konvertierungsregel muss zuvor als Textdatei angelegt worden sein.

Bearbeiten: Über den Dialog "Konvertierungs-Regel editieren" kann eine bestehende Konvertierungsregel bearbeitet werden.

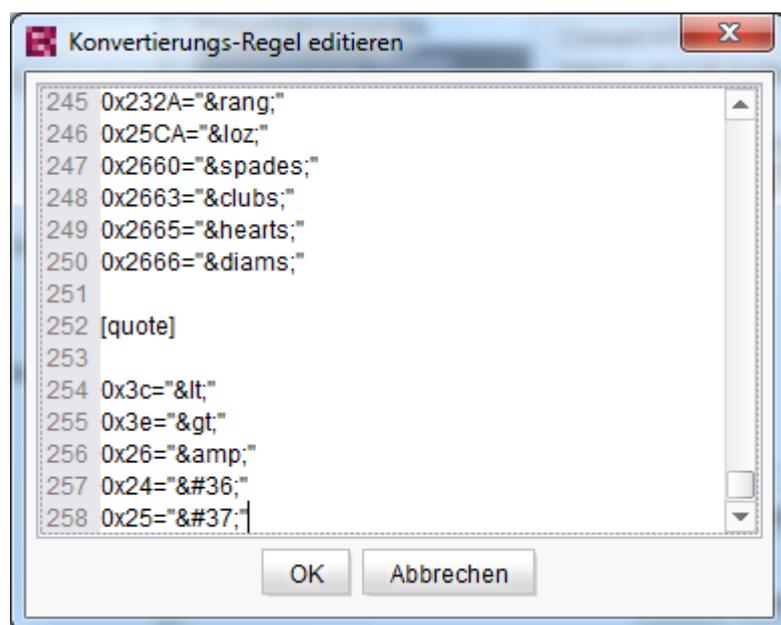


Abbildung 7-30: Server-Eigenschaften – Konvertierungs-Regel editieren

Löschen: Mit einem Klick auf den Button können bestehende Konvertierungsregeln vom Server gelöscht werden. Vor dem Löschen erfolgt eine Sicherheitsabfrage.



7.3.4 Installierte Fonts

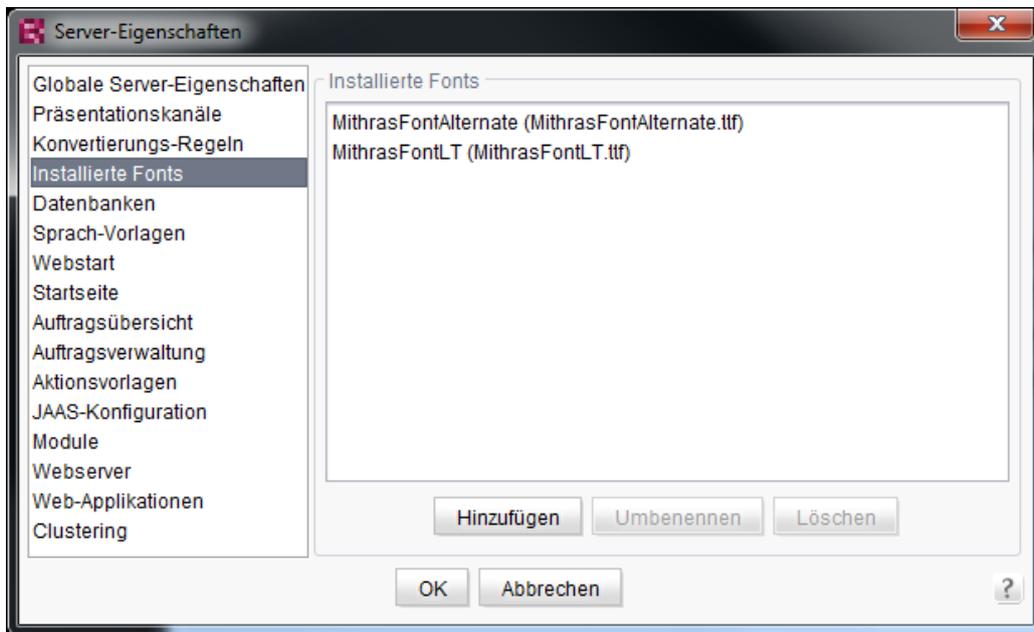


Abbildung 7-31: Server-Eigenschaften – Fonts installieren

Hier werden die auf dem Server installierten True Type Fonts (TTF) aufgelistet. Auf diese Fonts kann innerhalb einer Vorlage zugegriffen werden, beispielsweise um Bilder mit einem Schriftzug in der entsprechenden Schriftart zu versehen:

```
§CMS_REF(cmsFont(font:"MetaMediumCaps",size:12,color:"#666666",justify:"right",valign:"center",xoffset:-4,yoffset:-2,media:"version_from",bounds:"image",text:#content.toString))§
```

Dargestellt wird der Referenzname, über den die Schrift referenziert werden kann und in Klammern der Dateiname, unter dem die Schriftart im Dateisystem des Servers abgelegt wurde (siehe Abbildung 7-31).

Hinzufügen: Soll ein weiterer Font hinzugefügt werden, ist es erforderlich, dass dieser Font bereits auf dem Betriebssystem des Server-Rechners installiert ist. Es reicht nicht aus, den Font auf dem Betriebssystem des Client-Rechners zu installieren. Um einen neuen Font auf dem Server hinzuzufügen, muss zuerst ein eindeutiger Referenzname des neuen Fonts angegeben werden. Mit einem Klick auf "Font upload" kann ein True Type Font aus dem Dateisystem ausgewählt werden.





Abbildung 7-32: Neuen Font anlegen

Wird die Auswahl mit "OK" bestätigt, wird der neue Font angelegt.

Umbenennen: Umbenennen eines in der Liste angezeigten Fonts.

Löschen: Löschen des in der Liste markierten Fonts. Der Font wird aus dem Server-Verzeichnis `data/fonts` entfernt und in der Liste der installierten Fonts nicht mehr angezeigt.



Das Umbenennen oder Löschen eines Fonts kann zu Fehlern in Vorlagen führen, die diesen Font verwenden. Es erscheint daher eine Sicherheitsmeldung.

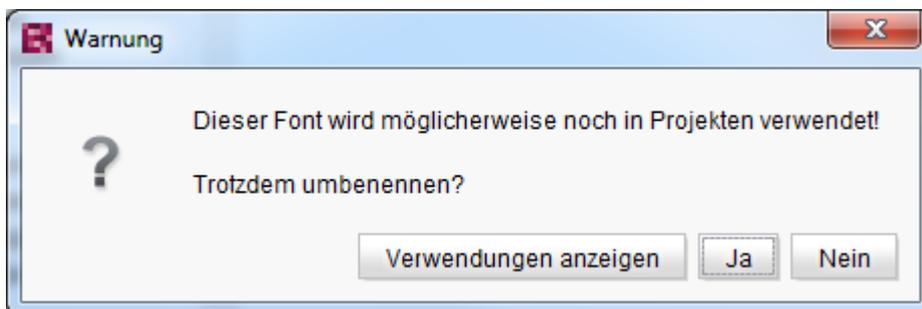


Abbildung 7-33: Warnung vor dem Löschen bzw. Umbenennen von Fonts

Mit einem Klick auf den Button "Ja" wird der Font umbenannt bzw. gelöscht. Dabei kann es zu Fehlern innerhalb der Projekte kommen, die diesen Font verwenden. Im Zweifelsfall können über den Button "Verwendungen anzeigen", diese Projekte ermittelt werden.

Mit einem Klick auf den Button "Verwendungen anzeigen" können die Verwendungen eines Fonts innerhalb der Projekte ermittelt werden. Die Projekte werden in einer Liste angezeigt. Wurden keine Verwendungen gefunden, wird das durch eine Meldung angezeigt.

Mit einem Klick auf den Button "Nein" wird der Dialog geschlossen. Der betroffene



Font wird nicht gelöscht bzw. umbenannt.

7.3.5 Datenbanken

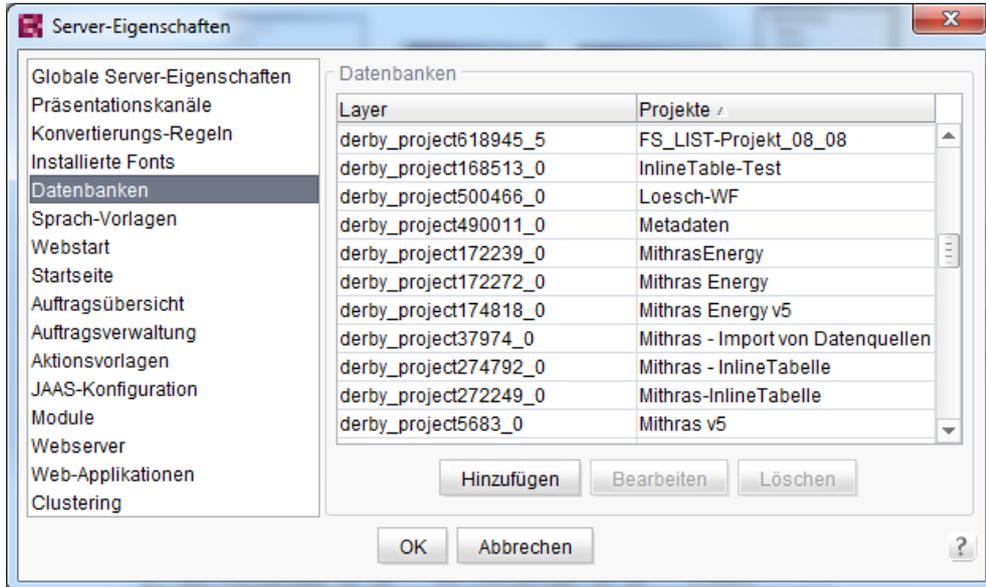


Abbildung 7-34: Server-Eigenschaften – Datenbanken

In diesem Bereich werden alle auf dem Server eingebundenen Datenbank-Layer aufgelistet, das bedeutet, alle auf dem Server zur Verfügung stehenden Verbindungen zu einer Datenbank. Ein hier definierter Datenbank-Layer wird innerhalb der Projekteigenschaften für ein Projekt ausgewählt (vgl. Kapitel 7.4.12 Seite 312) und kann anschließend in der Vorlagen-Verwaltung des Projekts bei der Definition eines Datenbank-Schema ausgewählt werden (weiterführende Informationen zur Verwendung von Datenbank-Layern im Projekt siehe "FirstSpirit Handbuch für Entwickler (Grundlagen)").

Mit einem Klick auf den Button "Hinzufügen" kann ein neuer Datenbank-Layer auf dem Server hinzugefügt werden. Es öffnet sich ein Fenster, in das lediglich der Name der neuen Konfiguration eingetragen werden muss.



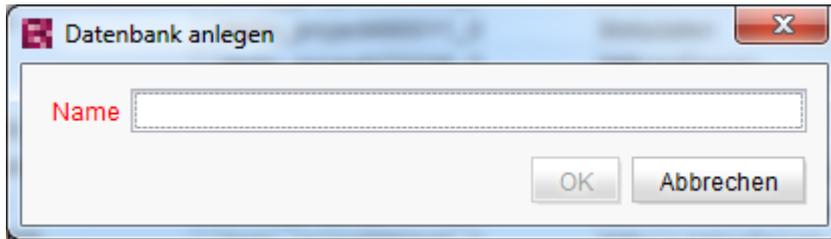


Abbildung 7-35: Anlegen eines neuen Datenbank-Layers

Anschließend erscheint ein Fenster zur Konfiguration der Datenbank, in diesem Fenster ist bereits die Grundstruktur für eine MySQL-Datenbank eingetragen. Die Konfiguration der Datenbank muss nun an die lokalen Anforderungen angepasst werden (siehe Kapitel 7.3.5.1 f.). (Genauere Informationen zur Datenbankanbindung siehe Kapitel 4.3.3 Seite 76 und zur Konfiguration der Datenbank siehe Kapitel 4.8.4 Seite 156.)



FirstSpirit verwendet für den Zugriff auf Datenbanken JDBC²⁸. Für jede Datenbank ist der geeignete Treiber in den Classpath des Servers aufzunehmen bzw. die JDBC-Treiberdateien als FirstSpirit-Modul einzubinden (siehe auch Kapitel 4.8.1 Seite 142).

Eine bestehende Konfiguration lässt sich mit einem Klick auf den Button "Bearbeiten" editieren. Es öffnet sich ein Fenster mit der bestehenden Konfiguration (siehe Abbildung 7-36). Es lässt sich unterscheiden zwischen:

- Konfiguration JDBC-Parameter (siehe Kapitel 7.3.5.1 Seite 243)
- Konfiguration Verbindung (siehe Kapitel 7.3.5.2 Seite 244)

Ein Klick auf den Button "Löschen" entfernt einen bestehenden Datenbank-Layer vom FirstSpirit-Server. Falls der Layer eine Embedded Derby-Datenbank repräsentiert, wird mit einem Klick auf den Button zusätzlich zum Layer auch die zugehörige Datenbank gelöscht. (Bei allen anderen Layern muss die zugehörige Datenbank manuell vom Datenbank-Administrator gelöscht werden.)

²⁸ Java Database Connectivity





Bei Datenbanken, die aktuell in einem Projekt verwendet werden, erscheint vor dem Löschen eine Sicherheitsabfrage.

7.3.5.1 JDBC Parameter konfigurieren

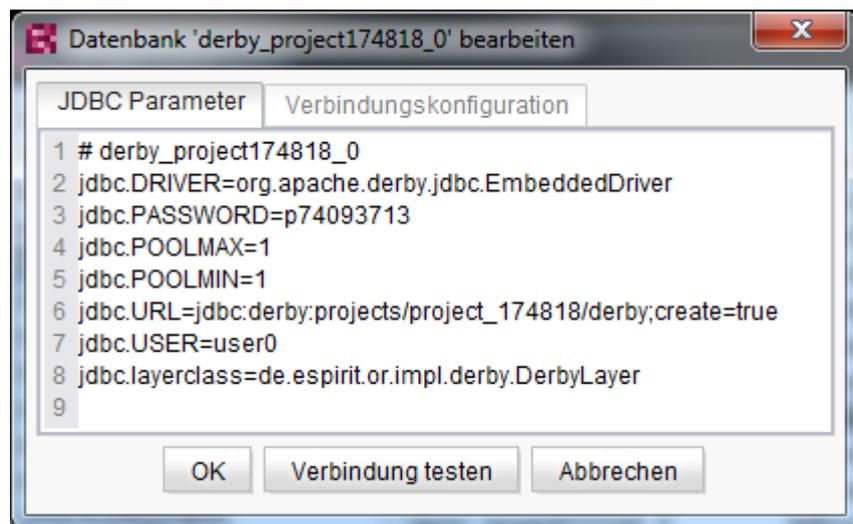


Abbildung 7-36: Datenbankbindung konfigurieren (JDBC)

Konfiguration der JDBC-Parameter für die Datenbankbindung.

Die Konfiguration kann geändert und anschließend mit einem Klick auf "OK" gespeichert werden. Bevor eine Änderung gespeichert wird, sollte immer die Verbindung zur Datenbank getestet werden. Genauere Informationen zur Konfiguration der Datenbank siehe Kapitel 4.8.4 Seite 156.

Verbindung testen: Nachdem die Konfiguration für die Datenbank abgeschlossen ist, kann durch einen Klick auf diesen Button überprüft werden, ob alle Angaben korrekt vorgenommen wurden.



7.3.5.2 Verbindungskonfiguration testen

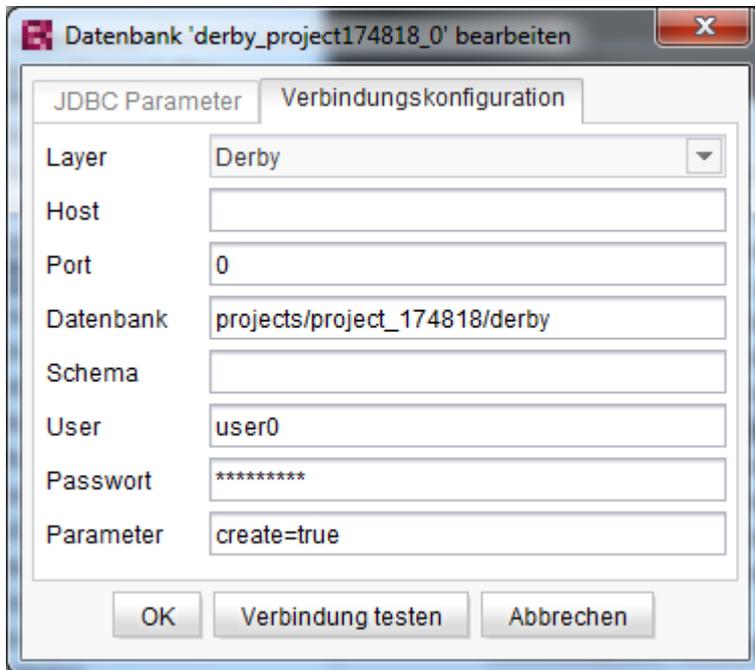


Abbildung 7-37: Datenbankanbindung konfigurieren (Verbindung)

Konfiguration der Verbindungs-Parameter für die Datenbankanbindung.

Layer: Über die Combobox wird die Klasse angegeben, die den Datenbank-Layer für dieses spezielle Datenbanksystem implementiert (vgl. Parameter `layerclass` Kapitel 4.8.4.1 Seite 158).

Host: Hostname für die Bindeadresse des Datenbank-Servers. Bei Angabe eines Derby-Layers sollte dieses Feld leer bleiben.

Port: Portnummer für die Bindeadresse des Datenbank-Servers. Bei Angabe eines Derby-Layers sollte dieses Feld leer bleiben.

Datenbank: Name der Datenbank, in welcher die Daten abgelegt werden sollen. Standardmäßig wird dieser Wert durch den Name des Datenbank-Layers vorgegeben.

Schema: Es existieren zwei Schema-Typen:



- **Standard-Layer:** Hier wird ein vorhandenes DB-Schema in der Layer-Konfiguration mit dem Parameter `jdbc.SCHEMA` spezifiziert. Der Name des Schemas muss identisch mit dem Namen des Schemas innerhalb der Datenbank sein. In diesem Fall werden alle Datenbank-Inhalte aus FirstSpirit in dieses Schema geschrieben.
- **DBA-Layer:** Hier ist der Parameter `jdbc.SCHEMA` nicht definiert. In diesem Fall legt jedes FirstSpirit-Schema ein eigenes Datenbank-Schema für die jeweiligen Datenbank-Inhalte an.

User: gültiger Login-Name eines Datenbank-Users. Mit diesem Account baut der FirstSpirit-Server zur Laufzeit eine Verbindung zur Datenbank auf (vgl. Parameter `jdbc.user` Kapitel 4.8.4.1 Seite 158).

Passwort: Gültiges Passwort zum Login unter `<database>.jdbc.USER` (vgl. Parameter `jdbc.password` Kapitel 4.8.4.1 Seite 158).

Parameter: Eingabemöglichkeit für weitere Datenbankspezifische Parameter.

Die Konfiguration kann geändert und anschließend mit einem Klick auf "OK" gespeichert werden. Bevor eine Änderung gespeichert wird, sollte immer die Verbindung zur Datenbank getestet werden. Genauere Informationen zur Konfiguration der Datenbank siehe Kapitel 4.8.4 Seite 156.

Verbindung testen: Nachdem die Konfiguration für die neue Datenbank abgeschlossen ist, kann durch einen Klick auf diesen Button überprüft werden, ob alle Angaben korrekt vorgenommen wurden.



7.3.6 Sprach-Vorlagen

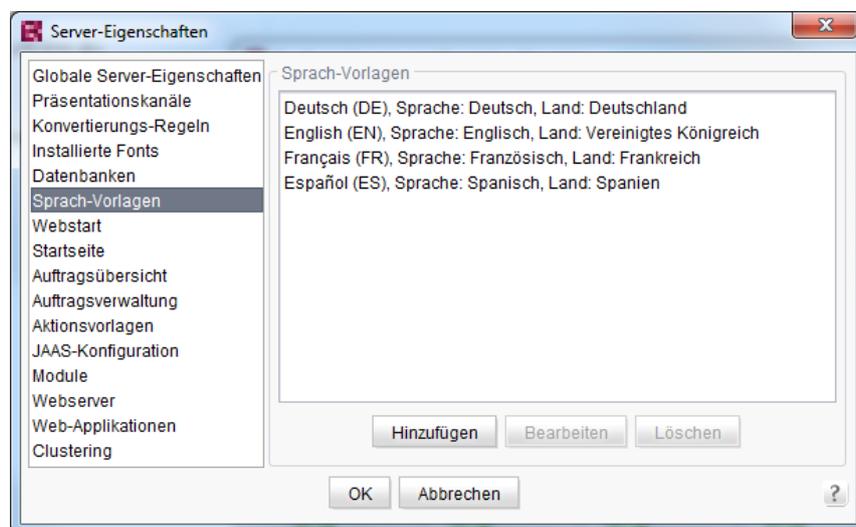


Abbildung 7-38: Server-Eigenschaften – Sprachvorlagen

FirstSpirit unterstützt die mehrsprachige Ausgabe einer Publikation. Wird in einem FirstSpirit Projekt eine neue Sprache benötigt, muss die neue Sprache zuerst auf dem Server angelegt werden.

Hinzufügen: Über den Button kann eine neue Sprache in das System integriert werden. Ein Klick öffnet das Dialogfenster "Neue Sprache" (siehe Abbildung 7-39). Eine neu angelegte Sprache steht nach der Definition allen Projekten zur Verfügung, muss jedoch vor ihrer Verwendung in das konkrete Projekt eingebunden werden (siehe Kapitel 7.4.5 Seite 295). Erst dann wird der Sprachreiter in der Redaktionsumgebung angezeigt und kann mit Inhalt gefüllt werden.

Bearbeiten: Mit einem Klick auf den Button kann eine bestehende Sprache bearbeitet werden (siehe Abbildung 7-39).

Löschen: Mit einem Klick auf den Button kann eine bestehende Sprache gelöscht werden.

Das Bearbeiten oder Löschen einer Sprach-Vorlage führt nicht dazu, dass Inhalte die bereits für ein Projekt in dieser Sprache eingepflegt wurden, verloren gehen. Diese Inhalte bleiben erhalten, solange die Sprache in den Projekteigenschaften eines Projekts erhalten bleibt (siehe Kapitel 7.4.5 Seite 295), da diese mit einer Kopie der Sprachvorlage arbeiten. Nach dem Löschen einer Sprach-Vorlage in den Servereigenschaften kann die entsprechende Sprache aber keinem Projekt



innerhalb der Projekteigenschaften neu hinzugefügt werden.

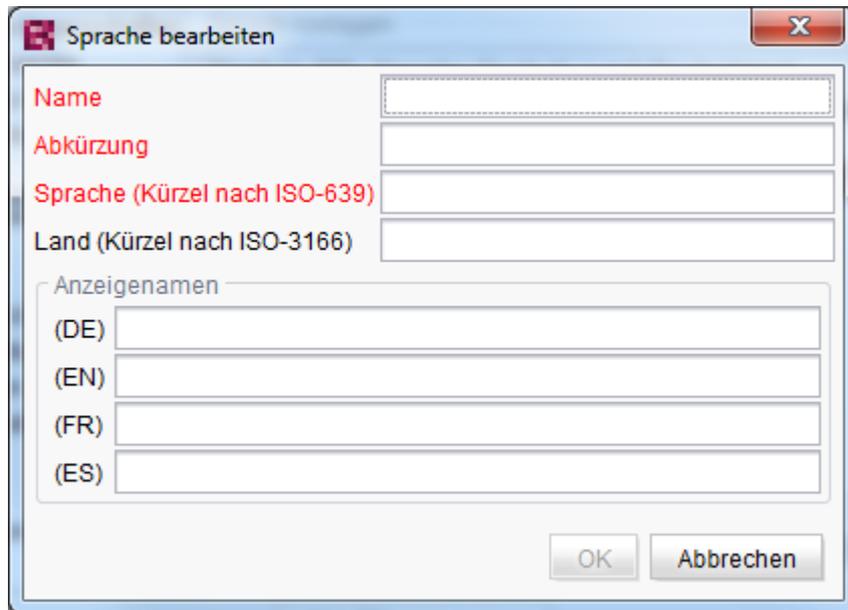


Abbildung 7-39: Server-Eigenschaften – neue Sprache anlegen

Die rote Schrift kennzeichnet ein Pflichtfeld. Bei einer gültigen Eingabe färbt sich die Schriftfarbe schwarz.

Name: In diesem Feld muss ein Name für die Sprache eingegeben werden. Der Name der Sprache darf aus beliebigen Zeichen bestehen.

Abkürzung: In diesem Feld muss ein Namens Kürzel für die Sprache definiert werden. Unter dieser Bezeichnung wird die Sprache innerhalb der FirstSpirit Redaktionsumgebung in den Sprachregistern angezeigt. Der Wert, der hier angegeben wird, wird dazu auch in Vorlagen für das Attribut *lang* verwendet (siehe FirstSpirit Online Dokumentation, Attribut *lang* in allen Eingabekomponenten, Bereich *Vorlagenentwicklung / Formulare*). Folgende Zeichen können dabei beliebig oft verwendet werden: -, _, 0-9 und A-Z. Die Abkürzung der Sprache kann sich z. B. an den ISO-Normen 3166 oder 639 orientieren. Für Sprachen, die in mehreren Ländern gesprochen werden, z. B. britisches Englisch und amerikanisches Englisch, bieten sich zusammengesetzte Abkürzungen an, z. B. *EN-GB* oder *EN_GB* für britisches und *EN-US* oder *EN_US* für amerikanisches Englisch.

Die Werte für Sprache und (optional) Land bestimmen das spätere Locale der Sprache:

Sprache: In diesem Feld muss das 2-Buchstaben-Kürzel nach ISO-639 für die neue Sprache angegeben werden, z. B. *de* für Deutsch oder *en* für Englisch. Es können



nur Kleinbuchstaben eingegeben werden. (Erster Teil der LOCALE)

Land: In diesem Feld muss das 2-Buchstaben-Kürzel für das Land der neuen Sprache nach ISO 3166 eingegeben werden, z. B. *DE* für Deutschland oder *GB* für England. Es können nur Großbuchstaben eingegeben werden. (Zweiter Teil der LOCALE)

So kann beispielsweise eine Trennung von en_US (englische Sprache im Gebiet der Vereinigten Staaten von Amerika; amerikanisch) zu en_GB (englische Sprache im Gebiet des Vereinigten Königreichs; britisch) erfolgen.

Für jede Sprach-Vorlage können dabei sprachabhängige Anzeigenamen definiert werden. Im Formularbereich "Anzeigenamen" werden dazu die entsprechenden Eingabefelder angezeigt. Sprachabhängige Anzeigenamen können zu allen Redaktionssprachen eines Projekts definiert werden.



7.3.7 Webstart

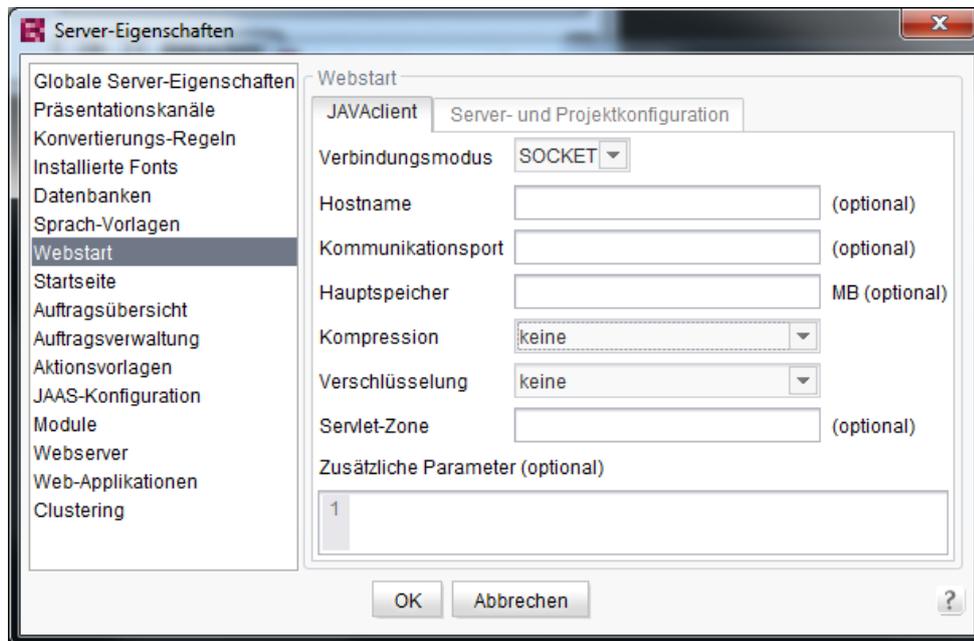


Abbildung 7-40: Server-Eigenschaften – Webstart konfigurieren (Schnellstart)

Die Konfiguration der Schnellstart-Einträge in FirstSpirit erfolgt über JNLP-Dateien. Die Standardkonfigurationen (Standard-JNLP-Dateien) können über den Menüeintrag "Webstart" definiert werden.

Register JavaClient: Konfiguration für den Start des JavaClients und der Schnellstart-Einträge auf der Startseite. Die hier konfigurierten Parameter wirken sich auf alle Schnellstart-Einträge der Startseite vom Typ JavaClient "Java" aus, sofern nicht explizit andere Parameter für den Eintrag im Bereich "Startseite" definiert wurden.

Die Konfiguration für einzelne Schnellstart-Einträge kann über den Dialog "Startseite" erfolgen. Damit werden die allgemeingültigen Werte im Bereich "Webstart" überschrieben (siehe Kapitel 7.3.8 Seite 250).

Die Konfigurationsmöglichkeiten entsprechen der benutzerspezifischen Webstart-Konfiguration (siehe Kapitel 6.3.4.1 Seite 192).

Register Server- und Projektkonfiguration: Konfiguration für die Server- und Projektkonfiguration der Startseite. Die Parameter sind analog zu den Schnellstart-Einträgen im Register "JavaClient".



7.3.8 Startseite

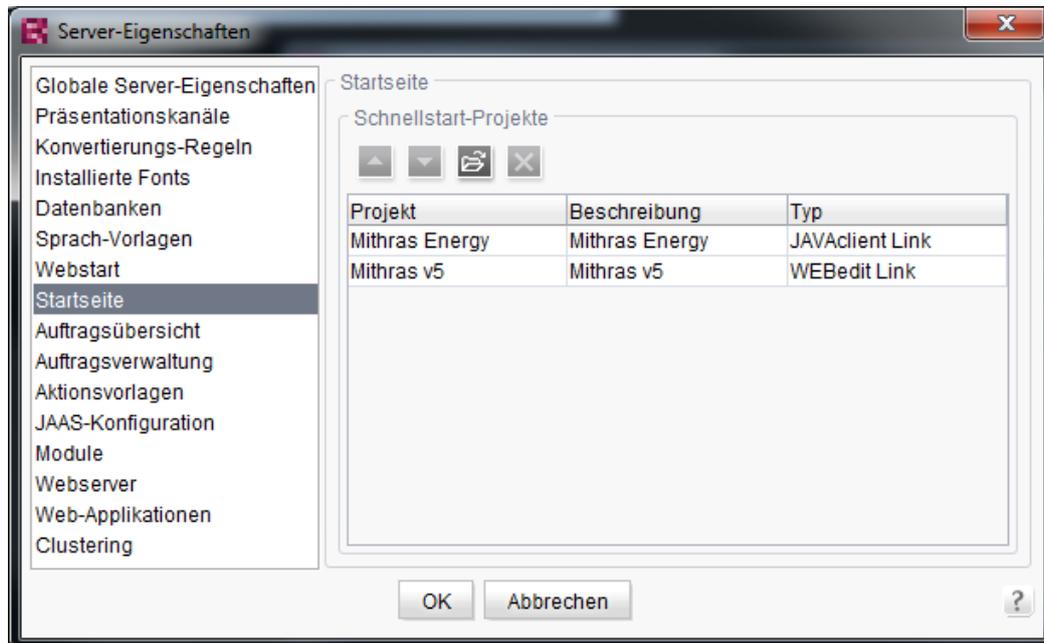


Abbildung 7-41: Server-Eigenschaften – Startseite konfigurieren

Schnellstart-Projekte: Über das Auswahl-Icon kann aus allen Projekten, die sich auf dem Server befinden, eine Auswahl getroffen werden. Die ausgewählten Projekte werden anschließend in der tabellarischen Übersicht dargestellt und erscheinen auf der FirstSpirit Startseite unterhalb der Schnellstart-Einträge (vgl. Kapitel 6.3.1 Seite 188). Mit einem Klick auf das Entfernen-Icon können zuvor in der Tabelle markierte Projekte wieder gelöscht werden. Eine Sortierung der Einträge ist über die Pfeiltasten möglich.

Ein Klick auf einen "Schnellstart Eintrag" auf der Startseite, öffnet direkt das hier referenzierte Projekt, ohne dass es der Benutzer zuvor über die Projektauswahlliste selektieren muss. Abhängig von der Konfiguration können für ein Projekt sowohl der JavaClient als auch der WebClient (nur für Projekte mit WebClient-Konfiguration) referenziert werden.

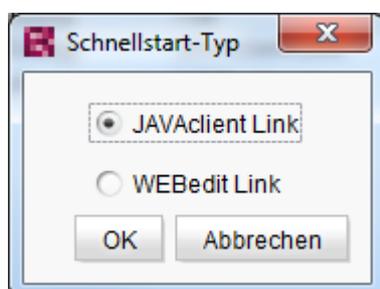


Abbildung 7-42: Server-Eigenschaften – Anlegen der Schnellstart-Referenzen



Abhängig davon, welche Einstellung gewählt wird, wird das ausgewählte Projekt im JavaClient oder im WebClient geöffnet. Die Schnellstart-Einträge sind zur besseren Unterscheidung mit dem Schriftzug "JAVA" oder "WEB" versehen.

Für Schnellstart-Einträge, die im JavaClient geöffnet werden, können weitere Verbindungseinstellungen (Einstellungen für Java Web Start) definiert werden. Der Dialog "Webstart Einstellungen" wird mit einem Doppelklick auf den entsprechenden Schnellstart-Eintrag geöffnet (siehe Abbildung 7-43). Die Konfigurationsmöglichkeiten entsprechen der benutzerspezifischen Webstart-Konfiguration (siehe Kapitel 6.3.4.1 Seite 192).

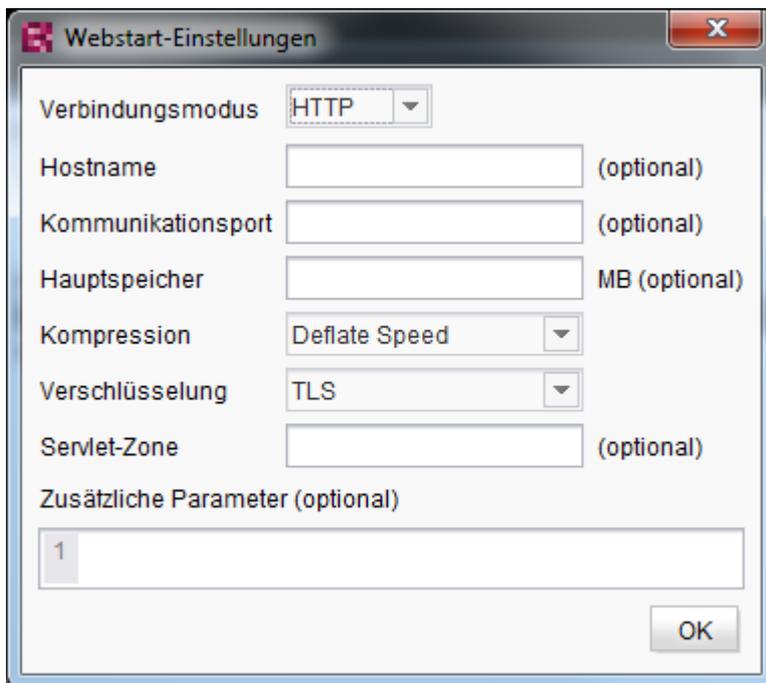


Abbildung 7-43: Webstart-Einstellungen für Projekt-Verweise (nur JavaClient)

Auswertungsreihenfolge: Die initiale Konfiguration wird über die Konfigurationsdatei `fs-server.conf` vorgenommen (siehe Kapitel 4.3.1.1 Seite 32). Die dort definierten Parameter können über die Server- und Projektkonfiguration überschrieben werden und zwar unter dem Menüeintrag "Webstart" einmal für alle Java Web Start-Anwendungen (über die JNLP-Dateien) (siehe Kapitel 7.3.7 Seite 249) und unter "Startseite" in den Eigenschaften der einzelnen Schnellstarteinträge. Die Auswertungsreihenfolge ist:

1. Schnellstart-Eigenschaften
2. JNLP-Datei
3. `fs-server.conf`



Sind Verbindungseinstellungen auf der Startseite definiert und aktiviert (siehe Kapitel 6.3.4.1 Seite 192), werden diese zusätzlich vorrangig ausgewertet.

Bei deaktivierten Verbindungseinstellungen werden die `fs.url`-Parameter der `fs-server.conf` (beginnend mit "fs.url.", siehe Kapitel 4.3.1.1 Seite 32) nicht durch die entsprechenden Parameter, die in den Servereigenschaften in den Bereichen "Webstart" und "Startseite" definiert sind, überschrieben.

7.3.9 Auftragsübersicht, Auftragsverwaltung und Aktionsvorlagen

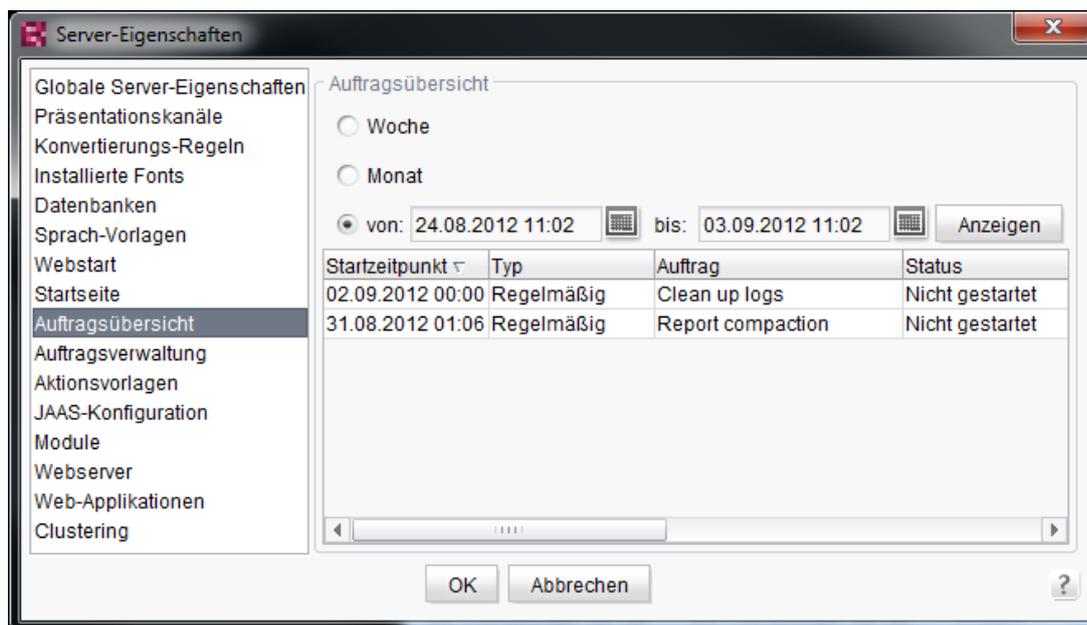


Abbildung 7-44: Server-Eigenschaften – Auftragsübersicht

Zur Auftragsübersicht siehe Kapitel 7.5.1 ab Seite 351.

Zur Auftragsverwaltung siehe Kapitel 7.5.2 ab Seite 354.

Zu Aktionsvorlagen siehe Kapitel 7.5.3 ab Seite 358.



7.3.10 JAAS-Konfiguration

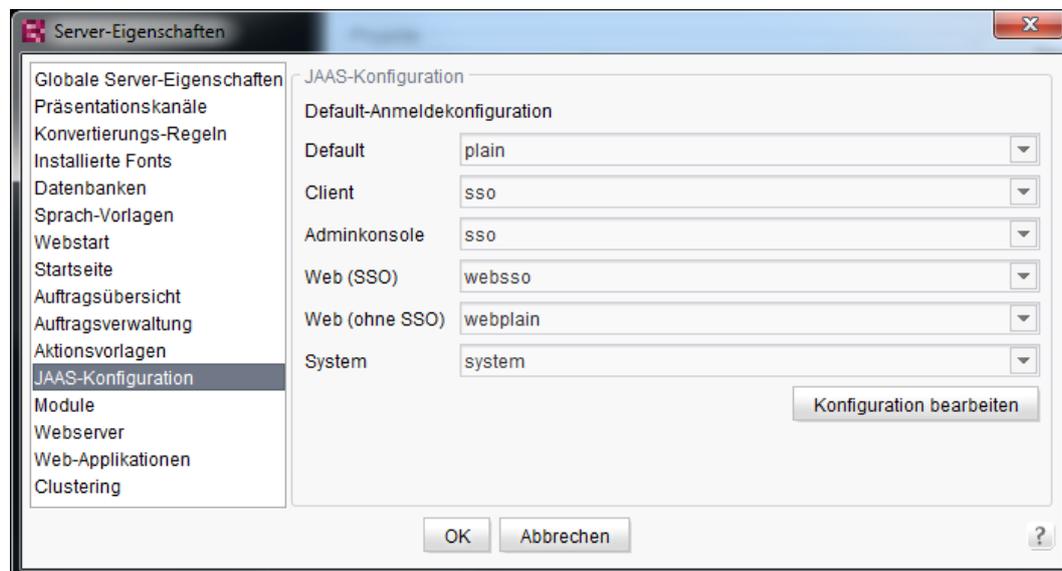


Abbildung 7-45: Server-Eigenschaften – JAAS²⁹-Konfiguration

In diesem Bereich können wichtige Konfigurationseinstellungen für den Anmeldevorgang definiert und angepasst werden. Die hier definierten Einstellungen werden in der Konfigurationsdatei `fs-jaas.conf` gespeichert (vgl. Kapitel 4.3.4 Seite 77). Bei einer Änderung der Datei über die Server- und Projektkonfiguration wird die Datei neu geschrieben und automatisch neu geladen.

Über die Klapplisten kann der jeweilige Anmeldevorgang für die unterschiedlichen Anwendungsmöglichkeiten (z. B. Anmeldung Client oder Anforderung einer Vorschau) definiert werden. Dieses Mapping wird innerhalb der Konfigurationsdatei `fs-server.conf` im Bereich: JAAS gespeichert (siehe Kapitel 4.3.1.6 Seite 42).

Konfiguration bearbeiten: Mit einem Klick auf den Button wird ein Editor für die Konfigurationsdatei `fs-jaas.conf` angezeigt. Die Datei enthält eine Beispielkonfiguration aller verfügbaren Login-Module. Über den Editor können weitere Module definiert werden. Beim Bestätigen der Eingaben mit "OK" werden die Änderungen in der `fs-jaas.conf` gespeichert und automatisch geladen.

²⁹ Java Authentication and Authorization Service



7.3.11 Module

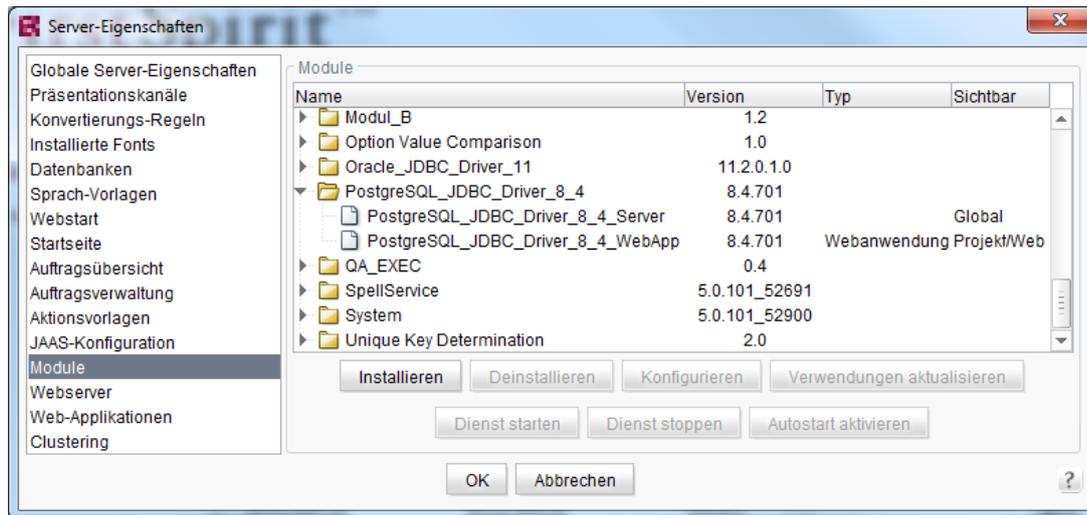


Abbildung 7-46: Server-Eigenschaften – Module

In diesem Bereich können FirstSpirit-Module installiert und konfiguriert werden. Initial ist das System-Modul mit den Standard-Komponenten vorhanden. Die Anzeige der Module ist alphabetisch (nach den Namen der Module) sortiert.



Bei Installation und Aktualisierung von Modulen, die entweder selbst oder durch direkt oder indirekt abhängige Dienste Grundlage für Daten sind, stehen diese Daten bis zum Neustart der auf diese zugreifenden Prozesse (Generierungen, Clients...) diesen Prozessen nicht mehr zur Verfügung.

Name: Name des Moduls bzw. einer Komponente des Moduls. Die Module werden innerhalb des Dialogs mit einem Ordner-Icon gekennzeichnet. Zu jedem Modul gehören eine oder mehrere Komponenten, die unterhalb des Moduls angezeigt werden. Komponenten werden mit einem Datei-Icon gekennzeichnet. Es gibt unterschiedliche Arten von Komponenten (siehe "Typ").

Version: Versionsnummer des Moduls bzw. der Komponente, die auf dem FirstSpirit-Server installiert ist.

Typ: Art der Komponente. Es wird zwischen folgenden Komponenten unterschieden:

- **Bibliothek:** Eine Bibliothek ist eine konfigurationslose Sammlung von Klassen, verpackt in einer oder mehreren Jar-Dateien. Sie stehen nach der Installation auf dem FirstSpirit-Server, innerhalb der Clients, in Skripten und anderen Modulen zur Verfügung.



- **Editor:** Ein Editor ist eine Kombination aus GUI- und Render-Komponente. Hierüber ist es möglich, den FirstSpirit-Client um eigene Eingabemöglichkeiten zu erweitern. (Beispiel CMS_INPUT_PERMISSION-Eingabekomponente zur Definition von Rechten. Diese Eingabekomponente arbeitet mit dem passenden Dienst zusammen, der die Gruppenseiten vom Server lädt und bereitstellt).
- **Dienst:** Ein Dienst ist eine Serverkomponente, die über eine öffentliche Schnittstelle aus Eingabekomponenten oder Skripten angesprochen werden kann. (Beispiele sind die Rechtschreibprüfung oder der Dienst der Rechte-Eingabekomponente CMS_INPUT_PERMISSION).
- **Webanwendung:** Eine Webanwendung definiert JSP-Tags und Servlets, die in Projekten verwendet und aufgerufen werden können. (Die Personalisierung und Suche sind Beispiele für Webanwendungen).
- **Webserver:** Eine Webserver-Komponente stellt Funktionalitäten zur Verfügung, um Webanwendungen zu installieren und zu deinstallieren. (Beispiele hierfür sind die interne Webserver-Steuerung oder eine Tomcat-Unterstützung) (siehe dazu Kapitel 7.3.12 Seite 261).

Sichtbar: Komponenten stehen nach der Installation nur innerhalb eines bestimmten Bereichs zur Verfügung. Es wird zwischen folgenden Bereichen unterschieden:

- **Global:** Globale (systemweite) Komponenten stehen nach der Installation auf dem Server zur Verfügung; damit sind diese Komponenten auch innerhalb der FirstSpirit-Clients in allen Skripten und anderen Komponenten verfügbar. (Beispiel: alle Services; z. B. Permission-Service)
- **Projekt:** Projekt-lokale Komponenten können nach der Installation den gewünschten Projekten über deren Projekteigenschaften hinzugefügt werden (siehe Kapitel 7.4.18 Seite 327). Anschließend besteht die Möglichkeit, diese Komponenten zu konfigurieren. Die Konfigurationsmöglichkeit ist abhängig von der installierten Komponente.
- **Web:** Web-lokale Komponenten können nach der Installation den einzelnen Web-Bereichen (preview, staging, live) innerhalb der gewünschten Projekte zugefügt werden (siehe Kapitel 7.4.18 Seite 327). Die Konfigurationsmöglichkeit ist abhängig von der installierten Komponente. Die Komponenten können für die jeweiligen Projekte unterschiedlich konfiguriert werden.



Installieren / Aktualisieren: Mit einem Klick auf den Button "Installieren" öffnet sich ein Dateiauswahldialog. Hier kann die zu installierende fsm-Datei ausgewählt werden (beispielsweise das Modul zu FirstSpirit Personalisation). Die erfolgreich installierte Datei wird anschließend im Dialog "Server-Eigenschaften" angezeigt (siehe Abbildung 7-46).

Enthält die installierte Komponente einen Dienst, wird außerdem ein Dialog angezeigt, mit dem die Autostart-Option für den Dienst konfiguriert werden kann. Wird der Dialog mit "Ja" bestätigt, wird der Dienst bei jedem Neustart des Servers automatisch gestartet (vgl. Autostart aktivieren / deaktivieren):

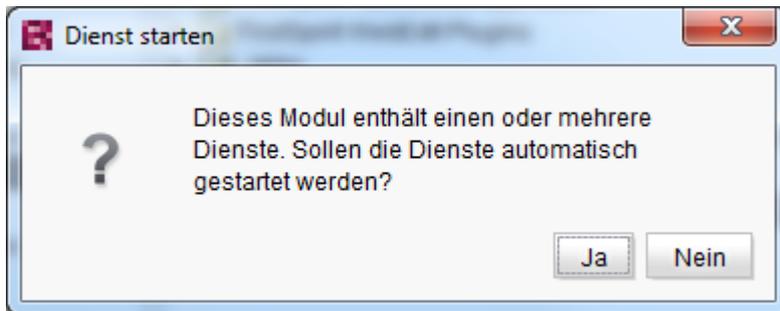


Abbildung 7-47: Autostart für Dienste installieren

Steht eine neuere Version des Moduls zur Verfügung, kann das Modul ebenfalls über den Button "Installieren" auf dem FirstSpirit-Server aktualisiert werden. Es ist nicht notwendig, das Modul dafür zuvor zu deinstallieren. Obwohl die ursprüngliche Konfiguration erhalten bleibt, sind jedoch eventuell Anpassungen in der Konfiguration oder innerhalb der das Modul verwendenden Projekte erforderlich. Daher wird die neue Version des Moduls nicht automatisch in die Projekte übernommen und die Aktualisierung innerhalb eines Projekts muss manuell durchgeführt werden (siehe Kapitel 7.4.17 Seite 325 und Kapitel 7.4.18 Seite 327). Die Verwendungen können über den Button "Verwendungen aktualisieren" vorgenommen werden.



Nach dem Update von Modulen, die Abhängigkeiten zu Modulen mit Diensten ("Service") haben, müssen diese Dienste manuell neugestartet werden. (Buttons "Dienst stoppen" / "Dienst starten" bzw. über das Server-Monitoring, Kapitel 8.6.2.4 Seite 463.)

Deinstallieren: Zum Deinstallieren wird das gewünschte Modul in der Übersicht der installierten Module ausgewählt und mit einem Klick auf den Button deinstalliert. Das System-Modul kann nicht deinstalliert werden.

Es können nur Module deinstalliert werden, die nicht in einem Projekt verwendet



werden. Beim Versuch ein Modul zu deinstallieren, das noch verwendet wird, erscheint eine Fehlermeldung mit einer Liste aller Projekte, die das Modul zurzeit verwenden. Zur Deinstallation muss das Modul zunächst aus diesen Projekten entfernt werden (siehe Kapitel 7.4.18 Seite 327).

Konfigurieren: Mit einem Klick auf den Button kann die selektierte Komponente konfiguriert werden. Die Konfigurationsmöglichkeit ist abhängig von der ausgewählten Komponente (siehe Beispiel "Spell Service"). Außerdem können Rechte für ein Modul gesetzt werden (siehe Kapitel 7.3.11.1 Seite 259):

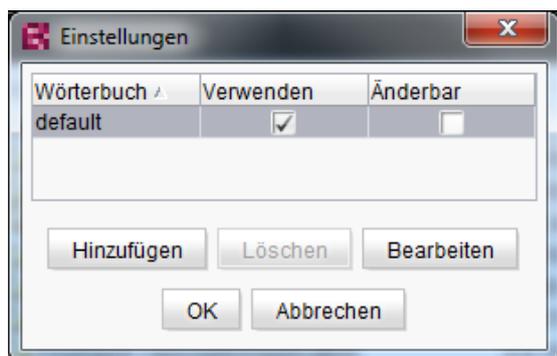


Abbildung 7-48: Konfigurationsbeispiel Spell Service

Verwendungen aktualisieren: Für das komfortable Aktualisieren eines Moduls, das eine Projekt- oder eine Webanwendung beinhaltet, steht der Button "Verwendungen aktualisieren" zur Verfügung. Beim Klick auf den Button öffnet sich ein Dialogfenster zur Aktualisierung der Projekt- bzw. der Webanwendungen für alle Projekte, die diese Anwendung bisher verwenden:

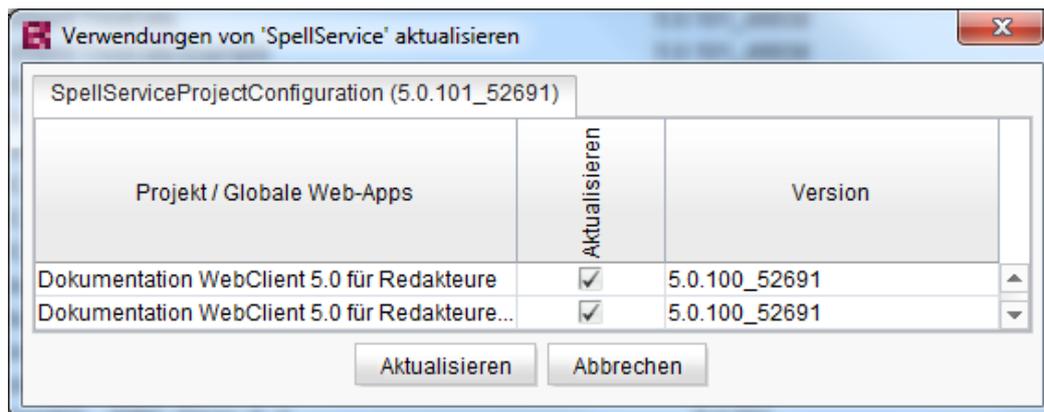


Abbildung 7-49: Alle Verwendungen eines Moduls aktualisieren

Die Aktualisierung muss also nicht mehr einzeln für jedes Projekt über die Projekteigenschaften ausgeführt werden, sondern kann zentral über die Servereigenschaften gesteuert werden.



Dienst starten: Mit einem Klick auf den Button kann ein Dienst gestartet werden. Das Starten eines Dienstes ist auch über das Server-Monitoring möglich (siehe Kapitel 8.6.2.4 Seite 463).



Nach dem Update von Modulen, die Abhängigkeiten zu Modulen mit Diensten ("Service") haben, müssen diese Dienste manuell neugestartet werden.

Dienst stoppen: Mit einem Klick auf den Button kann ein gestarteter Service gestoppt werden. Das Stoppen eines Dienstes ist auch über das Server-Monitoring möglich (siehe Kapitel 8.6.2.4 Seite 463).

Autostart aktivieren/deaktivieren: Mit einem Klick auf den Button kann der automatische Start für einen Dienst aktiviert bzw. deaktiviert werden. Wird die Option Autostart aktiviert, wird der Dienst bei jedem Server-Neustart automatisch gestartet. Wird die Option Autostart deaktiviert, muss der Dienst bei jedem Server-Neustart manuell gestartet werden. Diese Einstellung kann initial auch beim Installieren bzw. Aktualisieren von Diensten gesetzt werden.



7.3.11.1 Vertrauenswürdige Module

Der FirstSpirit-JavaClient und die FirstSpirit Server- und Projektkonfiguration werden über eine JNLP-Datei, d.h. über Java Web Start ausgeführt. Daraus ergeben sich Einschränkungen in der Nutzung einiger Funktionalitäten für nicht von e-Spirit signierte Module bzw. in den Jar-Archiven enthaltene Klassen. Java-Programme laufen normalerweise in einer "Sandbox" ab. Das bedeutet, sie besitzen keinen vollwertigen Zugriff auf den Rechner (und dessen Ressourcen), auf dem sie ausgeführt werden. Der Zugriff auf lokale Ressourcen wie Dateien, Zwischenablage, Netzwerk etc. erfolgt über einen Security-Manager.

Die FirstSpirit-internen Module sind mit dem "e-Spirit AG"-Schlüssel signiert. Dieser ist wiederum Bestandteil der FirstSpirit-internen Security-Policy. Des Weiteren ist der Schlüssel von einer Root-Authority bestätigt, die wiederum dem Java Zertifikat Manager bekannt ist.

Externe Komponenten oder Module, die auf sicherheitsrelevante Funktionalitäten zugreifen, können komfortabel über die FirstSpirit Server- und Projektkonfiguration konfiguriert werden. Dabei kann jedes installierte Modul (mit Ausnahme der FirstSpirit System-Module) optional mit Rechten auf lokale System-Ressourcen ausgestattet werden. Damit besteht die Möglichkeit, einem Modul, das sicherheitsrelevante Operationen durchführt, z. B. den Zugriff auf die Zwischenablage (`java.awt.AWTPermission ClipboardAccess`), zu vertrauen. Diesem Modul können Rechte zur Durchführung der Operationen zugewiesen werden. Dies geschieht intern über den FirstSpirit Security-Manager/Classloader.

Die Konfigurationsoberfläche zum Setzen der Modulrechte in der FirstSpirit Server- und Projektkonfiguration stellt sich wie folgt dar:



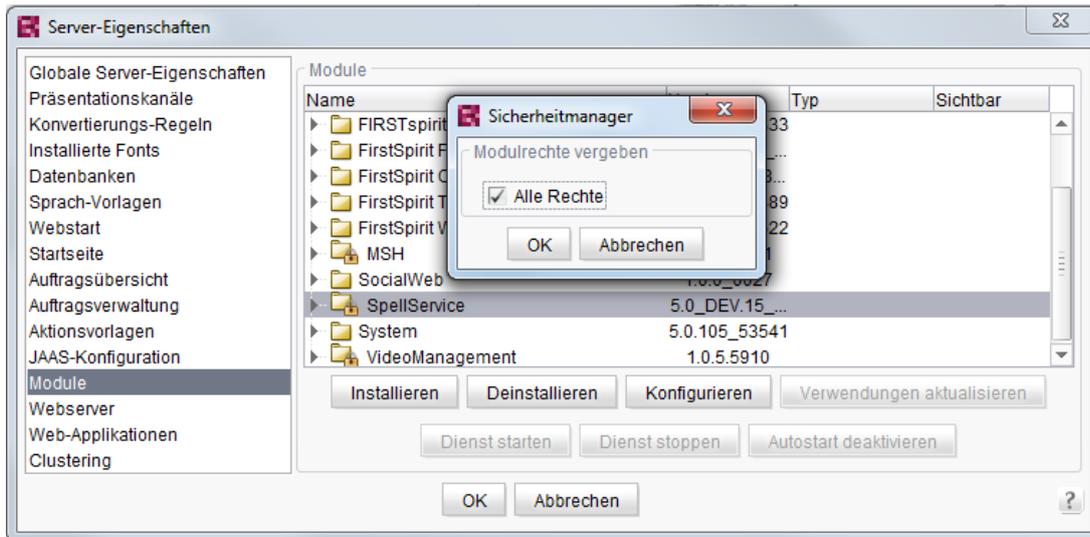


Abbildung 7-50: Sicherheitsmanager – Alle Rechte für dieses Modul vergeben



Wird eine externe Komponente oder ein Modul als vertrauenswürdig eingestuft, kann nicht sichergestellt werden, dass die Zugriffsschutz-Mechanismen von FirstSpirit voll wirksam sind. Eventuell auftretende Fehlfunktionen können nicht mehr eindeutig zugeordnet werden, so dass eine Fehlerdiagnose erschwert wird oder unmöglich ist. Im Rahmen der Produktwartung von FirstSpirit werden daher Systemkonfigurationen mit als vertrauenswürdig eingestuft, externen Komponenten bzw. Modulen nicht akzeptiert.

Weiterführende Informationen siehe "FirstSpirit Entwickler-Handbuch für Komponenten".



7.3.12 Webserver

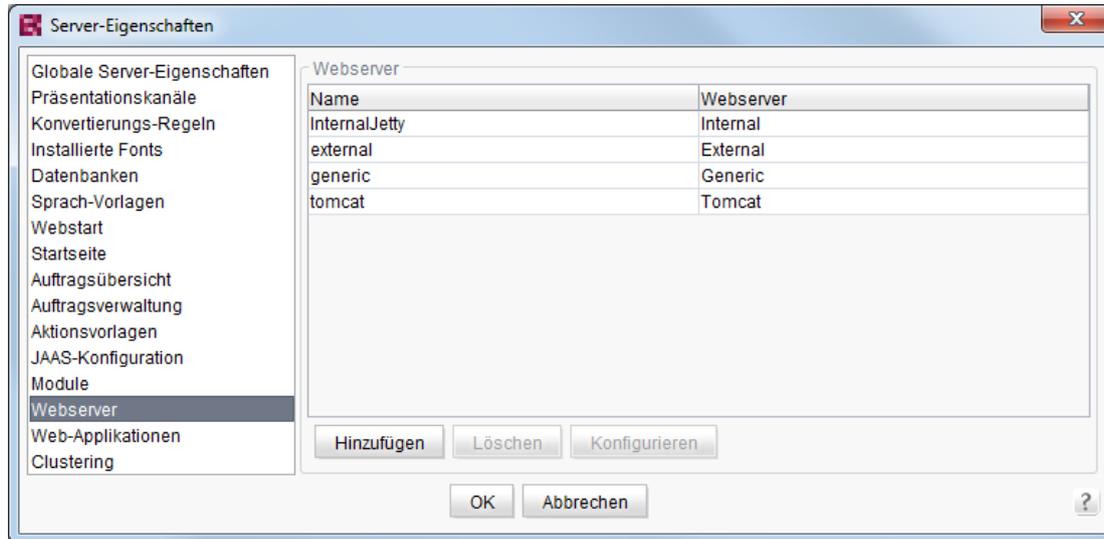


Abbildung 7-51: Server-Eigenschaften – Webserver

In diesem Bereich kann die Webserver-Steuerung für das Installieren und Deinstallieren innerhalb der Projektbereiche eingefügt werden (siehe Kapitel 7.4.17 Seite 325).

Die Steuerung für den internen Webserver Jetty ist standardmäßig vorhanden. Der entsprechende Eintrag wird in der Tabelle angezeigt. Diese Webserver-Steuerung kann nicht verändert oder gelöscht werden und steht automatisch, direkt nach der Installation von FirstSpirit, zur Verfügung.

Neben diesem Standardeintrag können in diesem Bereich beliebige generische bzw. externe Webserver-Steuerungen sowie Tomcat-Webserver hinzugefügt und konfiguriert werden.

Name: Eindeutiger Name unter dem die Instanz des Webservers hinzugefügt wurde.

Webserver: Innerhalb des Standard-Moduls "System" stehen vier unterschiedliche Arten von Webservern zur Verfügung:

- Internal (Interner Webserver): Für den internen Jetty steht direkt nach der Installation von FirstSpirit eine Steuerung (Installation bzw. Deinstallation von Webanwendungen) zur Verfügung.
- External (Externer Webserver): Dabei handelt es sich um einen externen Webserver, der von FirstSpirit nicht unterstützt wird. Das Installieren bzw. Deinstallieren von Webanwendungen auf dem Webserver, sowie alle sonstigen Konfigurationseinstellungen müssen daher manuell erfolgen.



- Tomcat: Dabei handelt es sich um einen von FirstSpirit unterstützten, externen Webserver. Über die FirstSpirit-Oberfläche lassen sich die Webanwendungen auf dem Webserver automatisch installieren und aktualisieren.
- Generic (Lokaler Webserver): FirstSpirit ermöglicht eine einfache Anbindung von Webservern, sofern es möglich ist, die Steuerung (Installation bzw. Deinstallation von Webanwendungen) für diese Webserver über ein BeanShell-Skript zu implementieren (vgl. Kapitel 7.3.12.2 Seite 263).

Mit einem Klick auf den Button "Hinzufügen" kann dem FirstSpirit-Server eine neue generische Webserver-Instanz oder eine neue externe Webserver-Instanz hinzugefügt werden:

- Interner Webserver: Der "InternalJetty" ist standardmäßig vorhanden, muss also nicht hinzugefügt werden.
- Generischer Webserver (siehe Kapitel 7.3.12.1 Seite 263)
- Externer Webserver (siehe Kapitel 7.3.12.3 Seite 265)
- Tomcat (siehe Kapitel 7.3.12.5 Seite 267)

Mit einem Klick auf den Button "Löschen" kann eine hinzugefügte Webserver-Instanz wieder entfernt werden. Die Standardinstanz "InternalJetty" kann nicht entfernt werden.

Mit einem Klick auf den Button "Konfigurieren" öffnet sich der Konfigurationsdialog für den Webserver.

- Interner Webserver: Diese Konfiguration kann nicht bearbeitet werden.
- Generischer Webserver: Handelt es sich um einen generischen Webserver, können hier die notwendigen Funktionalitäten zum Installieren bzw. Deinstallieren konfiguriert werden (siehe Kapitel 7.3.12.2 Seite 263).
- Externer Webserver: Eine Konfiguration für externe Webserver wird nicht unterstützt und muss manuell erfolgen (siehe Kapitel 7.3.12.4 Seite 266)
- Tomcat: Eine Konfiguration für Tomcat-Webserver wird unterstützt und ist automatisch möglich. (siehe Kapitel 7.3.12.6 Seite 268)



7.3.12.1 Hinzufügen eines generischen Webservers

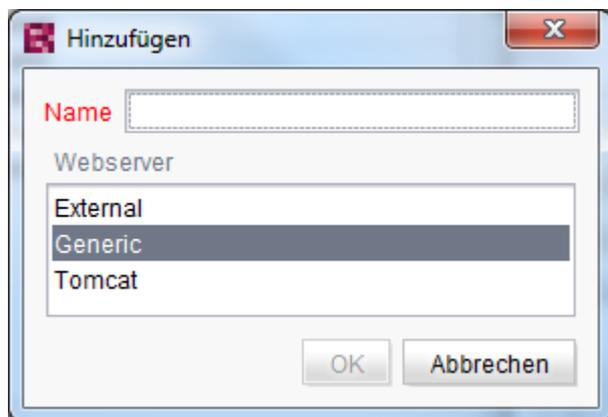


Abbildung 7-52: Generischen Webserver hinzufügen

Name: Hier muss ein eindeutiger Name für die jeweilige Webserver-Instanz vergeben werden.

Webserver: Das Hinzufügen einer neuen generischen Webserver-Instanz ist durch die Auswahl des Eintrags "Generic" möglich.



Nach dem Hinzufügen einer neuen generischen Webserver-Instanz müssen die Installations- und Deinstallations-Funktionalität implementiert werden (siehe Kapitel 7.3.12.2 Seite 263).

7.3.12.2 Konfigurieren eines generischen Webservers

Um einen neuen Webserver auf dem FirstSpirit-Server einzubinden, sind folgende Funktionalitäten notwendig:

1. Installieren
2. Installationsstatus prüfen
3. Deinstallieren

Mit einem Klick auf den Button "Konfigurieren" öffnet sich der folgende Dialog:



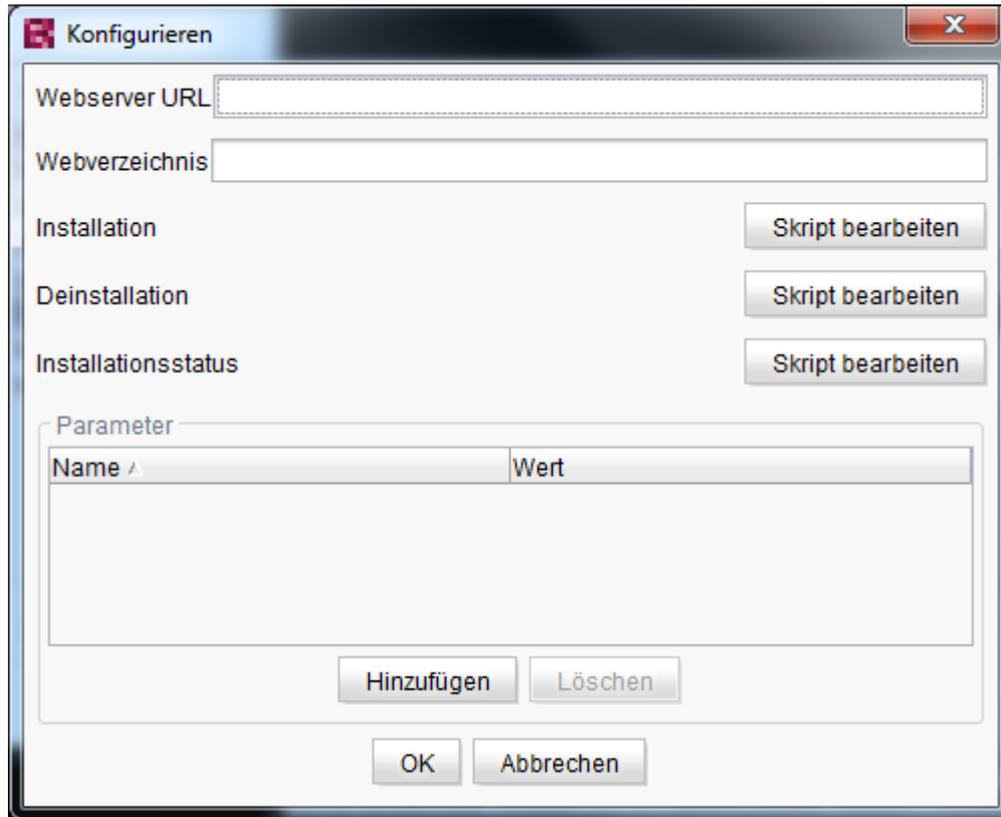


Abbildung 7-53: Webserver-Funktionen konfigurieren

Web-Server URL: In diesem Feld kann der URL des neuen Webserver eingetragen werden. Der URL wird beispielsweise benötigt, um die Verweise für die FirstSpirit-Webanwendungen auf der Startseite anzupassen.

Webverzeichnis: Pfadangabe zum Web-Verzeichnis des generischen Webserver für die Nutzung der FirstSpirit-Webanwendungen (z. B. fs5staging). Wird der generische Webserver beispielsweise für die Webanwendung fs5staging verwendet, werden die Projektdateien in das angegebene Verzeichnis des generischen Webserver generiert. Die Pfadangabe wird außerdem für die Ausführung der konfigurierten Skripte (beispielsweise zur Installation oder Deinstallation) benötigt.

Skript bearbeiten: Alle Funktionalitäten können mithilfe eines Beanshell-Skripts realisiert werden. Mit einem Klick auf den Button "Skript bearbeiten " (vor der jeweiligen Funktionalität) öffnet sich der Skript-Dialog.

Die gewünschten Funktionen können für jeden von FirstSpirit unterstützten Webserver skriptgesteuert ausgeführt werden. Die benötigten Parameter für die Skripte können im Bereich Parameter hinzugefügt werden. Die Funktionalität sollte vor dem Speichern mithilfe des Buttons "Testen" geprüft werden. Nach dem Speichern steht die Funktionalität im Bereich "Web-Applikationen" bzw.



"-komponenten" zur Verfügung:

- Server-Eigenschaften / Web-Applikationen (siehe Kapitel 7.3.13 Seite 269)
- Projekteigenschaften / Web-Komponenten (siehe Kapitel 7.4.18 Seite 327).

Zum Ausführen der Skripte kann das Webverzeichnis angegeben werden, indem sich der generische Webserver befindet (siehe "Webverzeichnis").

Parameter hinzufügen: Im Bereich "Parameter" (vgl. Abbildung 7-53) können Parameter hinzugefügt werden, die anschließend in allen Skript-Kontexten (von Installation bis Deinstallation) zur Verfügung stehen (Beispiel: Angabe eines Pfads).

Parameter löschen: Mit einem Klick auf den Button wird ein markierter Parameter aus der Liste gelöscht.



Das Löschen von Parametern kann dazu führen, dass die Skripte nicht mehr funktionieren und angepasst werden müssen.

7.3.12.3 Hinzufügen eines externen Webserver

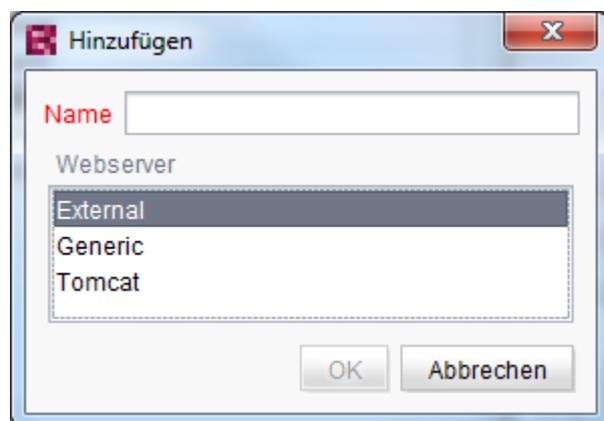


Abbildung 7-54: Externen Webserver hinzufügen

Name: Hier muss ein eindeutiger Name für die jeweilige Webserver-Instanz vergeben werden.

Webserver: Das Hinzufügen einer neuen externen Webserver-Instanz ist durch die Auswahl des Eintrags "External" möglich.





Nach dem Hinzufügen einer neuen externen Webserver-Instanz kann die URL des Webservers angegeben werden (siehe Kapitel 7.3.12.4 Seite 266). Eine weitere Unterstützung (Installation- bzw. Deinstallation) wird nicht angeboten und muss manuell erfolgen.

7.3.12.4 Konfigurieren eines externen Webservers

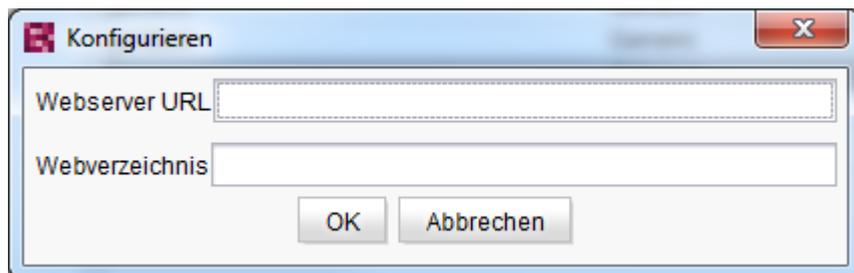


Abbildung 7-55: Konfiguration eines externen Webservers

Web-Server URL: In diesem Feld kann der URL des neuen Webservers eingetragen werden. Der URL wird beispielsweise benötigt, um die Verweise für die FirstSpirit-Webanwendungen auf der Startseite anzupassen.

Webverzeichnis: In diesem Feld kann der Pfad zum Verzeichnis des Webservers eingetragen werden. Dieser wird beispielsweise für das Löschen der Staging- und Preview-Verzeichnisse benötigt, wenn das zugehörige Projekt gelöscht wird. Auch für die Aktualisierung der Webapplikationen wird der Pfad ausgelesen und verwendet.



7.3.12.5 Hinzufügen eines Tomcat-Webserver

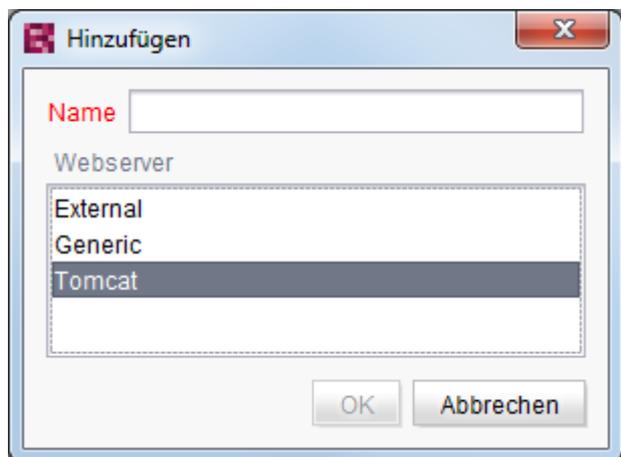


Abbildung 7-56: Tomcat-Webserver hinzufügen

Name: Hier muss ein eindeutiger Name für die jeweilige Webserver-Instanz vergeben werden.

Webserver: Das Hinzufügen einer neuen Tomcat-Webserver-Instanz ist durch die Auswahl des Eintrags "Tomcat" möglich.



Nach dem Hinzufügen einer neuen Tomcat-Webserver-Instanz müssen die Installations- und Deinstallations-Funktionalität konfiguriert werden (siehe Kapitel 7.3.12.6 Seite 268).



Weitere Informationen zur Verwendung von Tomcat siehe auch Kapitel 4.5 Seite 108).



7.3.12.6 Konfigurieren eines Tomcat-Webservers

Mit einem Klick auf den Button "Konfigurieren" öffnet sich der folgende Dialog:

The image shows a Windows-style dialog box titled "Konfigurieren" (Configure). It contains five text input fields stacked vertically, each with a label to its left: "Webserver URL", "Webverzeichnis", "Tomcat-Benutzer", "Tomcat-Passwort", and "Tomcat-Manager-URLs". At the bottom of the dialog, there are two buttons: "OK" and "Abbrechen" (Cancel). The dialog has a standard title bar with a close button (X) in the top right corner.

Abbildung 7-57: Konfiguration eines Tomcat-Webservers

Webserver URL: In dieses Feld muss die URL, die beim Aufrufen voreingestellt wird, eingetragen werden, z. B. `http://tomcat:123/fs5webedit`. Diese URL wird beispielsweise benötigt, um die Verweise für die FirstSpirit-Webanwendungen auf der Startseite anzupassen

Webverzeichnis: In dieses Feld muss der Pfad zum Webverzeichnis des Tomcat-Webservers für die Nutzung der FirstSpirit-Webanwendungen (z. B. `fs5staging`) eingetragen werden. Wird der Tomcat-Webserver beispielsweise für die Webanwendung `fs5staging` verwendet, werden die Projektdateien in das angegebene Verzeichnis des Tomcat-Webservers generiert. Wenn noch nicht vorhanden, wird das hier eingetragene Webverzeichnis bei der Generierung angelegt.

Tomcat-Benutzer: In dieses Feld muss der Benutzername eingetragen werden, mit dem auf die URL des Tomcat-Managers (siehe unten) zugegriffen werden kann. Benutzername und Passwort werden bei der Installation des Tomcat-Servers angegeben und können in der Datei `tomcat-users.xml` im Verzeichnis `conf` des Tomcat-Webservers nachgeschlagen werden.

Tomcat-Passwort: In dieses Feld muss das Passwort zum zuvor angegebenen Benutzernamen eingetragen werden, mit dem Zugriff auf die URL des Tomcat-Managers (siehe unten) besteht. Benutzername und Passwort werden bei der Installation des Tomcat-Servers angegeben und können in der Datei `tomcat-`



users.xml im Verzeichnis conf des Tomcat-Webservers nachgeschlagen werden.

Tomcat-Manager-URLs: Mit dem Tomcat Manager können die Webanwendungen eingesehen und verwaltet werden. Neben einer HTML-Oberfläche stellt er auch eine textuelle Oberfläche zur Verfügung. In dieses Feld muss die URL zur textuellen Oberfläche des Tomcat Managers eingetragen werden, z. B.

`http://localhost:8080/manager`

für Tomcat 6

oder

`http://localhost:8080/manager/text`

für Tomcat 7

Mehrere URLs können kommasepariert angegeben werden.

7.3.13 Web-Applikationen

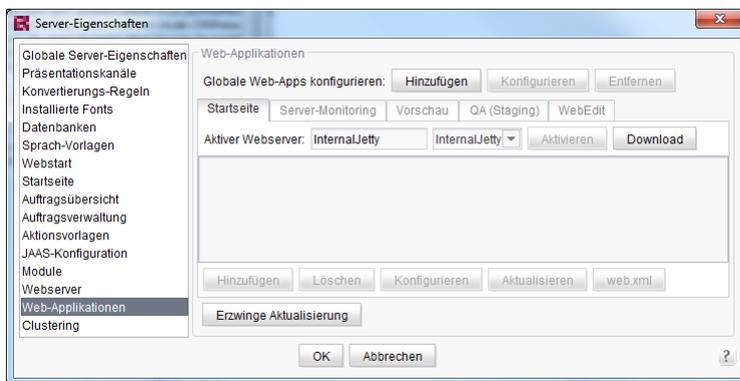


Abbildung 7-58: Server-Eigenschaften – Web-Applikationen

In diesem Bereich können die FirstSpirit Web-Applikationen konfiguriert werden. Es können auch eigene Web-Applikationen definiert und konfiguriert werden, die dann in allen Projekten des Servers zur Verfügung stehen ("global", siehe Kapitel 7.3.13.1 Seite 270). Damit können z. B. eigene Webanwendungen für das FirstSpirit AppCenter (siehe Dokumentation zum *FirstSpirit AppCenter*) installiert werden.



7.3.13.1 Installieren von globalen Web-Applikationen

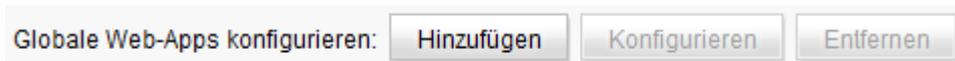


Abbildung 7-59: Globale Web-Apps konfigurieren

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

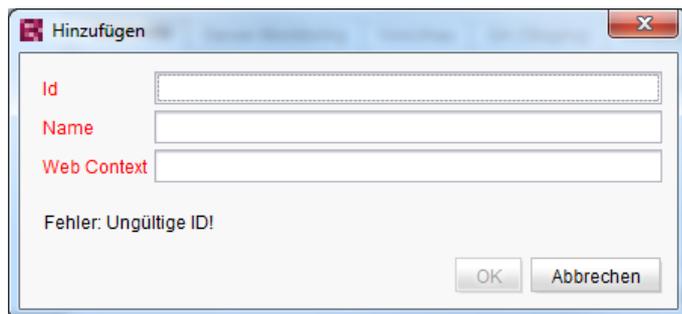


Abbildung 7-60: Neue Web-Applikation hinzufügen

Id: Hier muss ein eindeutiger Bezeichner für die Web-Applikation eingegeben werden. Mit diesem Bezeichner wird ein Unterverzeichnis auf dem Server für Web-Applikationen erstellt. Es dürfen nur Groß-, Kleinbuchstaben, Ziffern und Unterstriche verwendet werden. Die id kann nach dem Speichern nicht mehr geändert werden.

Über die FirstSpirit-Developer-API kann anhand der Id einer Web-Applikation die URL zu einer globalen Web-Applikation über das Interface `de.espirit.firstspirit.agency.LegacyModuleAgent` ermittelt 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 *fs5root*, *fs5preview*, *fs5staging*, *fs5webedit*, *fs5webmon*) übereinstimmen.



7.3.13.2 Konfigurieren von Web-Applikationen

Jede Web-Applikation wird über ein eigenes Register angezeigt und konfiguriert. Die Standard Web-Applikationen hierbei sind:

Startseite / Vorschau: Konfiguration für die FirstSpirit Startseite und die Vorschau (siehe Kapitel 6 Seite 185).

QA (Staging): Konfiguration für die generierten Projektinhalte QA (Staging).

WebEdit: Konfiguration für das Redaktionssystem WebClient.

Server-Monitoring: Konfiguration für das FirstSpirit Server-Monitoring (siehe Kapitel 8 Seite 429)

- Zu jeder Web-Applikation kann ein Webserver eingestellt werden:

Aktiver Webserver: Zur Auswahl

stehen alle Webserver, die im Bereich "Webserver" konfiguriert wurden (vgl. Kapitel 7.3.12 Seite 261).

Abhängig von der Art des Webserver sind zur Umstellung unterschiedliche Schritte erforderlich:

- Interner Webserver (siehe Kapitel 7.3.13.3 Seite 272)
- Generischer Webserver (siehe Kapitel 7.3.13.4 Seite 273)
- Externer Webserver (siehe Kapitel 7.3.13.5 Seite 273)

Aktivieren: Mit einem Klick auf den Button "*Aktivieren*" wird die Konfiguration der Web-Applikation auf den selektierten Webserver umgestellt, der anschließend als aktiver Webserver für die Anwendung angezeigt wird. Die Verweise auf die Web-Applikation (z. B. auf der Startseite) werden dabei automatisch auf den neuen Webserver umgestellt.

Installieren: Mit einem Klick auf den Button "*Installieren*" wird die Webanwendung auf dem ausgewählten Webserver installiert. Ist der Button deaktiviert, wurde die Webanwendung bereits installiert. Handelt es sich beim ausgewählten Webserver, um einen externen Webserver oder um einen generischen Webserver (ohne die erforderliche Skript-Funktionalität), wird stattdessen der Button "Download" angezeigt (vgl. Kapitel 7.3.13.5 Seite 273).



Download: Mit einem Klick auf den Button "*Download*" kann eine WAR-Datei der Anwendung herunter geladen werden, die manuell auf dem Webserver installiert werden muss. Der Button wird nur für die Konfiguration externer Webserver oder die Konfiguration generischer Webserver (ohne die erforderliche Skript-Funktionalität) angezeigt.

Deinstallieren: Mit einem Klick auf den Button "*Deinstallieren*" wird die Webanwendung auf dem ausgewählten Webserver deinstalliert. Ist der Button deaktiviert, wurde die Webanwendung noch nicht installiert. Wird eine Webanwendung deinstalliert, wird der entsprechende Eintrag aus der Konfigurationsdatei `fs-server.conf` entfernt.

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



- Zu jeder Web-Applikation 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 Kapitel 7.4.18 Seite 327.

7.3.13.3 Internen Webserver für eine Web-Applikation konfigurieren

Die Steuerung für den internen Webserver Jetty ist standardmäßig vorhanden und kann nicht verändert werden. Nach der Installation von FirstSpirit ist der interne Webserver für alle FirstSpirit Web-Applikationen aktiviert.

Wurde ein anderer Webserver für eine FirstSpirit Web-Applikation aktiviert, kann die Konfiguration über die folgenden Schritte wieder auf den internen Webserver zurückgesetzt werden:

1. In der Combobox den Eintrag "internalJetty" auswählen.
2. Der Button "Installieren" wird aktiv. Mit einem Klick auf den Button kann die Webapplikation auf dem internen Webserver installiert werden.
3. Nach der Installation wird der Button "Aktivieren" aktiv. Mit einem Klick auf den Button wird die Konfiguration der Web-Applikation auf den internen Webserver umgestellt, der nun als aktiver Webserver für die Anwendung angezeigt wird. Die Verweise auf die Web-Applikation (z. B. auf der Startseite) werden dabei



automatisch auf den neuen Webserver umgestellt.

4. Alle Änderungen der Konfiguration müssen mit einem Klick auf "OK" bestätigt und gespeichert werden.

7.3.13.4 Generischen Webserver für eine Web-Applikation konfigurieren

Die Auswahl eines generischen Webserver ist nur möglich, wenn eine entsprechende Webserver-Instanz zuvor auf dem Server hinzugefügt wurde (siehe Kapitel 7.3.12.1 Seite 263). Die Steuerung für den generischen Webserver ist nicht standardmäßig vorhanden, kann aber über Skripte realisiert werden (siehe Kapitel 7.3.12.2 Seite 263). Sind diese Skripte nicht vorhanden, ist das Vorgehen identisch zum externen Webserver (siehe Kapitel 4).

Soll für eine FirstSpirit Web-Applikation ein generischer Webserver (z. B. Tomcat) aktiviert werden, müssen die folgenden Schritte ausgeführt werden:

1. In der Combobox den Eintrag für den gewünschten generischen Webserver auswählen.
2. Der Button "Installieren" wird aktiv. Sofern die entsprechende Funktionalität über ein Skript zur Verfügung gestellt wurde, kann mit einem Klick auf den Button die Webapplikation auf dem generischen Webserver installiert werden.
3. Nach der Installation wird der Button "Aktivieren" aktiv. Mit einem Klick auf den Button wird die Konfiguration der Web-Applikation auf den generischen Webserver umgestellt, der nun als aktiver Webserver für die Anwendung angezeigt wird. Die Verweise auf die Web-Applikation (z. B. auf der Startseite) werden dabei automatisch auf den neuen Webserver umgestellt. Dazu wird der im unter "Webserver URL" eingetragene URL verwendet (siehe Abbildung 7-53).
4. Alle Änderungen der Konfiguration müssen mit einem Klick auf "Ok" bestätigt und gespeichert werden.

7.3.13.5 Externen Webserver für eine Web-Applikation konfigurieren

Die Auswahl eines externen Webserver ist nur möglich, wenn eine entsprechende Webserver-Instanz zuvor auf dem Server hinzugefügt wurde (siehe Kapitel 7.3.12.3 Seite 265). Die Steuerung für einen externen Webserver wird über FirstSpirit nicht unterstützt und muss manuell erfolgen (siehe Kapitel 7.3.12.4 Seite 266).

Soll für eine FirstSpirit Web-Applikation ein externer Webserver aktiviert werden, müssen die folgenden Schritte ausgeführt werden:

1. In der Combobox den Eintrag für den gewünschten externen Webserver auswählen.



2. Der Button "Download" wird aktiv. Mit einem Klick auf den Button kann eine WAR-Datei der Anwendung heruntergeladen werden.
3. Die WAR-Datei muss manuell auf dem externen Webserver installiert werden. Die Installation erfolgt entweder manuell über die Verwaltungsoberfläche des externen Webservers oder automatisch aus dem Dateisystem des Webservers heraus. Als Start-URL sollte ein eigener virtueller Server erstellt werden, so dass die FirstSpirit-Startseite in der Webanwendung `/fs5root` als Root-Application definiert werden kann, und z. B. direkt über <http://fs5.yourdomain.net> erreicht werden kann. Ein URL mit Präfix wie <http://appserver.yourdomain.net/fs5> ist aber ebenso möglich.
Das Installieren der WAR-Datei muss nach jeder Aktualisierung des FirstSpirit-Servers wiederholt werden!
4. Nach der Installation kann mit einem Klick auf den Button "Aktivieren" die Konfiguration der Web-Applikation auf den externen Webserver umgestellt werden. Der externe Webserver wird nun als aktiver Webserver für die Anwendung angezeigt. Die Verweise auf die Web-Applikation (z. B. auf der Startseite) werden dabei automatisch auf den neuen Webserver umgestellt. Dazu wird der im unter "Webserver URL" eingetragene URL verwendet (siehe Abbildung 7-55).
5. Alle Änderungen der Konfiguration müssen mit einem Klick auf "OK" bestätigt und gespeichert werden.

7.3.13.6 Tomcat-Webserver für eine Web-Applikation konfigurieren

Die Auswahl eines Tomcat-Webservers ist nur möglich, wenn eine entsprechende Webserver-Instanz zuvor auf dem Server hinzugefügt wurde (siehe Kapitel 7.3.12.5 Seite 267). Die Steuerung für einen Tomcat-Webserver wird von FirstSpirit unterstützt und erfolgt automatisch (siehe Kapitel 7.3.12.6 Seite 268). Die Installation auf dem Webserver und die Registrierung der Web-Komponenten erfolgt durch FirstSpirit.

Soll für einen projektspezifischen Webbereich ein Tomcat-Webserver aktiviert werden, müssen die folgenden Schritte ausgeführt werden:

1. In der Combobox den Eintrag für den gewünschten Tomcat-Webserver auswählen.
2. Der Button "Installieren" wird aktiv. Mit einem Klick auf den Button wird die WAR-Datei kopiert und vom Webserver automatisch ausgepackt, bevor die Web-Komponenten registriert werden.
3. Nach der Installation wird der Button "Aktivieren" aktiv. Mit einem Klick auf diesen Button wird die Konfiguration des Webbereichs auf den Tomcat-Webserver umgestellt, der nun als aktiver Webserver für diesen Bereich



angezeigt wird.

- Alle Änderungen der Konfiguration müssen mit einem Klick auf "OK" bestätigt und gespeichert werden.



Wurde die fs-server.jar-Datei aktualisiert, müssen die zuvor beschriebenen Schritte erneut für alle Web-Applikationen in allen Projekten, für die ein Webserver des Typs "Tomcat" ausgewählt wurde, erneut manuell ausgeführt werden. Es erfolgt kein automatisches Update.

7.3.14 Clustering

Als Clustering ist in diesem Fall die Lastverteilung bei der Generierung von Projekten auf weitere FirstSpirit-Server gemeint ("horizontale Skalierbarkeit").

Zum Konzept und Architekturbeschreibung siehe Kapitel 7.6 Seite 402.

Die konfigurierten FirstSpirit-Server innerhalb des Cluster werden in der FirstSpirit Server- und Projektkonfiguration im Bereich "Clustering" innerhalb der Servereigenschaften aufgelistet. Unter Windows dient dieser Bereich auch zur erstmaligen Installation des Clusters. Unter Unix wird der Cluster nur über die Konfigurationsdateien des FirstSpirit-Servers installiert und konfiguriert.

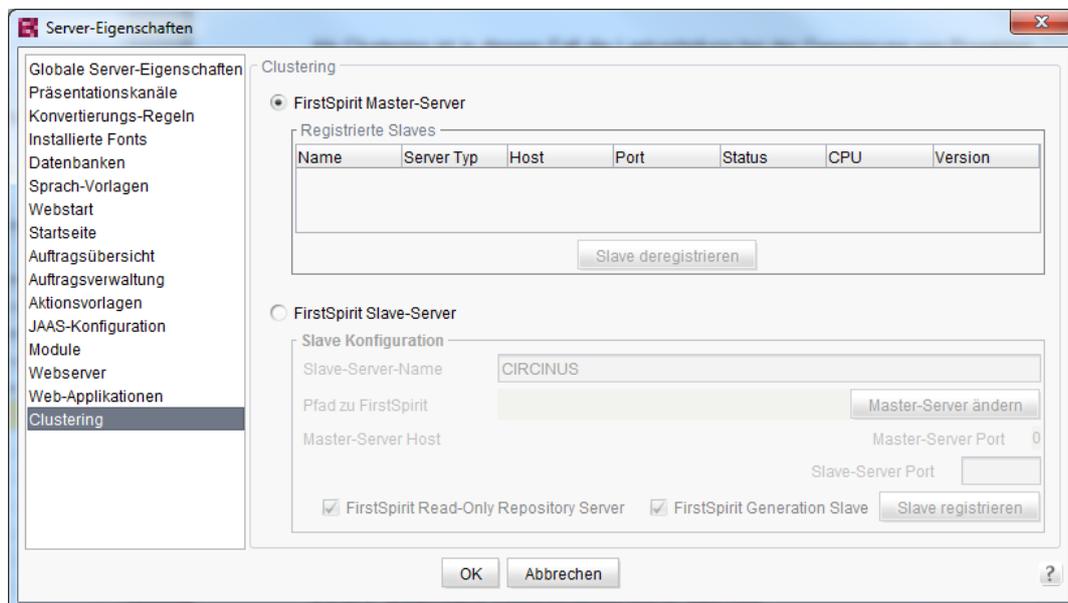


Abbildung 7-61: Servereigenschaften – Clustering



- Konfiguration des Master-Servers unter Unix: Kapitel 7.3.14.1
- Konfiguration des Slave-Servers unter Unix: Kapitel 7.3.14.2
- Konfiguration des Master-Servers unter Windows: Kapitel 7.3.14.3
- Konfiguration des Slave-Servers unter Windows: Kapitel 7.3.14.4

7.3.14.1 Konfiguration FirstSpirit Master-Server unter Unix

Zunächst wird ein FirstSpirit-Server wie im Installationshandbuch beschrieben, installiert. Für den Installationspfad wird als Beispiel `/opt/firstspirit5` angenommen und als Hostname `fs5host`. Nach der Installation wird der FirstSpirit-Server heruntergefahren, um durch die nachfolgend beschriebene Konfigurationsänderungen den FirstSpirit-Server als Master-Server zu definieren und Daten an die FirstSpirit Slave-Server senden zu können.

Der Master-Server enthält den gesamten Funktionsumfang eines normalen FirstSpirit-Servers und sollte bezüglich der Hardware ausfallsicher ausgelegt sein, da alle FirstSpirit-Clients den Master-Server als Endpunkt der Netzverbindungen verwenden. Die Slave-Server alleine sind nicht eigenständig benutzbar. Umgekehrt sind aber alle FirstSpirit-Funktionen auch bei ausgefallenen Slave-Servern alleine durch den Master-Server ohne Konfigurationsänderungen im Betrieb verfügbar. Die Generierungsaufträge werden dann automatisch auf dem Master, statt auf dem Slave ausgeführt, was nur zu einer höheren Auslastung des Masters führt.

In der Datei `/opt/firstspirit5/conf/fs-server.conf` ist folgende Zeile hinzuzufügen:

```
HOST=fs5host
```

Um Daten an die Slave-Server zu übertragen und von den Slave-Servern aus Log-Dateien zum Master-Server zu schreiben, ist ein verteiltes Dateisystem notwendig. Unter Unix bietet sich hierzu NFS an. Der NFS-Server sollte auf demselben System wie der FirstSpirit Master-Server betrieben werden, so dass der Master-Server lokal in das Dateisystem schreiben kann. In diesem Beispiel wird Solaris verwendet. Für andere Unix-Systeme kann die NFS-Konfiguration sinngemäß übernommen werden. Die Unterscheidung zwischen Verzeichnissen mit `readonly` und `read/write`-Zugriff dient dazu, die Betriebssicherheit und den Datendurchsatz zu erhöhen, indem nur die Log-Dateien beschreibbar über NFS verteilt werden. Die zentralen Repository-Daten des Master-Servers werden nur im Lesezugriff verteilt.

In der Datei `/etc/dfs/dfstab` sind folgende Zeilen hinzuzufügen, um einige ausgewählte Verzeichnisse des Master-Servers über NFS freizugeben:



```
share -F nfs -o ro /opt/firstspirit5/bin
share -F nfs -o ro /opt/firstspirit5/conf
share -F nfs -o ro /opt/firstspirit5/server
share -F nfs -o ro /opt/firstspirit5/shared
share -F nfs -o ro /opt/firstspirit5/data/projects
share -F nfs -o rw /opt/firstspirit5/data/schedule
share -F nfs -o rw /opt/firstspirit5/web/fs5staging
share -F nfs -o rw /opt/firstspirit5/log
share -F nfs -o rw /opt/firstspirit5/archive
share -F nfs -o rw /opt/firstspirit5/backup
share -F nfs -o rw /opt/firstspirit5/export
```

Falls erhöhte Sicherheitsanforderungen bestehen, sollte die NFS-Freigabe mittels der vom Betriebssystem unterstützten Authentifizierungsverfahren (IP-Adresse, Kerberos, ...) auf den exklusiven Zugriff der FirstSpirit-Slaves eingeschränkt werden.

Der NFS-Server muss nun über die geänderte Konfiguration informiert werden, z. B. über `shareall`.

Der FirstSpirit Master-Server kann nun über `fs5 start` (bzw. `svcadm enable fs5`) gestartet werden. Auf der FirstSpirit-Startseite kann über das FirstSpirit Server-Monitoring im Bereich "Clustering" die Verfügbarkeit und die Auslastung der Slave-Server überwacht werden (siehe Kapitel 8.6.6 Seite 475) .

Folgende Log-Dateien des Masters dienen zur Fehleranalyse bei Problemen während und nach der Konfiguration:

```
/opt/firstspirit5/log/fs-wrapper.log
/opt/firstspirit5/log/fs-server.log
```

7.3.14.2 Konfiguration FirstSpirit Slave-Server unter Unix

Zunächst wird ein FirstSpirit-Server wie im Installationshandbuch beschrieben, installiert. Der automatische Import des Demo-Projektes wird dabei deaktiviert, über Angabe von `FSDEMO=false` (siehe *FirstSpirit Installationsanleitung*). Dabei muss beachtet werden, dass die zur FirstSpirit-Installation verwendete Unix-User-ID dieselbe ist, wie auf dem Master-Server. Ansonsten gestalten sich die Freigabe und die Beschreibbarkeit der gemeinsamen NFS-Verzeichnisse komplizierter. Als Installationspfad wird als Beispiel `/slave/firstspirit5` angenommen und als Hostname `fs5slave`. Nach der Installation wird der FirstSpirit-Server heruntergefahren, um durch die nachfolgend beschriebene Konfigurationsänderungen den FirstSpirit-Server als Slave-Server zu definieren und Daten an die FirstSpirit Master-Server senden zu können.

Um die spätere Aktualisierung auf eine neue FirstSpirit-Version zu vereinfachen, werden einige JAR-Dateien und Start-Skripte direkt über NFS vom Master-Server verwendet. Nach einer Aktualisierung der Master-Server brauchen die Slave-Server



daher nur neu gestartet zu werden, ohne dass Dateien manuell ausgetauscht werden müssen.

Zur Konfiguration werden nun in der Datei `/slave/firstspirit5/conf/fs-wrapper.conf` die bereits vorhandenen Parameter `wrapper.mainclass` und `#include` geändert:

```
wrapper.java.mainclass=de.espirit.firstspirit.server.ClusterHost
#include ../conf/fs-wrapper-license.slave.conf
```

und folgende Parameter hinzufügen:

```
wrapper.java.additional.12=-Dcmsroot=/import/fs5master
wrapper.java.additional.13=-Dnode=Generierungs-Slave
wrapper.java.additional.14=-DinitialPort=1088
```

Die bei `initialPort` angegebene Portnummer definiert den TCP-Port, den der Slave-Server zum Empfang von Steuerdaten des Master-Servers verwendet.

Die bei der FirstSpirit-Standardinstallation automatisch angelegte Datei `/slave/firstspirit5/conf/fs-server.conf` wird für den Slave-Server nicht benötigt und sollte gelöscht werden:

```
rm /slave/firstspirit5/conf/fs-server.conf
```

Als Beispiel wird hier die Konfiguration des NFS-Clients unter Solaris gezeigt. Unter anderen Unix-Systemen kann diese Konfiguration sinngemäß übernommen werden.

Zunächst werden die Mountpoints erstellt auf denen die Master-Dateien bereitgestellt werden:

```
cd /slave/firstspirit5
rm -r bin server shared
mkdir bin server shared
mkdir -p /import/fs5master/conf
mkdir -p /import/fs5master/data/projects
mkdir -p /import/fs5master/data/schedule
mkdir -p /import/fs5master/web/fs5staging
mkdir -p /import/fs5master/log
mkdir -p /import/fs5master/archive
mkdir -p /import/fs5master/backup
mkdir -p /import/fs5master/export
```

In der Datei `/etc/vfstab` sind nun folgende Einträge notwendig, um die vom Master-Server bereitgestellten Verzeichnisse zu verwenden:

```
fs5host:/opt/firstspirit5/bin - /slave/firstspirit5/bin nfs - yes ro,hard,intr
fs5host:/opt/firstspirit5/server - /slave/firstspirit5/server nfs - yes ro,hard,intr
fs5host:/opt/firstspirit5/shared - /slave/firstspirit5/shared nfs - yes ro,hard,intr
fs5host:/opt/firstspirit5/conf - /import/fs5master/conf nfs - yes ro,hard,intr
fs5host:/opt/firstspirit5/data/projects - /import/fs5master/data/projects nfs - yes ro,hard,intr
fs5host:/opt/firstspirit5/data/schedule - /import/fs5master/data/schedule nfs - yes rw,hard,intr
fs5host:/opt/firstspirit5/web/fs5staging - /import/fs5master/web/fs5staging nfs - yes rw,hard,intr
fs5host:/opt/firstspirit5/log - /import/fs5master/log nfs - yes rw,hard,intr
fs5host:/opt/firstspirit5/archive - /import/fs5master/archive nfs - yes rw,hard,intr
fs5host:/opt/firstspirit5/backup - /import/fs5master/backup nfs - yes rw,hard,intr
fs5host:/opt/firstspirit5/export - /import/fs5master/export nfs - yes rw,hard,intr
```



Der NFS-Client muss nun über die geänderte Konfiguration informiert werden, z. B. über `mount -F nfs -a`.

Der FirstSpirit Slave-Server kann nun über `fs5 start` (bzw. `svcadm enable fs5`) gestartet werden. Auf der FirstSpirit-Startseite kann über das FirstSpirit Server-Monitoring im Bereich "Clustering" die Verfügbarkeit und die Auslastung der Slave-Server überwacht werden (siehe Kapitel 8.6.6 Seite 475) .

Folgende Log-Dateien des Slaves dienen zur Fehleranalyse bei Problemen während und nach der Konfiguration:

```
/slave/firstspirit5/log/fs-wrapper.log
```

```
/import/fs5master/log/fs5slave/fs-server.log
```



Damit ein Slave-Server bei der Projekt-Generierung genutzt wird, muss in der jeweiligen Auftragskonfiguration unter Eigenschaften "Auf Cluster ausführen" aktiviert sein (siehe Kapitel 7.6.4 Seite 406).

7.3.14.3 Konfiguration Master-Server unter Windows

Ist der Radiobutton "FirstSpirit Master-Server" ausgewählt (vgl. Abbildung 7-61), handelt es sich um den FirstSpirit Master-Server. Dieser verwaltet zentral alle FirstSpirit-Projekte und verteilt die Aufgaben, soweit möglich, auf andere FirstSpirit-Server. Die verfügbaren Server (z. B. Generation-Slaves) werden im Bereich "Registrierte Slaves" angezeigt (siehe Abbildung 7-62):

Name	Server Typ	Host	Port	Status	CPU	Version
cluster1	SLAVE	myServer	5051	IDLE	0%	FirstSpirit Ver

Abbildung 7-62: Konfiguration als FirstSpirit Master-Server

Mit einem Klick auf den Button "Slave deregistrieren" kann ein "Registrierter Slave" wieder aus der Auswahl entfernt werden.



7.3.14.4 Konfiguration Slave-Server unter Windows

Im unteren Dialogbereich kann der FirstSpirit-Server als so genannter "Slave-Server" definiert werden. Dazu muss im Bereich der "Slave Konfiguration" ein FirstSpirit-Master-Server bekannt gegeben werden. Nach der erfolgreichen Konfiguration ist der FirstSpirit-Server als Mitglied eines Clusterverbund verfügbar.

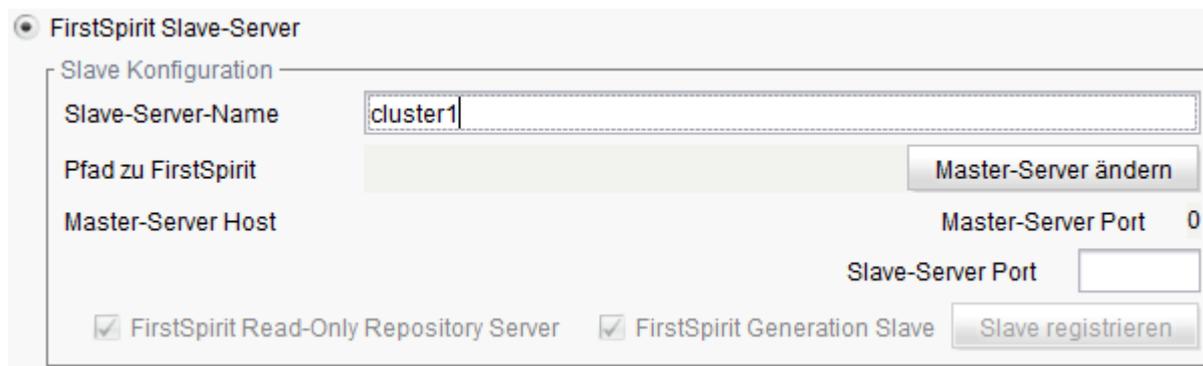


Abbildung 7-63: Konfiguration als FirstSpirit-Slave-Server

Slave Server Name: Name des FirstSpirit-Slaves.

Pfad zu FirstSpirit: Mithilfe des Buttons "Master-Server ändern" kann der Pfad zum FirstSpirit-Master-Server angegeben werden. Nach einem Klick auf den Button öffnet sich ein Dialog, in den der Pfad manuell eingegeben werden kann:

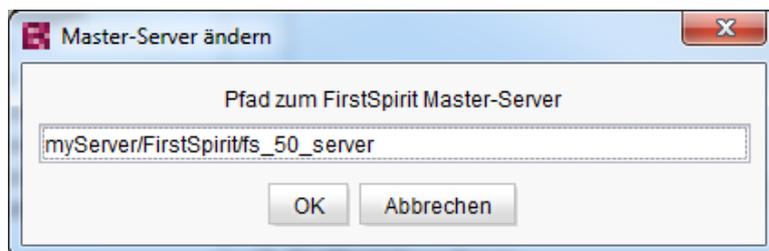


Abbildung 7-64: Pfad zum FirstSpirit-Master-Server

Nach einem Klick auf "OK" wird geprüft, ob ein weiterer FirstSpirit-Server installiert ist. Dieser FirstSpirit-Server (Master-Server) darf nicht dem Server entsprechen, der aktuell konfiguriert wird ("Slave Server"). Konnte ein weiterer FirstSpirit-Server identifiziert werden, werden die Parameter "Master-Server Host" und "Master-Server Port" automatisch gefüllt. Dazu werden die Parameter der Konfigurationsdatei `fs-server.conf` (des FirstSpirit-Master-Servers) verwendet.

Der Dialog für einen korrekt konfigurierten Slave-Server sieht folgendermaßen aus:



● FirstSpirit Slave-Server

Slave Konfiguration

Slave-Server-Name

Pfad zu FirstSpirit

Master-Server Host Master-Server Port

Slave-Server Port

FirstSpirit Read-Only Repository Server FirstSpirit Generation Slave

Abbildung 7-65: Konfiguration als FirstSpirit-Slave-Server

Über den Button "Master-Server ändern" kann der hier eingestellte Master-Server später wieder geändert werden.

Master-Server Host: Hostname des FirstSpirit-Master-Servers. Das Feld wird automatisch nach Angabe eines gültigen Pfads zum FirstSpirit-Server gefüllt.

Master-Server Port: Portnummer des FirstSpirit-Master-Servers. Das Feld wird automatisch nach Angabe eines gültigen Pfads zum FirstSpirit-Server gefüllt.

FirstSpirit Read-Only Repository Server / FirstSpirit Generation Slave: Die Checkboxes sind automatisch aktiviert. Der FirstSpirit-Server wird als Generierungsserver zum Clusterverbund hinzugefügt. Ein Generierungsserver enthält einen RORS³⁰. Die Checkbox "FirstSpirit Read-Only Repository Server" wird daher ebenfalls automatisch aktiviert.

Slave-Server Port: Hier muss ein freier Port für den Slave-Server angegeben werden.

Mit einem Klick auf den Button "Slave registrieren" öffnet sich eine Sicherheitsabfrage:

³⁰ ReadOnly-Repository-Server (siehe Kapitel 7.6.1 Seite 393)



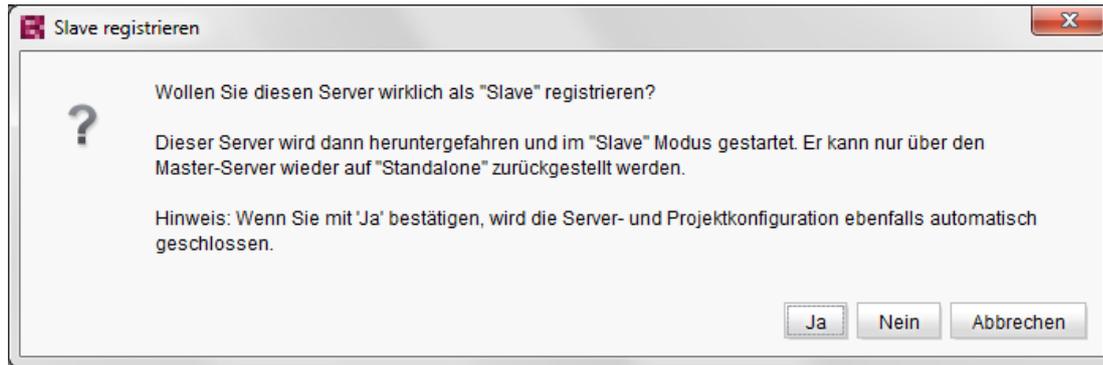


Abbildung 7-66: Slave registrieren

Wird der Dialog mit "Ja" bestätigt, werden die Konfigurationseinstellungen in die entsprechenden Konfigurationsdateien übernommen (fs-server.conf, fs-wrapper.conf). Anschließend wird der Server heruntergefahren und im "Slave" Modus gestartet. Die Anwendung zur Server- und Projektkonfiguration des neuen Slave-Servers wird ebenfalls automatisch geschlossen.

Ein bereits registrierter Slave kann nur über die Servereigenschaften des FirstSpirit-Master-Servers wieder deregistriert werden.

7.4 Projekteigenschaften

Die Projekteigenschaften können über einen Doppelklick auf das gewünschte Projekt in der Projektliste aufgerufen werden. Weitere Möglichkeiten die Projekteigenschaften aufzurufen sind, über den Menüpunkt "Projekt"/"Eigenschaften" in der Menüleiste (siehe Kapitel 7.2.3.7 Seite 219) oder über den Button "Eigenschaften ändern".

Ein Klick auf den Button "Eigenschaften ändern" öffnet die Projekteigenschaften des (über die Projektauswahlliste) markierten Projekts.

Die Projektauswahlliste zeigt nur Projekte an, für die der angemeldete Benutzer die notwendigen Berechtigungen besitzt. Ein Serveradministrator hat selbstverständlich Zugriff auf sämtliche Projekte des Servers, ein Projektadministrator nur auf Projekte, denen er explizit als Administrator zugeordnet wurde (weitere Informationen siehe Kapitel 7.1 Seite 205 und Kapitel 7.4.2 Seite 285).

Nach dem Aufruf erscheint das Dialogfenster "Projekt bearbeiten" (siehe Abbildung 7-68). Über die Menüeinträge im linken Fensterbereich können die Eigenschaften für das selektierte Projekt bearbeitet werden. Auf der rechten Fensterseite können zu jedem Menüeintrag die zugehörigen Einstellungen vorgenommen werden.



Die Projekteinstellungen eines Projektes können immer nur von einem Benutzer gleichzeitig bearbeitet werden. Versucht ein zweiter Benutzer, den Bereich Projekteinstellungen für ein bestimmtes Projekt zu öffnen, erscheint eine Fehlermeldung:



Abbildung 7-67: Fehlermeldung beim Versuch parallelen Bearbeitens



7.4.1 Projekt

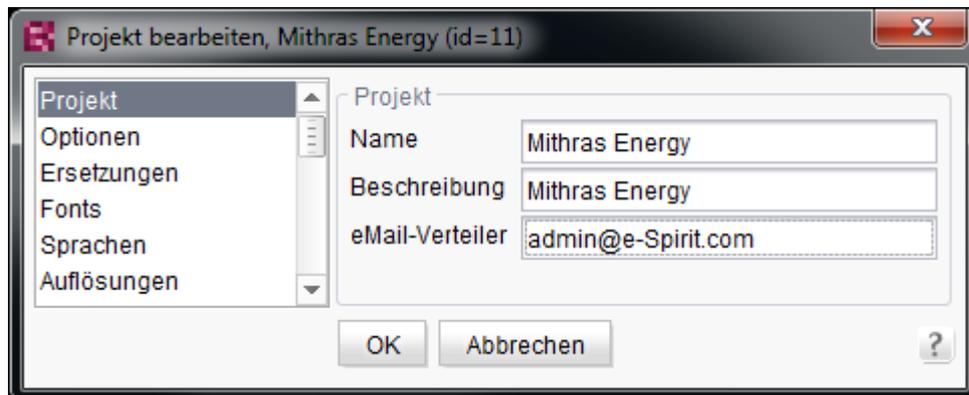


Abbildung 7-68: Projekteigenschaften – Projekt

Name: In diesem Feld wird der eindeutige Projektname angezeigt, der bei Bedarf auch geändert werden kann. Wird der Projektname geändert, bleibt der Button "OK" solange inaktiv, bis ein eindeutiger Name vergeben wurde.

Beschreibung: In diesem Feld wird die Projektbeschreibung angezeigt, die bei Bedarf auch geändert werden kann. Die Beschreibung muss nicht eindeutig sein, es handelt sich aber auch hier um ein Pflichtfeld.

eMail-Verteiler: Nach jeder Generierung wird eine E-Mail mit Statusinformationen an die hier eingetragenen Adressen verschickt. Mehrere Adressen müssen durch ein Semikolon getrennt werden.



7.4.2 Optionen

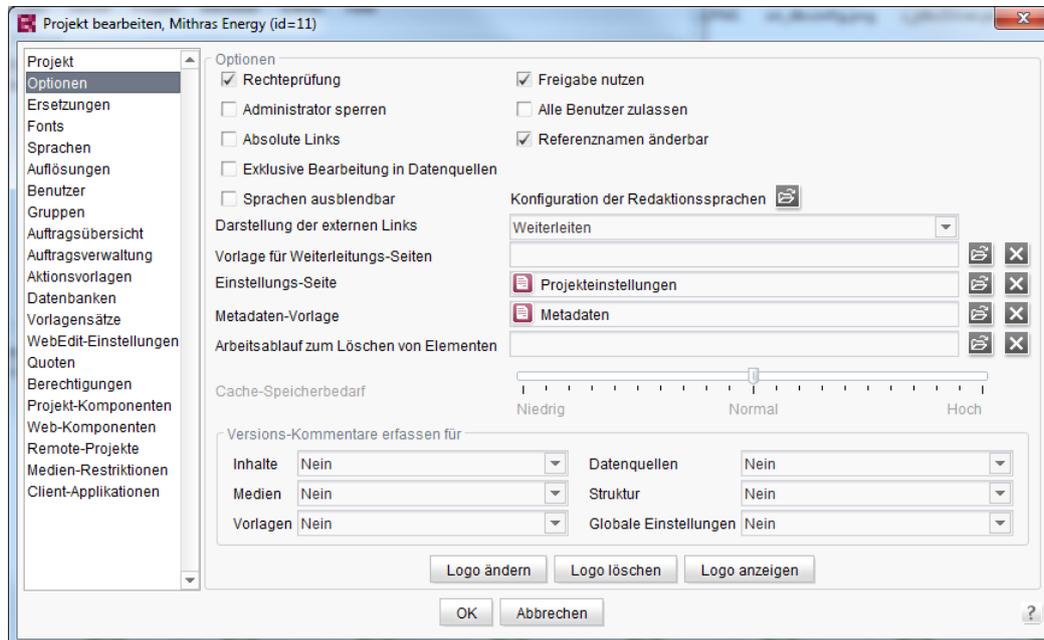


Abbildung 7-69: Projekteigenschaften – Optionen

Rechteprüfung: Mit dieser Option kann definiert werden, ob eine Rechteprüfung für das Projekt verwendet werden soll. Ist die Checkbox *deaktiviert*, erhalten die Benutzer automatisch volle Zugriffsrechte auf das Projekt. Sie können damit auf Menüfunktionen oder Objekte zugreifen, auf die normalerweise nur Projektadministratoren Zugriff erhalten. Ist die Checkbox *aktiviert*, werden die Rechte der Benutzer ausgewertet. Ein Benutzer, der kein Projektadministrator ist, kann beispielsweise kein Projekt generieren. Diese Option ist bei der Neuanlage eines Projekts standardmäßig aktiviert.

Freigabe nutzen: Ist die Checkbox *aktiviert*, unterscheidet FirstSpirit zwischen dem freigegebenen und dem aktuellen Projektstand. Sobald eine Änderung im Projekt vorgenommen wird, muss diese durch einen Benutzer mit entsprechenden Rechten (z. B. durch den "Chefredakteur") freigegeben werden. Ein nicht freigegebener Projektstand, beispielsweise einer Seite aus der Inhalte-Verwaltung, wird bei der nächsten Generierung nicht übernommen. (Stattdessen wird der letzte freigegebene Stand der Seite berücksichtigt.) Ist die Checkbox *deaktiviert*, wird jede Änderung am Projekt, automatisch in den Freigabestand übernommen ("Autorelease") und damit sofort wirksam.

Der WebClient 5 unterstützt keine Projekte, die nicht mit Freigaben arbeiten.



Wird die Option *neu aktiviert*, erfolgt eine Abfrage "Soll eine Freigabeversion von allen Verwaltungen erstellt werden?" Wird "Ja" gewählt, wird eine Freigabeversion aller Verwaltungen, mit Ausnahme der Vorlagen-Verwaltung, erzeugt. Dieser Vorgang kann einige Zeit in Anspruch nehmen. Wird "Nein" gewählt, wird die Freigabeoption aktiviert, ohne eine Freigabeversion der Verwaltungen zu erstellen. Diese Option ist bei der Neuanlage eines Projekts standardmäßig aktiviert.



Eine Deaktivierung dieser Option ist nur für kleinere Projekte sinnvoll und hat negative Auswirkungen auf die Gesamtleistung des Systems, da bei jeder Änderung eine automatische Freigabe erfolgt.



Um Multiviews Pfade erzeugen zu können, kann in einer Aktion "Generierung ausführen" die Option "Multiview URLs" ausgewählt werden (siehe Kapitel 7.5.10.2 Seite 385, Option "Pfad erzeugen").

Administrator sperren: Ist die Checkbox *aktiviert*, wird der Zugriff des Serveradministrators auf dieses Projekt gesperrt (siehe Kapitel 7.1 Seite 205). Der Serveradministrator bekommt das Projekt nicht mehr in der Projektauswahlliste des JavaClients oder WebClients angezeigt und kann es nicht mehr bearbeiten. Die Aufgaben eines Serveradministrators können jedoch weiterhin wahrgenommen werden:

- neue Projekte anlegen / exportieren / löschen
- Benutzer anlegen
- die Einstellungen aller Projekte verändern
- Projektadministratoren definieren
- Editor- und Funktions-Komponenten installieren und deinstallieren
- spezielle Operationen des Serverbetriebs ausführen

Wenn diese Option *deaktiviert* ist, dann kann der Serveradministrator außerdem die Aufgaben eines Projektadministrators wahrnehmen, und das Projekt über die Projektauswahlliste im JavaClient und im WebClient auswählen.

Alle Benutzer zulassen: Ist die Checkbox *aktiviert*, erhalten alle auf dem Server bekannten Benutzer automatisch Zugriff auf das Projekt. Das bedeutet, Zugriff auf das Projekt erhalten auch Benutzer, die nicht explizit als Benutzer dem Projekt hinzugefügt wurden und damit nicht unter Projekteigenschaften "Benutzer" (siehe Kapitel 7.4.7 Seite 301) aufgeführt werden, sondern alle Benutzer, die sich am Server authentifizieren konnten (z. B. über LDAP, SSO). Diese Benutzer gehören zu



keiner internen Projektgruppe und haben, sofern sie nicht Mitglieder einer externen Projektgruppe sind, nur eingeschränkte Rechte im Projekt. Diese Rechte können über die Zugriffsrechte der Projektgruppe "Everyone" konfiguriert werden. Der Zugriff auf das Projekt durch "externe Benutzer" kann eingeschränkt werden, indem statt "alle Benutzer zulassen", Mitglieder einer externen Gruppe Zugriff auf das Projekt erhalten. Auch diese müssen dem Projekt nicht mehr explizit als Benutzer hinzugefügt werden (weitere Informationen zu externen Gruppen siehe Kapitel 1.1.1 Seite 302). Ist die Checkbox *deaktiviert*, müssen Benutzer, die Zugriff auf das Projekt erhalten sollen, explizit dem Projekt hinzugefügt werden (siehe Kapitel 7.4.7 Seite 301).

Absolute Links: Statt der relativen Pfade für Seitenreferenzen lassen sich auch absolute Pfade generieren. Mit dieser Projekteigenschaft werden alle Referenzen auf Seiten beeinflusst. Der Präfix, der benötigt wird, um die absoluten Links zu vervollständigen, wird aus dem Feld "Präfix für absolute Pfade" aus dem Generierungsauftrag gelesen (siehe Kapitel 7.5.10.2 Seite 385). Diese Option bezieht sich nicht auf Medienreferenzen. Sollen Medienreferenzen mit absoluten Pfaden dargestellt werden, muss die Referenz mit dem Attribut `abs` versehen werden. Bei der Einstellung `abs:2` wird der Präfix für Verweise auf Medien und Seiten nicht verwendet (siehe FirstSpirit Online Dokumentation).

Referenznamen änderbar: Ist diese Checkbox 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, indem die Option "Referenznamen anzeigen" in den Projekteigenschaften, Bereich "Optionen", "Konfiguration der Redaktionssprachen", deaktiviert wird. Dann ist eine Änderung nur über BeanShell (API) möglich.

Exklusive Bearbeitung von Datenquellen: Ist diese Checkbox aktiviert, können Datenquellen nur noch exklusiv durch einen Benutzer bearbeitet werden. Für andere Benutzer ist die Bearbeitung eines Datensatzes in der Datenquelle solange nicht



möglich. Ist die Checkbox deaktiviert, wird Multi-User-Betrieb für Datenquellen angewendet, der jedoch zu Konflikten führen kann, wenn mehrere Redakteure versuchen, denselben Datensatz zu bearbeiten. Die Option ist standardmäßig nicht aktiviert. Wird die Option aktiviert, wirkt sie sich auf alle Datenquellen im betreffenden Projekt aus, sowohl im JavaClient als auch im WebClient.

Zu Auswirkungen im JavaClient siehe auch FirstSpirit Handbuch für Redakteure (JavaClient) Version 5.0, Kapitel "Datenerfassung".

Sprachen ausblendbar: Ist diese Checkbox aktiviert, lassen sich im JavaClient im Menü "Ansicht" über den Unterpunkt "Sichtbare Projektsprachen" einzelne Projektsprachen benutzerspezifisch ausblenden. Dadurch wird erreicht, dass der Redakteur nur die für ihn relevanten Sprachen sehen kann. Blendet der Redakteur eine Sprache aus, wird der entsprechende Sprachreiter nicht mehr angezeigt, die Inhalte für diese Sprache werden jedoch weiterhin geladen.



In den Formularen dürfen keine speicherverhindernden Pflichtfelder (z. B. "allowEmpty") vorhanden sein, weil es sonst im ausgeblendeten Sprachkanal zu Fehlern bei der Validierung der Felder kommen kann. Dadurch können Änderungen in einer Sprache nicht gespeichert werden.



Das Ausblenden einer Sprache kann sich u. U. erst nach einem eustart des JavaClients auswirken. Bei aktiviertem Bearbeitungsmodus wird ein Sprachreiter erst nach dem Verlassen des Bearbeitungsmodus und einer anschließenden Aktualisierung des Elements (<F5>) ausgeblendet.

Konfiguration der Redaktionssprachen: Mit einem Klick auf das Ordner-Icon hinter dem Eintrag "Konfiguration der Redaktionssprachen", öffnet sich ein Konfigurationsdialog:

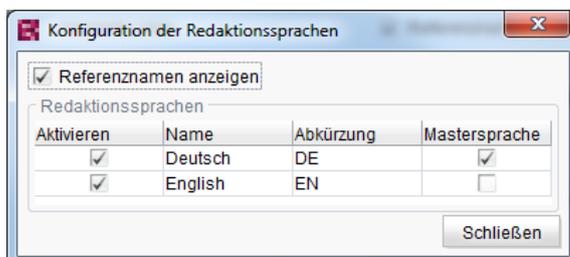


Abbildung 7-70: Redaktionssprachen konfigurieren



Referenznamen anzeigen: Ist die Checkbox aktiviert (Standardeinstellung), können neben den sprachabhängigen Anzeigenamen (vgl. "Aktivieren") zusätzlich die sprachunabhängigen Referenznamen der Objekte angezeigt werden.

Aktivieren: Mit Aktivieren der Checkbox, können Redaktions Sprachen für ein Projekt festgelegt werden. Die "aktivierten" Sprachen können anschließend in der FirstSpirit Redaktionsumgebung als "Bevorzugte Anzeigesprache" eingestellt werden (über das Menü "Extras"). Die Redaktions Sprachen wirken sich auf sprachabhängige Inhalte aus, die vom Vorlagenentwickler, z. B. innerhalb der Seiten- oder Absatzvorlagen, definiert wurden. Dem Redakteur werden die entsprechenden sprachabhängigen Beschriftungen beispielsweise im Formularbereich (Beschriftung der Eingabefelder, Tooltips, Elemente einer Combobox, usw.) angezeigt und wirken sich auch auf die Anzeige der Objekte im Baum aus. Die Redaktions Sprachen sind nicht zu verwechseln mit den Projektsprachen (vgl. Kapitel 7.4.5 Seite 295).

Name / Abkürzung: Name und Kürzel der gewünschten Redaktions Sprache.

Mastersprache: Die aktivierte Checkbox in dieser Spalte kennzeichnet die Projekt-Mastersprache. Diese ist als Redaktions Sprache standardmäßig aktiviert.

Darstellung der externen Links: Die Auswahl ist relevant, wenn in der Struktur-Verwaltung eines Projekts die Verknüpfung "externe Adresse (URL)" verwendet werden soll. Seitenreferenzen können direkt auf eine externe Seite verweisen, in diesem Fall wird entweder eine direkte "Weiterleitung" auf diese Seite ausgeführt (ohne dass die Seite auf dem FirstSpirit Server abgelegt wird) oder eine Weiterleitung ausgeführt und zusätzlich die "Seite generiert" und auf dem Server abgelegt.

Einstellungs-Seite: Hier kann eine Seitenvorlage aus dem Projekt ausgewählt werden, die als Grundlage für die Pflege der globalen Projekteinstellungen dienen soll. Die im Register "Formular" der Seitenvorlage definierten GUI-Elemente werden im Verwaltungsbereich "Globale Einstellungen" innerhalb der "Projekteinstellungen" angezeigt und können mit Inhalten gefüllt werden. Die Inhalte werden im Generierungs- und Preview-Kontext wie Struktur-Variablen behandelt, die auf dem Wurzelknoten der Struktur-Verwaltung definiert wurden. Die Vorbereitung der Daten erfolgt durch die Seitenvorlage, wo zentral die Funktionen aufgerufen werden, die zur Manipulation der Projektkonfigurationsdaten benötigt werden. In der Vorlage berechnete Werte können ggf. per `$CMS_SET(..)$` oder `CMS_HEADER`-Funktion in den Kontext eingefügt und so von jeder beliebigen Vorlage abgefragt werden.



Metadaten-Vorlage: In diesem Feld kann eine Vorlage ausgewählt werden, die für die Metadaten des Projektes verwendet werden soll.



Im Gegensatz zur Auswahl von Seitenvorlagen in der Redaktionsumgebung des JavaClients, ist die Vorlagenauswahl in diesem Bereich (sowohl für die Einstellungsseite als auch für die Metadaten-Vorlage) nicht eingeschränkt. Es werden immer alle verfügbaren Seitenvorlagen zur Auswahl angezeigt, auch Vorlagen, die als "nicht sichtbar" markiert wurden.

Arbeitsablauf zum Löschen von Elementen: Für das Löschen von Elementen im FirstSpirit-JavaClient und im FirstSpirit-WebClient kann ein projektspezifischer Arbeitsablauf erstellt und direkt an die bisherigen Bedienelemente zum Löschen (Buttons der Menüleiste, Kontextmenüeintrag) von Elementen gebunden werden. Statt dem einfachen Löschen eines Objekts, beispielsweise einer Seite, kann über den Arbeitsablauf eine komplexere Löschfunktionalität bereitgestellt werden, beispielsweise das zusätzliche Löschen abhängiger Objekte einer Seite.

Weiterführende Informationen zum Löschen von Elementen über einen Arbeitsablauf siehe "FirstSpirit Handbuch für Entwickler (Grundlagen)".



Abbildung 7-71: Projekteigenschaften – Einstellung des Cache-Speicherbedarfs

Cache-Speicherbedarf: Über den Schieberegler kann der Anteil der zugewiesenen Speichergröße für den Cache des Projekts erhöht (bei hoher Gewichtung) oder verringert (bei niedriger Gewichtung) werden. Der vom CacheSizeManager zugewiesene Speicher ist außerdem noch abhängig vom real benötigten Speicher ("Preferred Size") und dem zur Verfügung stehenden Gesamtspeicher (vgl. Kapitel 9.4 Seite 483).

Standardmäßig ist dieser Schieberegler deaktiviert. Soll er (für alle Projekte auf einem FirstSpirit-Server) aktiviert werden, muss in der Konfigurationsdatei `fs-server.conf` im Serververzeichnis des FirstSpirit-Servers "`~FS/conf/`" der Parameter `repository.sharedCache` eingetragen und auf einen Wert **ungleich 1** gesetzt werden. Der Ausdruck `repository.sharedCache=1` oder das Löschen des Parameters deaktiviert den Regler (siehe Kapitel 4.3.1.11 Seite 54).

Versions-Kommentare erfassen für: Über diese Comboboxen kann für jeden Verwaltungsbereich einzeln festgelegt werden, ob ein Kommentar für Änderungen



an Objekten abgefragt werden soll, wenn der Redakteur den Bearbeitungsmodus verlässt.

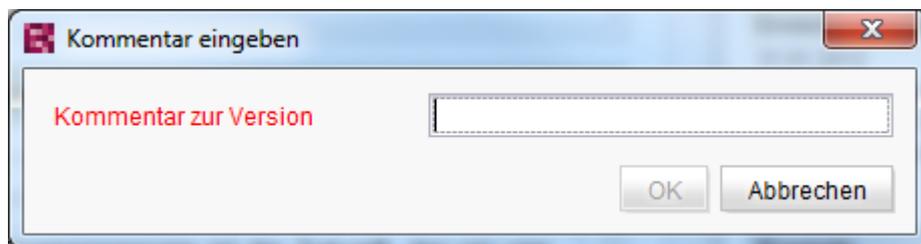


Abbildung 7-72: Versions-Kommentar eingeben

Folgende Optionen sind möglich:

Ja, erzwingen: Wenn diese Option aktiviert ist, wird dem Redakteur die in Abbildung 7-72 dargestellte Kommentarzeile angezeigt. Sie kann erst dann mit **OK** geschlossen werden, wenn der Redakteur einen Text eingetragen hat.

Ja, optional: Wenn diese Option aktiviert ist, wird dem Redakteur die in Abbildung 7-72 dargestellte Kommentarzeile angezeigt. Sie kann auch ohne Eintrag mit **OK** geschlossen werden.

Nein: Wenn diese Option aktiviert ist, kann der Redakteur den Bearbeitungs-Modus verlassen, ohne die Kommentarzeile angezeigt zu bekommen.

Wird ein Kommentar beim Ändern eines Objekts eingegeben, wird dieser anschließend auch in der Versionshistorie des Objekts angezeigt (in Abbildung 7-73 "Das ist ein Kommentar!"):



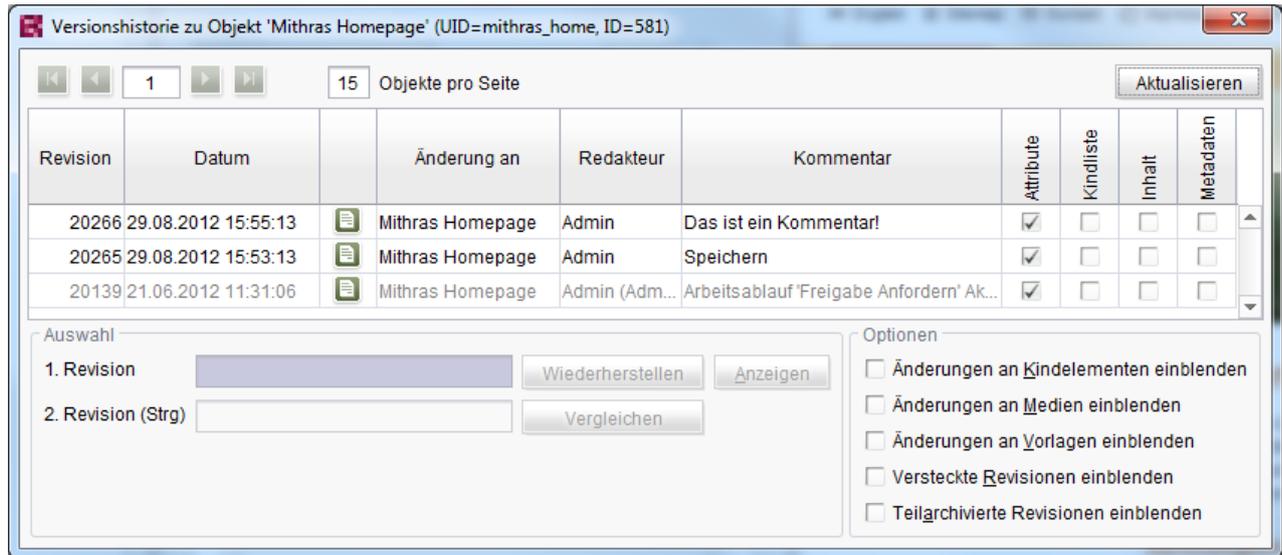


Abbildung 7-73: Änderungskommentar in der Versionshistorie

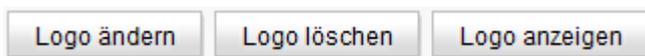


Abbildung 7-74: Projekteigenschaften – Logo bearbeiten

Logo ändern: Hier kann eine Grafik ausgewählt werden, die nach dem Starten des Clients auf der rechten Bildschirmseite (Bearbeitungsbereich) erscheinen soll.

Logo löschen: Eine zuvor definierte Grafik wird gelöscht. Nach dem Starten des Clients erscheint auf der rechten Bildschirmseite (Bearbeitungsbereich) das FirstSpirit Logo.

Logo anzeigen: Es wird eine Vorschau des ausgewählten Logos angezeigt. Ist hier kein Logo vorhanden, wird standardmäßig das FirstSpirit Logo angezeigt.



7.4.3 Ersetzungen

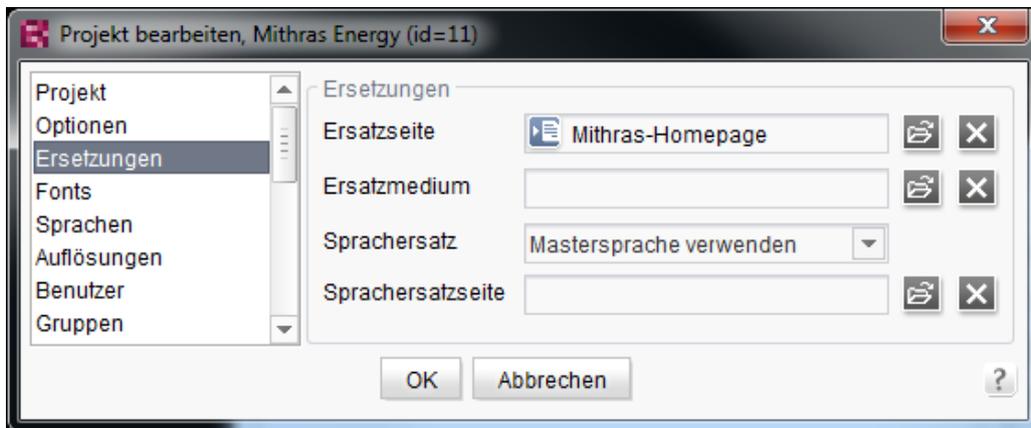


Abbildung 7-75: Projekteigenschaften – Ersetzungen

Ersatzseite: Sollte eine vom Besucher der Website angeforderte Seite einmal nicht vorhanden sein, so kann hier eine alternative Seite zur Anzeige ausgewählt werden. Über das Dateisymbol kann die gewünschte Seite aus der Struktur-Verwaltung ausgewählt werden.

Ersatzmedium: Sollte ein Medium einmal nicht vorhanden sein, so kann hier ein Medium angegeben werden, das alternativ angezeigt wird. Über das Dateisymbol kann das gewünschte Medium aus der Medien-Verwaltung ausgewählt werden.

Sprachersatz: In mehrsprachigen Projekten kann es vorkommen, dass die Übersetzung einiger Seiten noch nicht in jeder Sprache vorliegt. Dies wird in FirstSpirit durch einen entsprechenden Haken auf der Seite in der Inhalte-Verwaltung gekennzeichnet

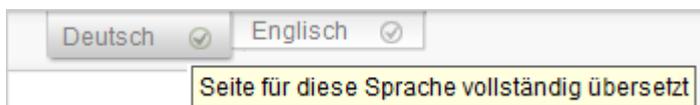


Abbildung 7-76: Sprache als vollständig kennzeichnen

Für diesen Fall kann eine Regel definiert werden, wie mit den entsprechenden Seiten verfahren werden soll. Es stehen folgende Optionen zur Auswahl:

- **Ignorieren:** Die Seite wird einfach in ihrem momentanen Zustand für die Sprache verwendet. Das System geht also davon aus, dass der Redakteur nur vergessen hat, den Haken zu setzen.
- **Mastersprache verwenden:** Die Seite wird statt in der angegebenen in der Mastersprache des Projektes generiert. Da sich bei der Verwendung dieser



Option für einen Besucher der Site bei einem Seitenwechsel plötzlich die Sprache ändern kann, sollte die Verwendung dieser Option gründlich überlegt sein.

- **Ersatzseite verwenden:** Hier wird die oben definierte Ersatzseite verwendet.
- **Spezielle Seite verwenden:** Es wird eine gesonderte Sprachersatzseite (s.u.) angezeigt.

Sprachersatzseite: Wenn die eigentlich angeforderte Seite für eine Projektsprache noch nicht übersetzt wurde und die Sprachersatzregel "Spezielle Seite verwenden" (s.o.) aktiv ist, so kann über das Dateisymbol eine alternative Seite aus der Strukturverwaltung zur Anzeige ausgewählt werden.

7.4.4 Fonts

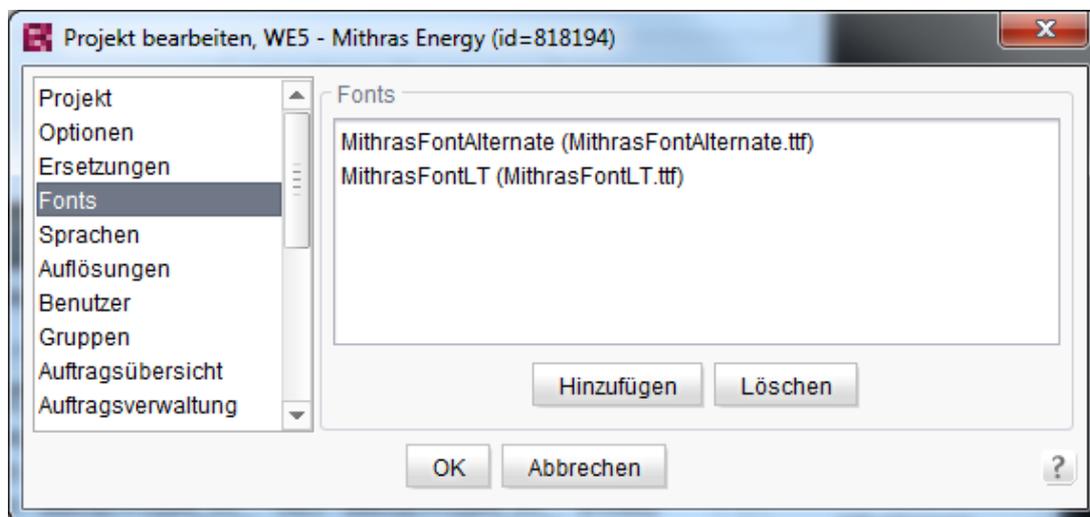


Abbildung 7-77: Projekteigenschaften – Fonts

Hier können aus den Fonts, die auf dem Server installiert sind (siehe Kapitel 7.3.4 Seite 239), über den Button "Hinzufügen" diejenigen ausgewählt werden, die für das aktuelle Projekt verfügbar sein sollen.



7.4.5 Sprachen

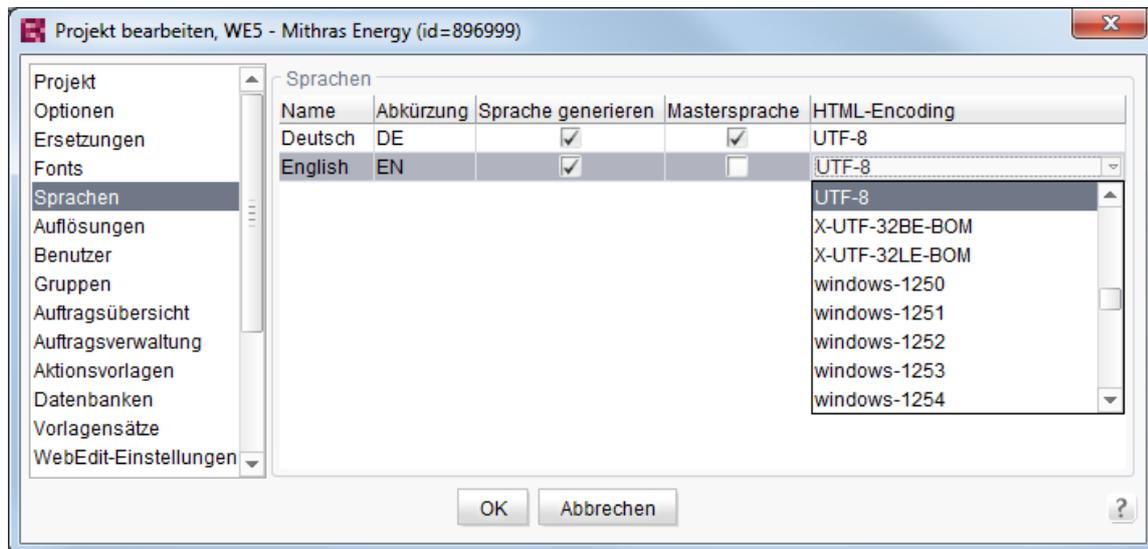


Abbildung 7-78: Projekteigenschaften – Sprachen

Hier sind alle Sprachen aufgelistet, die in dem Projekt verwendet werden. Die Tabelle verfügt über die folgenden Spalten:

Name: Hier wird der Name angezeigt, unter dem die Sprache in FirstSpirit integriert wurde. Neue Sprachen müssen zunächst über die Menüleiste für den Server bereitgestellt werden, bevor sie auf diesem Register für ein bestimmtes Projekt eingefügt werden können (siehe Kapitel 7.3.6 Seite 246).

Abkürzung: Hier wird das Sprachkürzel angezeigt, unter dem man die Sprache in den Clients verwendet.

Sprache generieren: Ist diese Option deaktiviert, dann wird die Sprache bei der Generierung nicht berücksichtigt. Dies ist z. B. sinnvoll, wenn die Inhalte in dieser Sprache noch nicht vollständig sind.

Mastersprache: Es ist immer genau eine Sprache als Mastersprache gekennzeichnet. Ein Ändern der Mastersprache ist durch Anklicken der Option für die entsprechende Sprache möglich.

HTML-Encoding: In dieser Combobox werden 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. Die Voreinstellung für jede neu hinzugefügte Sprache ist ISO-8859-1, was dem westeuropäischen Sprachraum entspricht.



Sollte bei einem Wechsel von einer JDK-Version zu einer anderen das Encoding nicht (mehr) unterstützt werden, wird dies 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.

Mit einem rechten Mausklick im Übersichtsfenster wird das Kontextmenü aufgerufen:

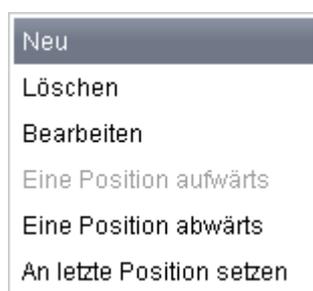


Abbildung 7-79: Kontextmenü (Bereich Sprachen)

Neu: Hiermit kann eine neue Sprache zum Projekt hinzugefügt werden. Es erscheint eine Auswahlliste der Sprachen, die auf diesem Server zur Verfügung stehen.



Wird eine neue Sprache eingefügt, müssen alle Verwaltungen im Client (Inhalte-Verwaltung, Struktur-Verwaltung, Medien-Verwaltung, Datenquellen-Verwaltung) neu freigegeben werden.



Bei einem größeren Projekt mit vielen Medien (>>1000 bzw. mediastore.xml-Datei ~1MB) muss darauf geachtet werden, dass für diese Aktion ausreichend Speicherreserven zur Verfügung stehen!

Sind ein oder mehrere Generierungsaufträge vorhanden, kann gewählt werden, ob die Generierung in dieser neu hinzugefügten Sprache durchgeführt werden soll: "Soll dieser Sprachkanal von allen Aufträgen dieses Projektes generiert werden?" Mit "Ja" werden die Häkchen in Abbildung 7-149 entsprechend gesetzt.



Löschen: Mit dieser Funktion wird die markierte Sprache aus dem Projekt entfernt.



Beim Entfernen einer Sprache können Projekthinhalte verloren gehen. Es ist zwar möglich, eine gelöschte Sprache auf Basis der entsprechenden Sprachvorlage erneut anzulegen. In diesem Fall werden die zuvor gepflegten Inhalte im Projekt im entsprechenden Sprachregister wieder angezeigt. Inhalte, die zum Zeitpunkt des Löschens (der Sprache) noch nicht gespeichert waren, können aber nicht wiederhergestellt werden.

Bearbeiten: Mit dieser Funktion kann die markierte Sprache zum Bearbeiten geöffnet werden. Dabei können alle Eigenschaften der Sprache (Name, Sprache nach ISO-639, Land nach ISO-3166, sprachabhängige Anzeigenamen) geändert werden, mit Ausnahme des Kürzels, das zur eindeutigen Identifizierung der Sprache im Projekt benötigt wird und nicht verändert werden darf.

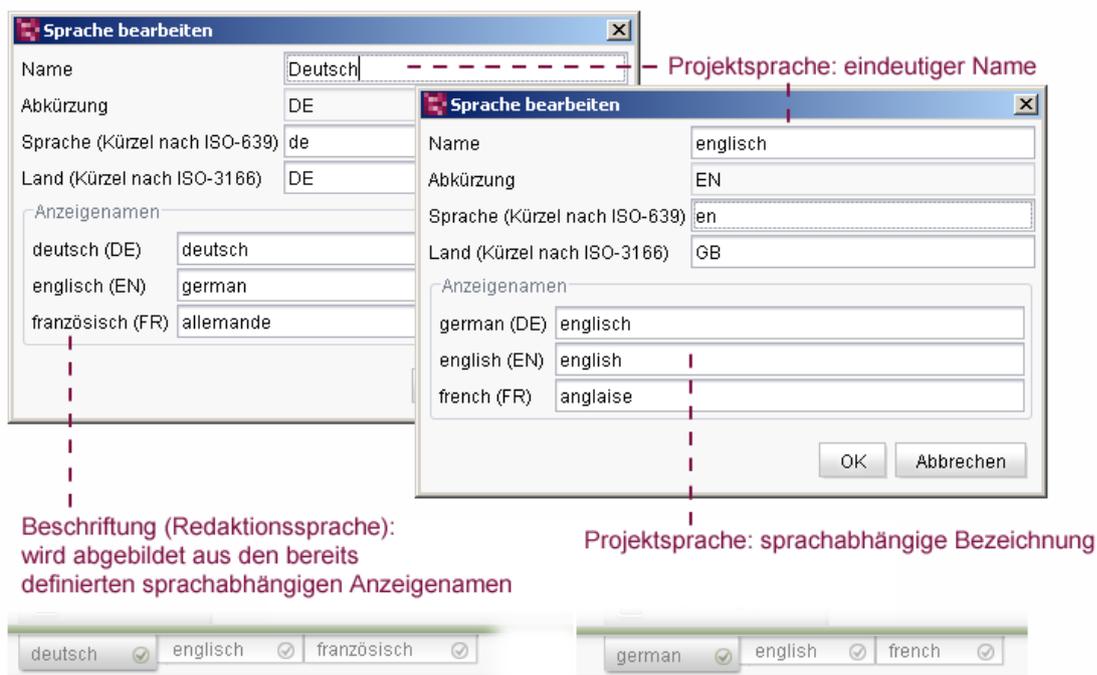


Abbildung 7-80: Sprachen bearbeiten

Für eine Projektsprache können sprachabhängige Anzeigenamen definiert werden. Im Formularebereich "Anzeigenamen" werden dazu die entsprechenden Eingabefelder angezeigt. Sprachabhängige Anzeigenamen können zu allen Redaktionssprachen eines Projekts definiert werden.

Eine Position aufwärts/abwärts: Die definierten Projektsprachen werden im JavaClient in der definierten Reihenfolge angezeigt. Mithilfe des Kontextmenüs kann



die Positionierung der Sprachregister im Projekt angepasst werden. Dazu muss einfach der entsprechende Eintrag selektiert und über das Kontextmenü oder die entsprechenden Buttons schrittweise nach oben oder unten verschoben werden. Dabei gilt: Die Mastersprache bleibt immer an erster Stelle der Reihenfolge.

An letzte Position setzen: Mithilfe dieses Eintrags kann die selektierte Projektsprache an die letzte Position gesetzt werden.

7.4.6 Auflösungen

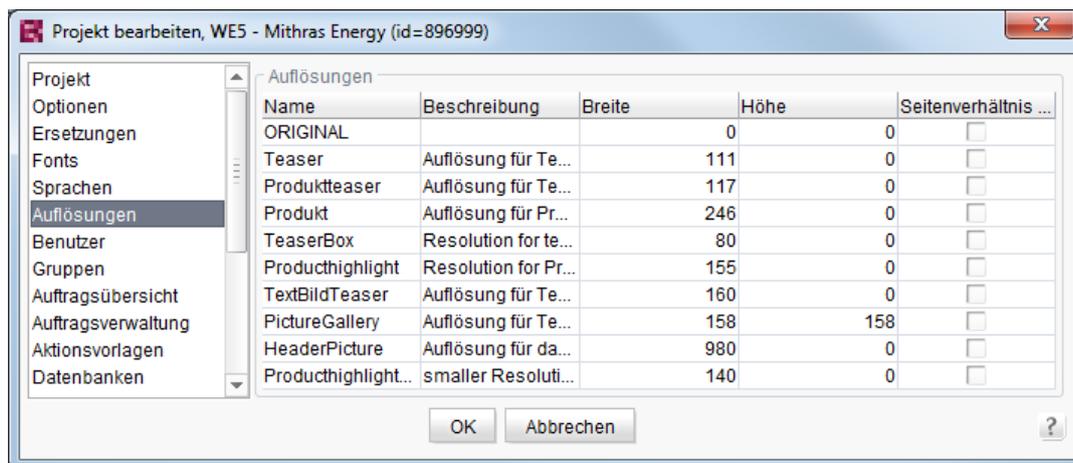


Abbildung 7-81: Projekteigenschaften – Auflösungen

Hier sind alle Auflösungen aufgelistet, die für das Projekt definiert werden. Die Tabelle verfügt über die folgenden Spalten:

Name: Hier wird der eindeutige technische Name angezeigt, der für eine Auflösung vergeben wurde. Dieser Name wird zur Identifizierung im Projekt benötigt und kann nachträglich nicht mehr geändert werden. Zunächst verfügt ein Projekt nur über die Auflösung "ORIGINAL". Diese Auflösung repräsentiert das unveränderte Medium in der Auflösung, in der es in der Medien-Verwaltung eingefügt wurde (symbolisiert durch den Wert "0" sowohl für Breite als auch Höhe). Zwischen Groß- und Kleinschreibung wird bei der Namensvergabe nicht unterschieden.

Beschreibung: Eingabefeld für eine optionale technische Beschreibung zur Auflösung. Diese Information wird nur innerhalb der Projekteigenschaften zur Verfügung gestellt.

Breite / Höhe: Hier wird die Breite bzw. Höhe einer Auflösung in Pixel angezeigt. Eine "0" symbolisiert, dass sich die Breite bzw. Höhe aus dem Seitenverhältnis zur Originalauflösung ergibt.



Seitenverhältnis beibehalten: Über diese Option kann festgelegt werden, ob das Seitenverhältnis des Originalbildes für die jeweilige Auflösung später bei der Ausgabe auf der Webseite berücksichtigt werden soll oder nicht. Ist die Option aktiviert, wird das Seitenverhältnis des Originalbildes beibehalten.



Auflösungen, für die diese Option deaktiviert ist, können auf der Webseite gestaucht ausgegeben werden, wenn das Seitenverhältnis des Originalbildes nicht gleich dem Seitenverhältnis der entsprechenden Auflösung ist. Um dies zu unterbinden, sollte die Option aktiviert werden.

Mit einem rechten Mausklick im Übersichtsfenster wird das Kontextmenü aufgerufen:

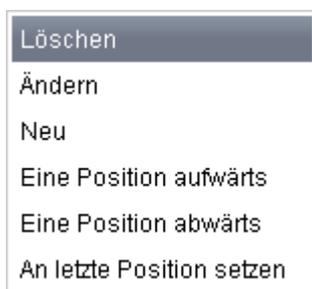


Abbildung 7-82: Kontextmenu (Bereich Auflösungen)

Löschen: Mithilfe dieser Funktion kann die ausgewählte Auflösung aus dem Projekt gelöscht werden.

Ändern: Mit dieser Funktion kann die markierte Auflösung zum Bearbeiten geöffnet werden. Darüber können auch die sprachabhängigen Anzeigenamen und die sprachabhängigen Beschreibungen für eine Auflösung geändert werden. Alle anderen Eigenschaften können nachträglich nicht mehr verändert werden.

Neu: Mithilfe dieser Funktion kann dem Projekt eine neue Auflösung hinzugefügt werden. Durch Aufrufen der Funktion öffnet sich ein Fenster in dem die Einstellungen für die neue Auflösung vorgenommen werden können.

Neben den bekannten Eingabefeldern für den eindeutigen, technischen Namen, die Beschreibung und die Definition der Breite und Höhe einer Auflösung, können sprachabhängige Anzeigenamen und sprachabhängige Beschreibungen für eine Auflösung in allen Redaktionssprachen definiert werden.

Es können keine Auflösungen mit den Namen "preview"/"PREVIEW" oder "original"/"ORIGINAL" angelegt werden, da es sich hierbei um reservierte, interne



Bezeichner handelt.

Wird im FirstSpirit-JavaClient eine "bevorzugte Anzeigesprache" definiert, werden die entsprechenden sprachabhängigen Anzeigenamen der Auflösungen im JavaClient angezeigt.

Nachdem die neue Auflösung für Medien definiert wurde, erscheint in der Medien-Verwaltung eine neue Auflösungszeile.

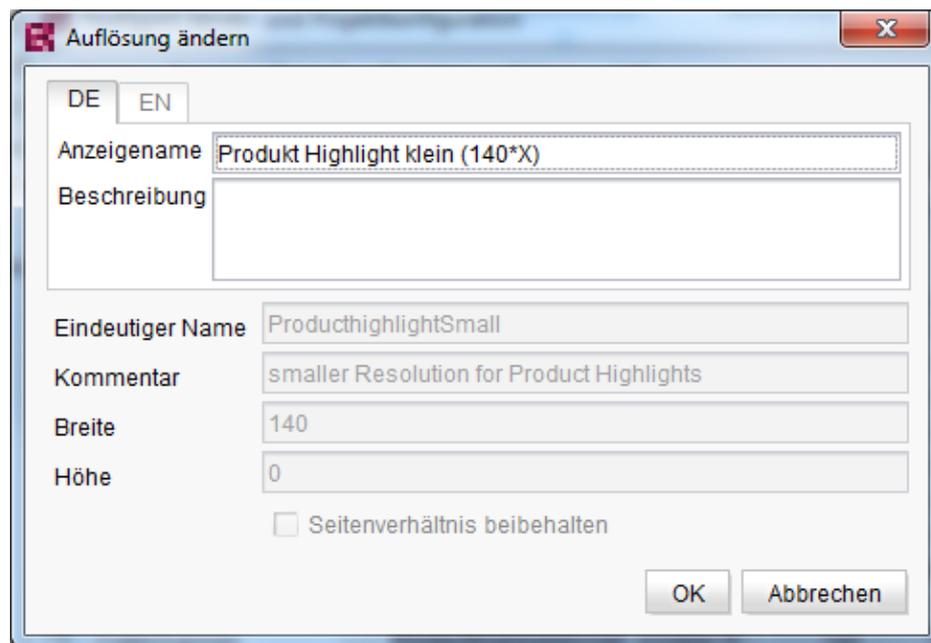


Abbildung 7-83: Neue Auflösung hinzufügen



Die Namen bereits gelöschter Auflösungen dürfen nicht wieder verwendet werden, da sonst Medien übernommen werden könnten, die nach der gelöschten Auflösung berechnet wurden.



7.4.7 Benutzer

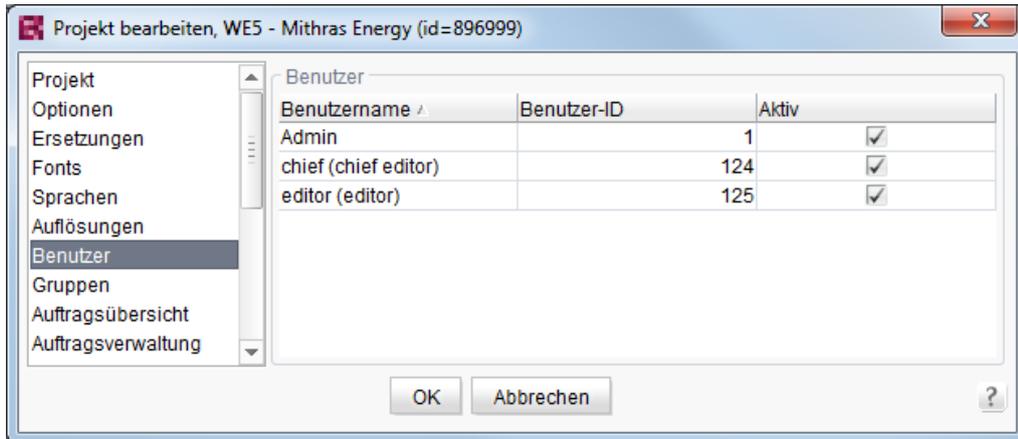


Abbildung 7-84: Projekteigenschaften – Benutzer

Hier sind alle Benutzer aufgelistet, die Zugriff auf das Projekt haben. Die Tabelle verfügt über die folgenden Spalten:

Benutzername: Hier wird der eindeutige Benutzername angezeigt, mit dem der Benutzer auf dem Server identifiziert wird. Bei manuell hinzugefügten Benutzern ist dies der Name, der beim Anlegen eines neuen Benutzers, im Feld "Login" eingetragen wird (siehe Abbildung 7-20). Benutzern, die aus einem Fremdsystem authentifiziert werden, werden ebenfalls über den Login identifiziert und unter diesem Namen hier aufgelistet. Wurde für den Benutzer ein optionaler Eintrag im Feld "Name" definiert, so wird dieser in der Liste in Klammern hinter dem Login angezeigt.

Benutzer-ID: Die ID wird automatisch vom System vergeben und ist nicht veränderbar.

Mit einem rechten Mausklick im Übersichtsfenster wird das Kontextmenü aufgerufen:



Abbildung 7-85: Kontextmenü (Bereich Benutzer)

Über das Kontextmenü können erweiterte Funktionalitäten genutzt werden, die in den nachfolgenden Kapiteln erläutert werden:



7.4.7.1 Benutzer aus dem Projekt löschen

Mit dieser Funktion wird der markierte Benutzer aus dem Projekt gelöscht. Der zu entfernende Benutzer wird in der Benutzerübersicht selektiert und dann der Kontextmenüeintrag "Löschen" aufgerufen. Der selektierte Benutzer wird aus dem Projekt entfernt.

7.4.7.2 Benutzer zum Projekt hinzufügen

Mit der Kontextmenüfunktion "Hinzufügen" können neue Benutzer für ein Projekt eingetragen werden. Es erscheint eine Auswahlliste aller auf dem Server vorhandenen Benutzer, von denen der gewünschte ausgewählt werden kann. Erst nach dieser Zuordnung erhalten die Benutzer Zugriffsrechte auf das Projekt. Über die Zuordnung zu einer Projektgruppe können die Projektbenutzer erweiterte Zugriffsrechte erhalten (siehe Kapitel 1.1.1 Seite 302).

FirstSpirit unterscheidet zwischen Benutzern, die manuell auf dem FirstSpirit Server angelegt wurden und solchen, die automatisch aus einem Fremdsystem importiert wurden:

1. Manuell angelegte Benutzer: Das manuelle Anlegen eines Benutzers erfolgt über den Menüpunkt "Benutzer" der Menüleiste der Server- und Projektkonfiguration (siehe Kapitel 7.2.4 Seite 225). Ein manuell auf dem Server angelegter Benutzer kann, innerhalb der Projekteigenschaften, einem Projekt zugeordnet werden (siehe 7.4.7 Seite 301) und ist danach automatisch ein Mitglied der Projektgruppe "Everyone", kann aber auch beliebig vielen weiteren internen Gruppen zugefügt werden (siehe Kapitel 7.4.8.6 Seite 309).
2. Automatisch angelegte Benutzer: Neben dem manuellen Anlegen eines Benutzers auf dem FirstSpirit Server, existiert noch die Möglichkeit Benutzer automatisch über ein Fremdsystem zu importieren. Benutzer, die sich über ein Fremdsystem (z. B. LDAP) auf dem FirstSpirit-Server authentifizieren, werden nach dem erstmaligen Einloggen automatisch als Benutzer auf dem FirstSpirit Server angelegt (und damit in der Liste der Benutzer angezeigt), ohne das sie explizit über die Server- und Projektkonfiguration angelegt wurden. Die Gruppenordnung erfolgt analog zu Punkt 1, wobei auch diese Benutzer zusätzlich Mitglied einer externen Gruppe sein können (siehe Kapitel 7.4.8.2 Seite 305).



7.4.8 Gruppen

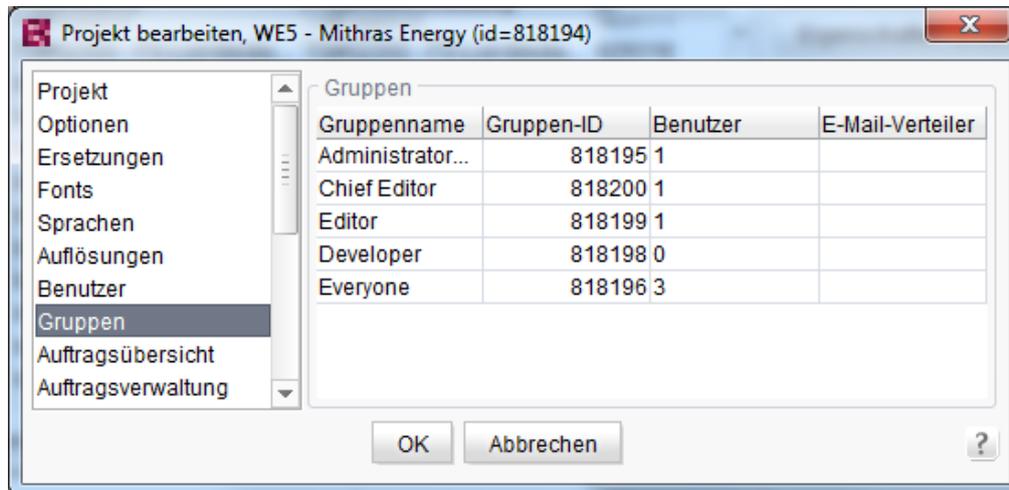


Abbildung 7-86: Projekteigenschaften – Gruppen

In diesem Bereich werden alle Gruppen aufgeführt, die Zugriff auf das Projekt besitzen. Jedes Projekt besitzt initial die Standardgruppen "Administrators" und "Everyone", die nicht gelöscht werden können. Die Gruppe "Everyone" hat in den Clients standardmäßig die Rechte "Sichtbar" und "Lesen", die Gruppe "Administrators" verfügt ab FirstSpirit Version 5.0R2 in den Clients standardmäßig über alle Rechte (zuvor konnten Projektadministratoren keine Anlege- und Löschoptionen durchführen). Zu Redaktionsrechten siehe auch *FirstSpirit Handbuch für Redakteure (JavaClient)*, Kapitel "Redaktionsrechte".

Für jedes Projekt können beliebig viele weitere Gruppen definiert werden. Gruppen enthalten eine Menge von Benutzern, aber keine weiteren Gruppen.

Gruppenname: Die Spalte zeigt die eindeutigen Gruppennamen. Bei der Auswertung wird zwischen Groß- und Kleinschreibung unterschieden.

Gruppen-ID: Die Spalte zeigt die eindeutige Gruppen-ID, die automatisch beim Anlegen einer neuen Gruppe vergeben wird.

Benutzer: Anzahl der Gruppenmitglieder. Anzahl der dieser Gruppe hinzugefügten Benutzer. Bei internen Gruppen können Benutzer zur Gruppe hinzugefügt oder entfernt werden (siehe Kapitel 7.4.7 Seite 301). Bei externen Gruppen kann die Anzahl der Benutzer nicht über die Server- und Projektkonfiguration geändert werden, statt der Anzahl der Benutzer wird hier der Vermerk "externe Gruppe" angezeigt (siehe Abbildung 7-87). (Weiterführende Informationen zu internen und externen Gruppen siehe Kapitel 7.4.8.2 Seite 305.)



E-Mail-Verteiler: Über dieses Feld können E-Mail-Verteiler für Gruppen angegeben werden, an die Benachrichtigungs-E-Mails versendet werden, wenn eine Arbeitsablauf-Aktivität oder -Transition ausgeführt wird. Auf diese Weise ist es nun möglich, E-Mails, die im Rahmen eines Arbeitsablaufs versendet werden, an alle Mitglieder von externen Gruppen zu versenden.

Gruppenname	Gruppen-...	Benutzer	E-Mail-Verteiler
Administrators ...	818195	1	
Chief Editor	818200	1	
Editor	818199	1	
Developer	818198	0	
Everyone	818196	3	mailinglist@e-Spirit.com

Abbildung 7-87: Projekteigenschaften – Gruppen: Gruppenübersicht

Gruppen und Zugriffsrechte: Mithilfe von Gruppendefinitionen, kann die Vergabe und Pflege von Zugriffsrechten stark vereinfacht werden. Soll beispielsweise ein bestimmter Bereich für eine Menge von Redakteuren unsichtbar gemacht werden und die Menge der Redakteure verändert sich gelegentlich, so bietet es sich an, eine neue Gruppe "Redakteure A" zu definieren. In diese Gruppe werden alle Redakteure eingefügt, für die der Bereich verborgen werden soll. Im FirstSpirit-JavaClient wird nun für die Gruppe "Redakteure A" die Wurzel des betroffenen Teilbaumes unsichtbar gemacht, indem die entsprechenden Rechte für die Gruppe entzogen werden. Ergibt sich zu einem späteren Zeitpunkt die Notwendigkeit für einen bestimmten Redakteur den Zugriff zu erlauben, so kann dieser Benutzer einfach aus der Gruppe "Redakteure A" entfernt werden – eine Modifikation der Rechtedefinition im FirstSpirit-JavaClient ist somit nicht erforderlich.

Mit einem rechten Mausklick im Übersichtsfenster wird das Kontextmenü aufgerufen:



Abbildung 7-88: Kontextmenü (Bereich Gruppen)

Über das Kontextmenu können erweiterte Funktionalitäten genutzt werden, die in den nachfolgenden Kapiteln erläutert werden:



- Gruppe löschen (siehe Kapitel 7.4.8.1 Seite 305)
- Neue Gruppe anlegen (siehe Kapitel 7.4.8.2 Seite 305)
- E-Mail-Verteiler bearbeiten (siehe Kapitel 7.4.8.3 Seite 307)
- Gruppe anzeigen (siehe Kapitel 7.4.8.4 Seite 308)
- Benutzer entfernen (siehe Kapitel 7.4.8.5 Seite 308)
- Benutzer hinzufügen (siehe Kapitel 7.4.8.6 Seite 309)

7.4.8.1 Gruppe löschen

Über den Kontextmenüeintrag "Gruppe löschen" können Gruppen, die einem Projekt hinzugefügt wurden, wieder entfernt werden. Vor dem expliziten Löschen wird eine Sicherheitsabfrage gestellt. Wird das Löschen der Gruppe bestätigt, wird die Gruppe aus dem Projekt entfernt und erscheint nicht mehr im Bereich Gruppen der Projekteigenschaften des Projekts. Alle Mitglieder der entfernten Gruppe verlieren, durch das Löschen der Gruppe, ihr Zugriffsrecht auf das Projekt (Ausnahme: Benutzer die Mitglied einer weiteren internen oder externen Gruppe sind, die dem Projekt weiterhin zugeordnet ist).



Die Standardgruppen "Administrators" und "Everyone" regeln die initiale Rechtevergabe in einem Projekt und können daher nicht gelöscht werden.

7.4.8.2 Neue Gruppe anlegen

FirstSpirit unterscheidet zwischen internen und externen Gruppen:

Interne Gruppen dienen zur internen Benutzer- und Rechteverwaltung und können direkt über FirstSpirit erzeugt und bearbeitet werden. Einer internen Gruppe können beispielsweise über die Projekteigenschaften Benutzer hinzugefügt oder entfernt werden. Die Eigenschaften der Gruppe können über die Server- und Projektkonfiguration bearbeitet werden. Zum Anlegen einer internen Gruppe wird lediglich das Feld "Gruppenname" gefüllt (siehe Abbildung 7-89). Nach dem Speichern des Dialogfensters "Neue Gruppe anlegen" erscheint die neue Gruppe in der Gruppenübersicht. Nach dem Erstellen, können der neuen internen Gruppe Benutzer hinzugefügt werden (siehe Kapitel 7.4.8.6 Seite 309).



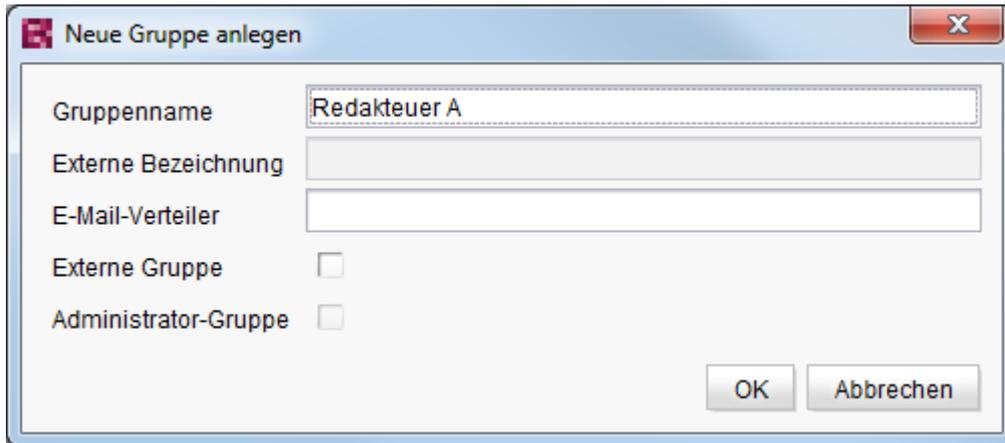


Abbildung 7-89: Neue interne Gruppe anlegen

Externe Gruppen werden ebenfalls über die Server- und Projektkonfiguration einem Projekt zugeordnet, können anders als interne Gruppen, aber nicht über FirstSpirit erzeugt werden, sondern kommen aus einem anderen System (z. B. LDAP). Die Mitgliedschaft in einer externen Gruppe wird über Benutzerattribute vergeben, das heißt, einer externen Gruppe können keine Benutzer über das Kontextmenü zugefügt werden (siehe Kapitel 7.4.8.6 Seite 309). Benutzer, die sich beispielsweise über LDAP am System authentifizieren, erhalten als Attribute die Mitgliedschaft in einer Gruppe (die nicht zwangsläufig einem Projekt zugeordnet ist) und können über diese Gruppe zum Projekt zugefügt werden. Die Mitglieder der externen Gruppe erhalten erst Zugriff auf ein Projekt, wenn die externe Gruppe dem Projekt zugeordnet wurde.

Um eine externe Gruppe im Projekt anzulegen, wird zuerst das Feld "Gruppenname" gefüllt. Das ist der interne Gruppenname, unter dem die Gruppe im FirstSpirit-Projekt bekannt ist und verwendet wird. Der externe Gruppenname, also die Bezeichnung der Gruppe im externen System, wird in das Feld "Externe Bezeichnung" eingetragen. Bei LDAP ist die "externe Bezeichnung" der LDAP-DN, z. B. `CN=Mitarbeiter,CN=Users,DC=e-spirit,DC=de`. Damit das Feld bearbeitet werden kann, muss zuvor die Checkbox "Externe Gruppe" aktiviert werden.

Bei der Prüfung der Gruppenzugehörigkeit wird intern geprüft, ob die bei "Externe Bezeichnung" angegebene Zeichenkette vollständig in einem LDAP-DN der Gruppen des angemeldeten Benutzers enthalten ist. Falls beispielweise bei "Externe Bezeichnung" nur `cn=Mitarbeiter` angegeben wird, würde die Gruppenzugehörigkeit zu den LDAP-Gruppen `cn=Mitarbeiter,ou=Entwicklung,ou=dc=domain,dc=com` und `cn=Mitarbeiter,ou=Vertrieb,dc=domain,com` passen. Um eine eindeutige



Zuordnung zu gewährleisten, muss bei "Externe Bezeichnung" der vollständige LDAP-DN der Gruppe eingetragen werden. Die Groß-/Kleinschreibung ist hier, wie allgemein bei LDAP, irrelevant.

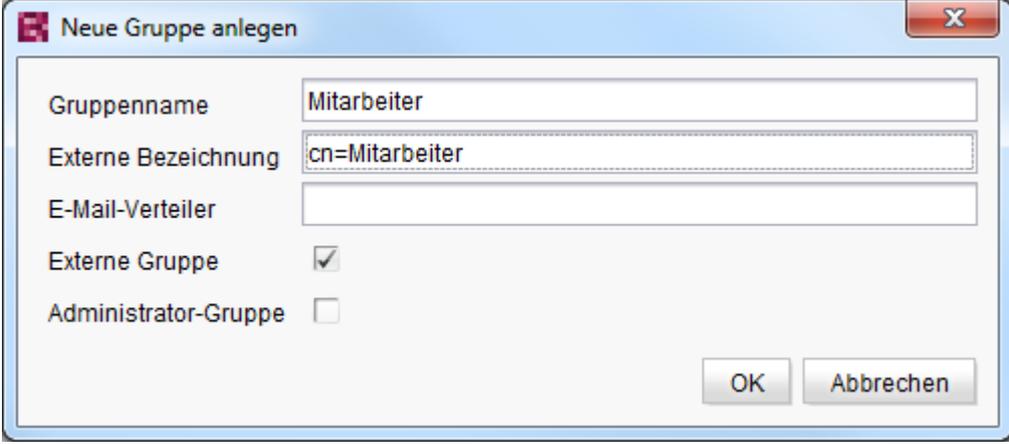


Abbildung 7-90: Neue externe Gruppe anlegen



Eine Überprüfung, ob eine externe Gruppe existiert, wird durch FirstSpirit nicht durchgeführt. Ist der externe Gruppenname nicht bekannt, wird die Gruppe dem Projekt trotzdem als externe Gruppe hinzugefügt, hat in diesem Fall aber keine Mitglieder (zugeordnete Benutzer).

7.4.8.3 E-Mail-Verteiler bearbeiten

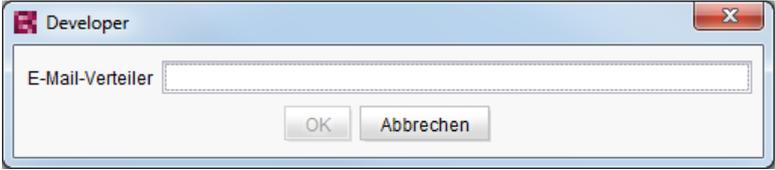


Abbildung 7-91: E-Mail-Verteiler angeben

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.



7.4.8.4 Gruppe anzeigen

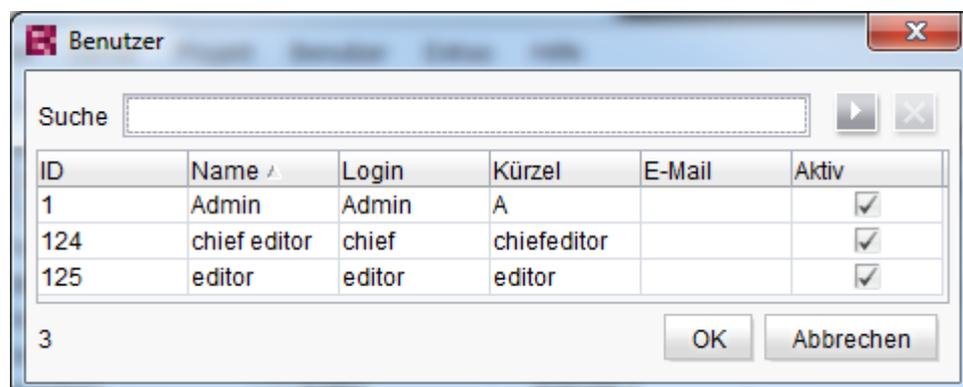


Abbildung 7-92: Gruppe anzeigen

Über den Kontextmenüeintrag "Gruppe anzeigen" können die Benutzer einer Gruppe angezeigt werden. Hier werden alle Benutzer angezeigt, die ein Mitglied der Gruppe sind und damit Zugriff auf das Projekt haben.

Handelt es sich bei der Gruppe um eine externe Gruppe, können die Mitglieder nicht angezeigt werden. Es ist ebenfalls nicht möglich, zu einer externen Gruppe Benutzer hinzuzufügen oder aus der Gruppe zu entfernen. An dieser Stelle kann nur die externe Bezeichnung der Gruppe geändert werden.

7.4.8.5 Benutzer entfernen

Für interne Gruppen erscheint eine Liste aller Mitglieder der markierten Gruppe. Der zu entfernende Benutzer wird in der Übersicht selektiert und dann die Auswahl über den Button "OK" bestätigt. Der selektierte Benutzer wird aus der internen Gruppe entfernt.

Mehrfachauswahl durch:

- Selektion in der Liste und gleichzeitiges Drücken der STRG-Taste
- STRG+ SHIFT (Selektion von Startpunkt bis Endpunkt)
- STRG + A (Selektion aller Benutzer)



Die Mitgliedschaft in einer externen Gruppe wird über Benutzerattribute des Fremdsystems vergeben, das heißt, die Benutzer einer externen Gruppe können über die Server- und Projektkonfiguration nicht entfernt werden.





Für die Standardgruppe "Everyone" können keine Benutzer hinzugefügt oder gelöscht werden. Für diese Gruppe gilt: Alle Benutzer, die Zugriff auf das Projekt haben (über die Mitgliedschaft in einer, dem Projekt zugeordneten, internen oder externen Gruppe), sind automatisch Mitglied der Gruppe "Everyone" und erhalten mindestens die für "Everyone" definierten Zugriffsrechte.

7.4.8.6 Benutzer hinzufügen

Benutzer können auch Mitglied in mehreren Gruppen sein. Jeder Gruppe können unterschiedlich konfigurierte Zugriffsrechte zugewiesen werden (siehe Kapitel 1.1.1 Seite 302). Die Mitglieder der Gruppe erhalten alle Zugriffsrechte der Gruppe auf das Projekt.

Für interne Gruppen erscheint eine Liste aller Benutzer, die zu diesem Zeitpunkt zum Projekt zugefügt und aktuell kein Mitglied der selektierten Gruppe sind. Die hinzuzufügenden Benutzer können in der Übersicht selektiert werden. Wird die Auswahl über den Button "OK" bestätigt, werden die selektierten Benutzer der internen Gruppe hinzugefügt.

Mehrfachauswahl durch:

- Selektion in der Liste und gleichzeitiges Drücken der STRG-Taste
- STRG+ SHIFT (Selektion von Startpunkt bis Endpunkt)
- STRG + A (Selektion aller Benutzer)



Die Mitgliedschaft in einer externen Gruppe wird über Benutzerattribute des Fremdsystems vergeben, das heißt, einer externen Gruppe können über die Server- und Projektkonfiguration keine Benutzer zugefügt werden.



Für die Standardgruppe "Everyone" können keine Benutzer hinzugefügt oder gelöscht werden. Für diese Gruppe gilt: Alle Benutzer, die Zugriff auf das Projekt haben (über die Mitgliedschaft in einer, dem Projekt zugeordneten, internen oder externen Gruppe), sind automatisch Mitglied der Gruppe "Everyone" und erhalten mindestens die für "Everyone" definierten Zugriffsrechte.



7.4.9 Auftragsübersicht

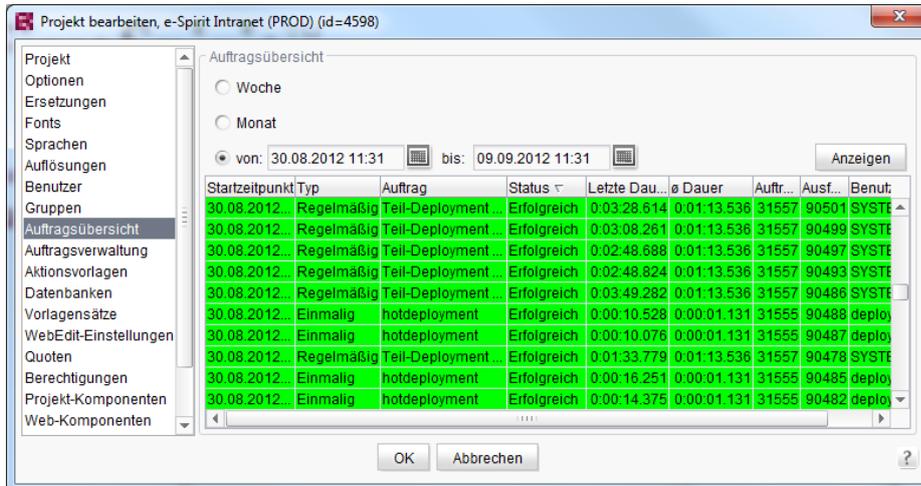


Abbildung 7-93: Projekteigenschaften – Auftragsübersicht

In diesem Bereich werden Details zur Auftragsplanung angezeigt. Neben den projektbezogenen Einstellungen in diesem Bereich können auch serverbezogene Einstellungen zur Auftragsplanung definiert werden (siehe Kapitel 7.3.9 Seite 252). Eine vollständige Übersicht der Auftragsplanung mit FirstSpirit erfolgt daher im übergeordneten Kapitel 7.5 (Seite 349 ff.).



7.4.10 Auftragsverwaltung

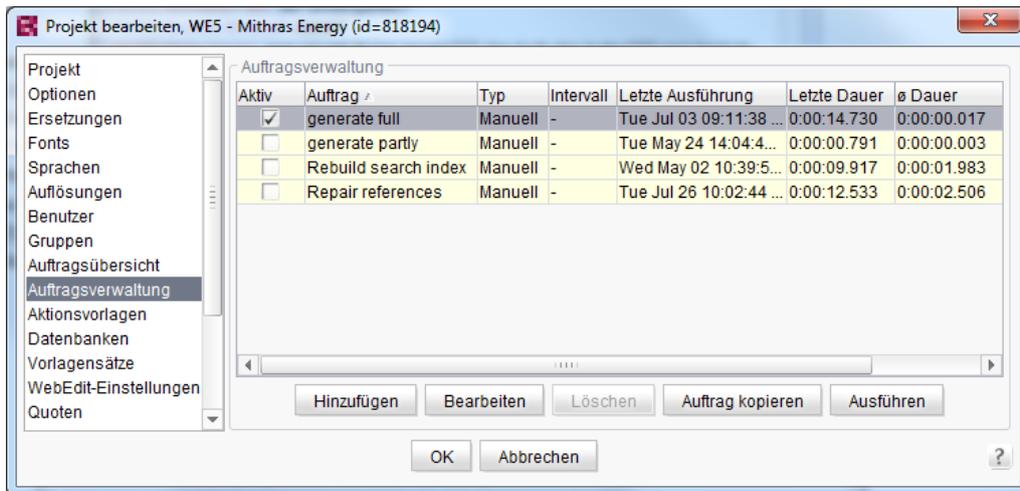


Abbildung 7-94: Projekteigenschaften – Auftragsverwaltung

Die Auftragsplanung stellt die Möglichkeit der Erstellung von Aufträgen zur Verfügung. Aufträge dienen zur sinnvollen Zusammenfassung sich ergänzender Aktionen und sind, je nach Erstellungs-Ort, entweder server- oder projektbezogen (siehe Kapitel 7.5.2 Seite 354).

7.4.11 Aktionsvorlagen

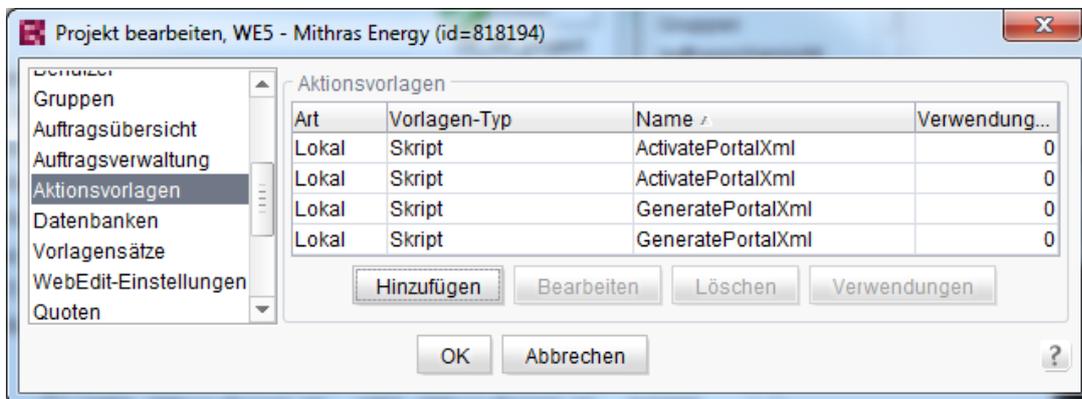


Abbildung 7-95: Projekteigenschaften – Aktionsvorlagen

Siehe Kapitel 7.5.3 Seite 358.



7.4.12 Datenbanken

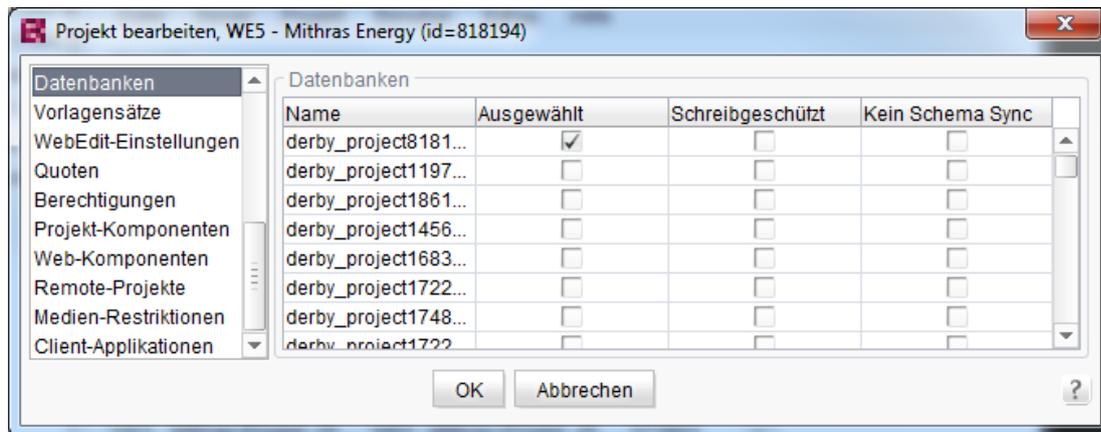


Abbildung 7-96: Projekteigenschaften – Datenbanken

Hier sind alle Datenbanken aufgelistet, die für das Projekt definiert werden. Die Tabelle verfügt über die folgenden Spalten:

Name: Hier werden die Namen aller auf dem Server liegenden Datenbanken angezeigt.

Ausgewählt: Jede Datenbank, für die dieses Häkchen gesetzt wird, kann im FirstSpirit-Client für ein Datenbank-Schema verwendet werden. Damit ist automatisch der schreibende Zugriff auf die Datenbank für dieses Projekt gesetzt.

Schreibgeschützt: Soll die ausgewählte Datenbank in dem Projekt nur schreibgeschützt zur Verfügung stehen, muss diese Option aktiviert sein. Für externe Datenbanken ("Kein Schema Sync" ist aktiviert) muss der Schreibschutz aktiviert werden.

Kein Schema Sync: Ist diese Option aktiviert, werden Änderungen am Datenbankschema, die im FirstSpirit-JavaClient vorgenommen werden, nicht in die physikalische Datenbank übernommen. Diese Einstellung muss für die Anbindung externer Datenbanken aktiviert werden (siehe Kapitel 4.8.8 Seite 173).



7.4.13 Vorlagensätze

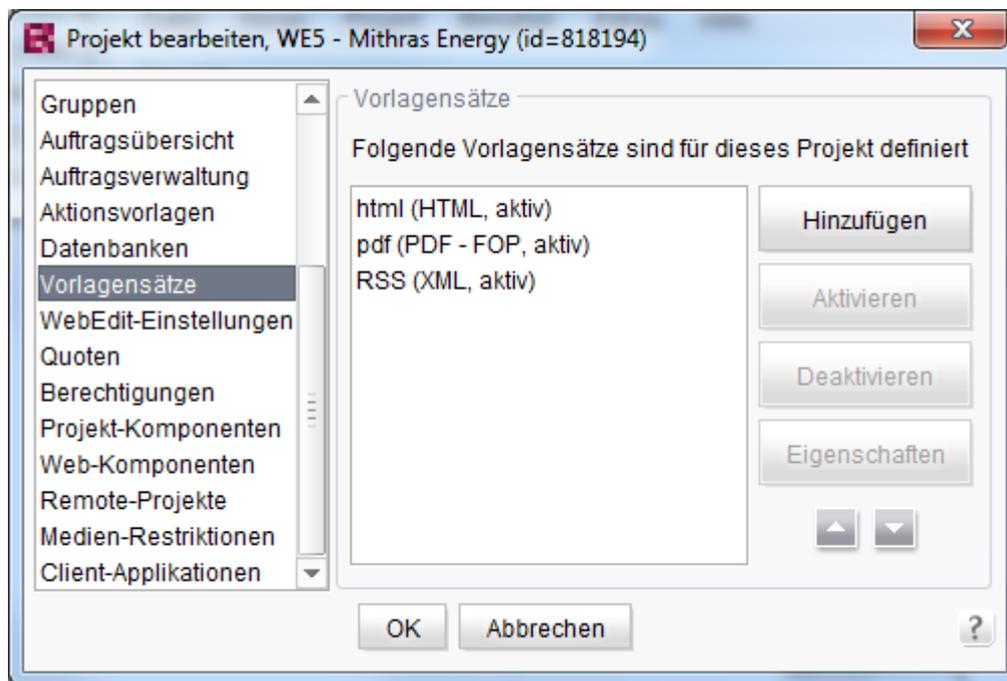


Abbildung 7-97: Projekteigenschaften – Vorlagensätze

Aus den Präsentationskanälen des Servers werden Vorlagensätze definiert, die in dem Projekt zur Verfügung stehen sollen.

Durch Klick auf den Button "Hinzufügen" öffnet sich ein Fenster, in dem die Einstellungen für den neuen Vorlagensatz vorgenommen werden können.



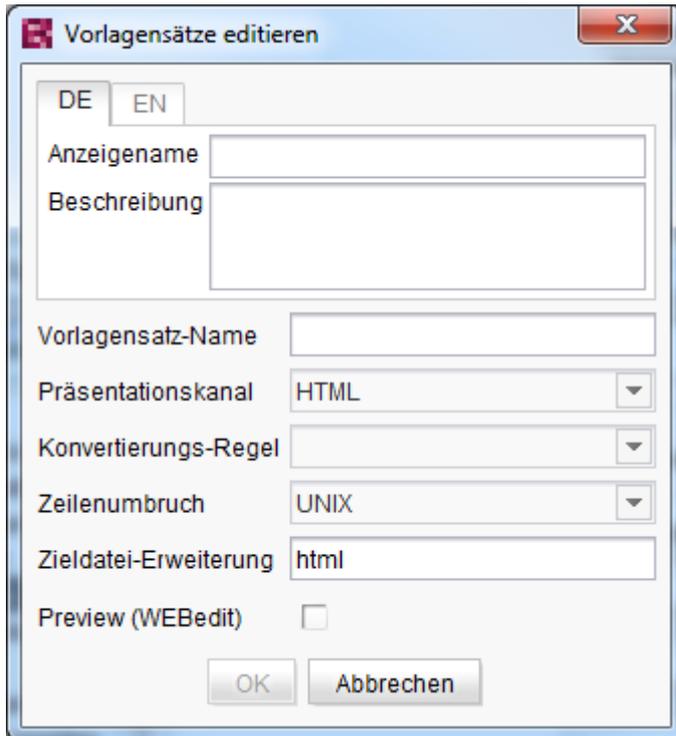


Abbildung 7-98: Neuen Vorlagensatz hinzufügen

Vorlagensatz-Name: Hier wird der Name des Vorlagensatzes eingetragen, der im JavaClient bei den Seiten- Absatz- und Formattemplates angezeigt wird.

Präsentationskanal: In diesem Feld kann ein unter Server-Eigenschaften definierter Präsentationskanal ausgewählt werden. (Siehe Kapitel 7.3.2 Seite 235)

Konvertierungs-Regel: In diesem Feld kann eine in den Server-Eigenschaften definierte Konvertierungsregel ausgewählt werden. (Siehe Kapitel 7.3.3 Seite 237)

Zeilenumbruch: Alle zu erzeugenden Zeilenumbrüche aus den Vorlagensätzen werden durch Betriebssystem spezifische Zeilenumbrüche ersetzt. Für die Standardeinstellung "Windows" wird der Zeilenumbruch automatisch durch die Steuerzeichen "carriage return" und "line feed" [\r\n] ersetzt, für "Unix" wird "line feed" [\n] und für "Mac" wird "carriage return" [\r] ersetzt.

Zieldatei-Erweiterung: Hier wird die entgültige Endung der Datei auf dem Webserver angegeben. Die von FirstSpirit erzeugte Verlinkung zu dieser Datei setzt sich aus dem in der Struktur-Verwaltung verwendeten Dateinamen und dieser Endung zusammen.

Preview (WebEdit): Standardeinstellung für die Vorschau-URL ist `/fs5preview/preview`. Die Vorschau der einzelner Projekte erfolgt, soweit konfiguriert, auf



`/fs5preview_{PID}/preview`. Wird die Checkbox "Preview (WebEdit)" aktiviert, wird die aktuell angeforderte Vorschau im WebClient angezeigt.

Durch einen Klick auf den Button "Aktivieren" wird der ausgewählte Vorlagensatz aktiviert. (Der aktuelle Status wird in Klammern angezeigt.)

Durch einen Klick auf den Button "Deaktivieren" wird der ausgewählte Vorlagensatz deaktiviert. (Der aktuelle Status wird in Klammern angezeigt.)

Durch Klick auf den Button "Eigenschaften" öffnet sich ein Fenster, in dem die Einstellungen für den ausgewählten Vorlagensatz bearbeitet werden können. (Siehe Abbildung 7-98: Neuen Vorlagensatz hinzufügen.)

Anzeigename / Beschreibung: Für jeden Vorlagensatz kann ein sprachabhängiger Anzeigename und eine sprachabhängige Beschreibung in allen Redaktionssprachen vergeben werden.

Wird im FirstSpirit-JavaClient eine "bevorzugte Anzeigesprache" definiert, werden die entsprechenden sprachabhängigen Anzeigenamen der Vorlagensätze im JavaClient angezeigt.

Sind ein oder mehrere Generierungsaufträge vorhanden, kann gewählt werden, ob die Generierung für einen neu hinzugefügten Vorlagensatz durchgeführt werden soll: "Soll dieser Vorlagensatz von allen Aufträgen dieses Projektes generiert werden?" Mit "Ja" werden die Häkchen in Abbildung 7-149 entsprechend gesetzt.

 Die Vorlagensätze eines Projekts können innerhalb der Server- und Projektkonfiguration umsortiert werden. Dazu muss einfach der entsprechende Eintrag selektiert und über die entsprechenden Buttons schrittweise nach oben oder unten verschoben werden. Die geänderte Reihenfolge wirkt sich dann auf die Reihenfolge der Register im JavaClient aus.



7.4.14 WebEdit-Einstellungen

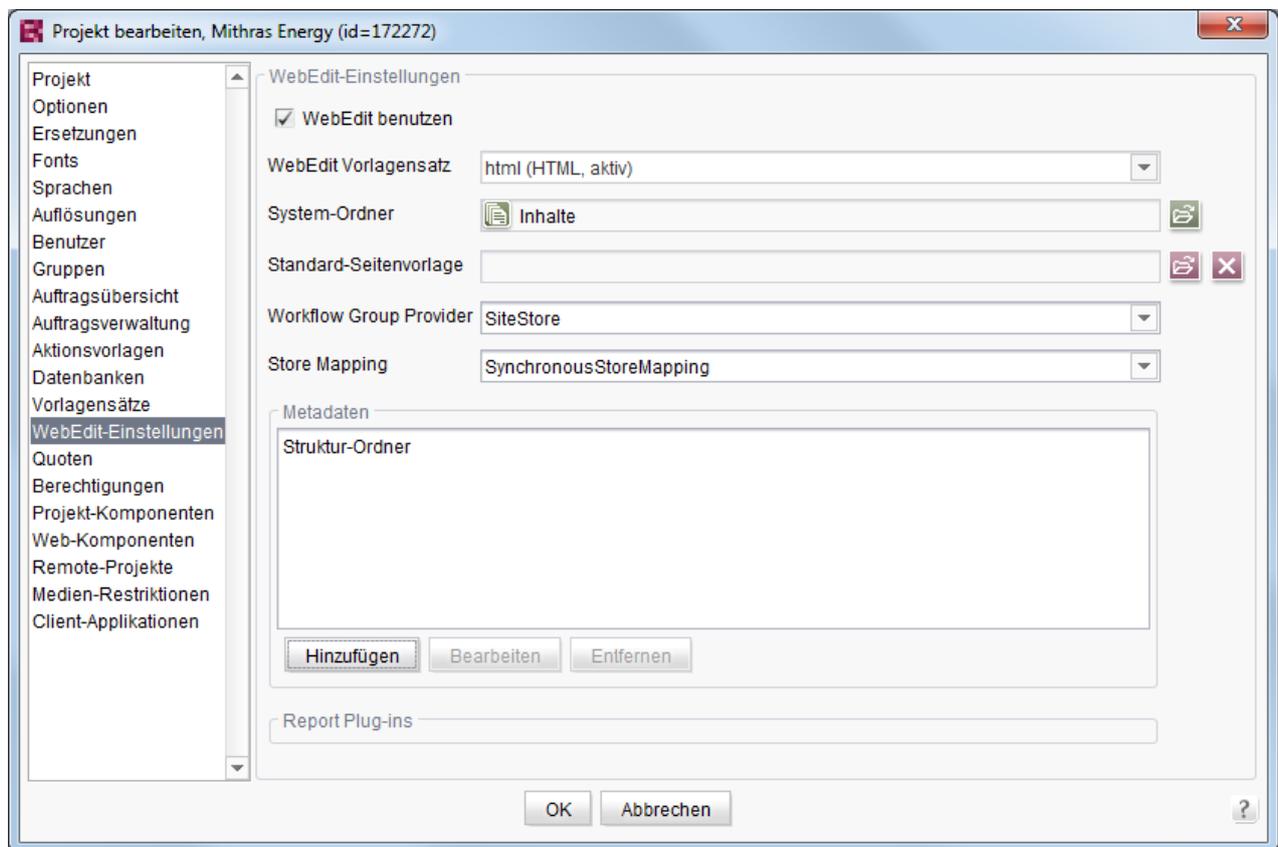


Abbildung 7-99: Projekteigenschaften – WebEdit-Einstellungen

WebEdit benutzen: Ist diese Option aktiviert, dann kann dieses Projekt im WebClient (Autoren Umgebung) bearbeitet werden. Ist diese Option deaktiviert, kann das Projekt nicht mehr im WebClient bearbeitet werden.

WebEdit Vorlagensatz: Über diese Option kann aus den dem Projekt hinzugefügten Vorlagensätzen ein Vorlagensatz für den WebClient ausgewählt werden. (siehe auch Kapitel 7.4.13 Seite 313).

System-Ordner: Über die Auswahl "System-Ordner" kann festgelegt werden, in welchem Ordner der Inhalte-Verwaltung neu angelegte Seiten abgelegt werden sollen. Es kann jeder bereits bestehende Ordner als Ziel der im WebClient angelegten Seiten gewählt werden. Alternativ kann auch der Wurzelknoten der Inhalte-Verwaltung gewählt werden ("Inhalte"). Dieser ist initial voreingestellt. Mit dieser Einstellung werden **alle** neuen Seiten unterhalb des Wurzelknotens eingefügt.

Da im WebClient standardmäßig Seiten mit Menüpunkt angelegt werden, also Ordner und Seitenreferenz in der Struktur-Verwaltung, wird auch in der Inhalte-



Verwaltung die Seite in einem Ordner angelegt. Dieser Ordner wird unterhalb des ausgewählten System-Ordners angeordnet.

Eine Seite "Unternehmen" mit Menüpunkt in der Standardeinstellung *Inhalte* als System-Ordner stellt sich im WebClient und im JavaClient folgendermaßen dar:

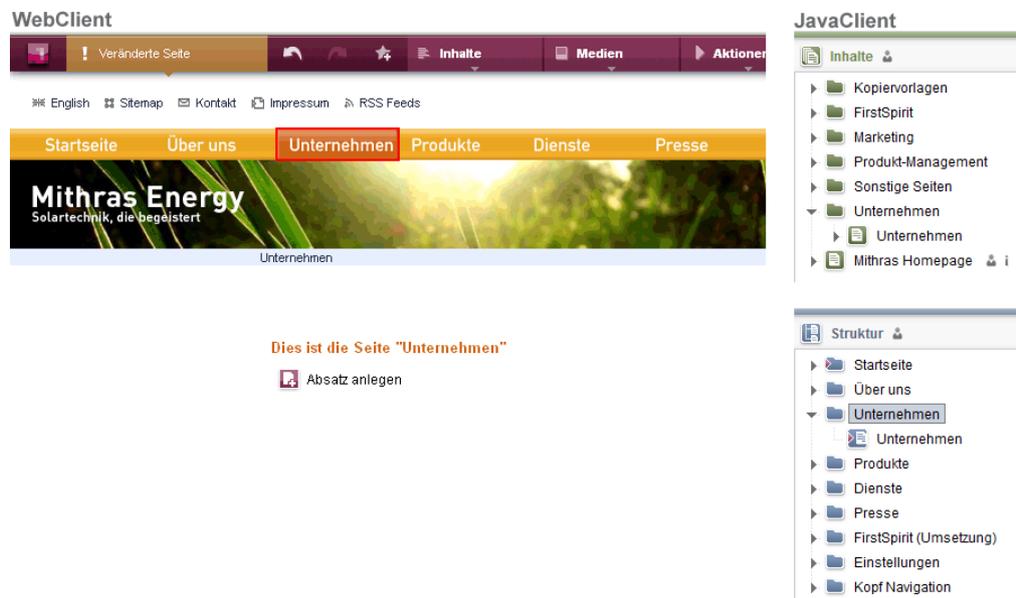


Abbildung 7-100: Seite mit Menüpunkt im WebClient und JavaClient

Wird der Menüpunkt im WebClient entfernt ("Inhalte" / "Bearbeiten" / "Menüpunkt in Seite umwandeln"), verschwindet der Menüpunkt "Unternehmen" aus der Navigation, im JavaClient wird in der **Struktur-Verwaltung** und in der **Inhalte-Verwaltung** der entsprechende Ordner entfernt, die Seite bleibt unterhalb dem gewählten "System-Ordner" erhalten.



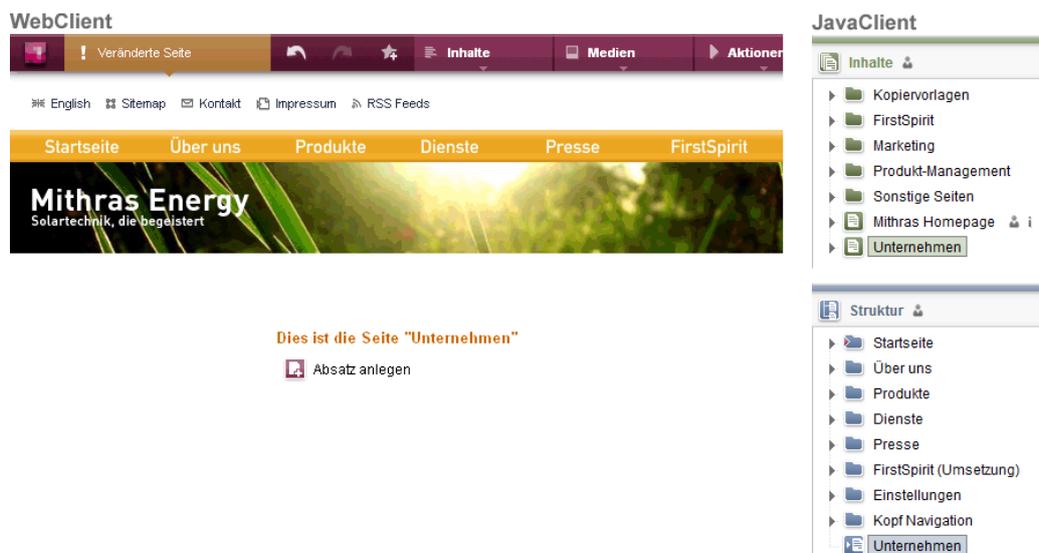


Abbildung 7-101: Seite mit gelöschtem Menüpunkt im WebClient und JavaClient

Standard-Seitenvorlage: Über diese Auswahl kann eine Seitenvorlage bestimmt werden, die dann beim Anlegen von Seiten im WebClient ("Inhalte" / "Neu" / "Neue Seite anlegen") vorausgewählt wird. Im Bereich "Layout" des entsprechenden Assistenten (siehe dazu auch *FirstSpirit Handbuch für Redakteure (WebClient)*) braucht der Redakteur dann keine Vorlage mehr auszuwählen. Bei Bedarf kann er aber auch eine andere Vorlage wählen. Initial ist keine Standard-Seitenvorlage ausgewählt. Eine getroffene Auswahl kann über die entsprechende Schaltfläche wieder auf den initialen Zustand zurückgesetzt werden.

Workflow Group Provider: Arbeitsabläufe werden im WebClient standardmäßig über den Seiten-Status in der Menüleiste durchgeführt (Einträge "Veränderte Seite", "im Arbeitsablauf", siehe dazu auch *WebClient-Dokumentation*). Dabei kann über den Workflow Group Provider eingestellt werden, auf welches FirstSpirit-Objekt/e sich der Arbeitsablauf beziehen soll, z. B. auf Seiten der Struktur-Verwaltung oder Seiten der Inhalte-Verwaltung. Folgende Provider werden standardmäßig mit FirstSpirit ausgeliefert und können aus dem Drop-Down-Menü ausgewählt werden:

- **SiteStore:** Wird dieser Provider ausgewählt, wirken sich Arbeitsabläufe, die im WebClient gestartet werden, nur auf Seiten in der Struktur-Verwaltung aus. Das bedeutet, wird z. B. ein Freigabe-Arbeitsablauf auf der Seite im WebClient durchgeführt, auf der der Redakteur sich aktuell befindet, ist nach Beendigung des Arbeitsablaufs nur die entsprechende Seite in der Struktur-Verwaltung freigegeben, nicht aber die zugehörige Menüebene und nicht die zugehörige Seite in der Inhalte-Verwaltung. Dieser Provider ist standardmäßig ausgewählt.



- **PageStore:** Wird dieser Provider ausgewählt, wirken sich Arbeitsabläufe, die im WebClient gestartet werden, nur auf Seiten in der Inhalte-Verwaltung aus. Das bedeutet, wird z. B. ein Freigabe-Arbeitsablauf auf der Seite im WebClient durchgeführt, auf der der Redakteur sich aktuell befindet, ist nach Beendigung des Arbeitsablaufs nur die entsprechende Seite in der Inhalte-Verwaltung freigegeben, nicht aber die zugehörige Seitenreferenz und die zugehörige Menüebene in der Struktur-Verwaltung. Eine mit diesem Provider freigegebene Seite wird dann u. U. bei der nächsten Generierung nicht berücksichtigt.

Es wird empfohlen, für jedes Projekt einen an die jeweiligen Anforderungen angepassten Provider zu erstellen, mit dem z. B. auch Medien oder mehrere Objekte gleichzeitig freigegeben werden können. Siehe dazu auch *FirstSpirit Online Dokumentation*, Bereich "Plugin-Entwicklung" / "WebClient-Erweiterungen" / "Interaktive Features" / "Elementstatus- und Arbeitsablaufanzeigen" (EN). Dort ist u.a. auch ein Beispiel-Arbeitsablauf zur Freigabe von Seiten inklusive veränderter Medien verfügbar ("WebClient Examples - Element Status Provider - Store Type").

Store Mapping: Um in FirstSpirit neue Inhalte anlegen zu können, wird immer eine Seite in der Inhalte-Verwaltung als Grundlage benötigt. Beim Anlegen neuer Seiten im WebClient (Menü "Inhalte" / "Neu" / "Neue Seite") wird vom Redakteur nur die Position in der Struktur-Verwaltung bestimmt, nicht aber die Position in der Inhalte-Verwaltung. Über das Drop-Down-Menü "Store Mapping" kann festgelegt werden, wie und wo neue Seiten in der Inhalte-Verwaltung abgelegt werden sollen und wie beim Umbenennen oder beim Verschieben von Seiten / Menüebenen im WebClient in der Inhalte-Verwaltung vorgegangen werden soll.

Je nach Projektanforderungen und -architektur identifiziert ein aus diesem Drop-Down-Menü gewähltes Plug-in z. B. – ausgehend von Ordner und Seite in der Struktur-Verwaltung – den passenden Ordner in der Inhalte-Verwaltung, in dem eine im WebClient angelegte Seite erstellt wird, oder legt bei Bedarf einen passenden Ordner bzw. eine Ordnerhierarchie an.

Standardmäßig wird das "SynchronousStoreMapping" mit FirstSpirit ausgeliefert. Es spiegelt die im WebClient angelegte Seiten-Struktur in den unter "**System-Ordner**" (siehe oben) ausgewählten Ordner. Wird eine neue Seite im WebClient angelegt, bildet dieses Plug-in die Struktur / Hierarchie der Struktur-Verwaltung in den "**System-Ordner**" in der Inhalte-Verwaltung ab, und zwar von der Wurzel (der Struktur-Verwaltung) bis zur neu erstellten Seitenreferenz.

Für Informationen über die Entwicklung eigener Plug-ins siehe *FirstSpirit Online Dokumentation*, Bereich "Plugin-Entwicklung" / "WebClient-Erweiterungen" / "Administrative Erweiterungen" / "Storesynchronisierung" (EN).



Metadaten: Über sogenannte Metadaten können zusätzliche Informationen zu FirstSpirit-Objekten hinterlegt werden. Dazu kann Redakteuren eine spezielle Seitenvorlage zur Verfügung gestellt werden, über die die gewünschten Informationen abgefragt und erfasst werden können. Im JavaClient steht das auf der für Metadaten definierten Vorlage (siehe dazu auch Kapitel 7.4.2 Seite 285, "Metadaten-Vorlage") basierende Formular direkt zur Verfügung. Damit auch über den WebClient Metadaten erfasst werden können, muss für jeden FirstSpirit-Objekttyp, auf dem im WebClient Metadaten erfasst werden können sollen, ein Eintrag in dieser Liste erstellt werden. Folgende FirstSpirit-Objekttypen stehen dabei zur Verfügung:

- Seite
- Absatz
- Seitenreferenz
- Struktur-Ordner (Menüebene)
- Medium
- Medien-Ordner

Hinzufügen: Der gewünschte FirstSpirit-Objekttyp kann über die Schaltfläche "Hinzufügen" der Liste hinzugefügt werden. Aus dem sich öffnenden Drop-Down-Menü kann der gewünschte Typ ausgewählt werden. Im folgenden Dialog muss für den Typ ein aussagekräftiger Anzeigename für jede Projektsprache angegeben werden. Dieser Anzeigename wird dem Redakteur im WebClient als Teil der Menübeschriftung zur Metadaten-Anlage und -Bearbeitung angezeigt, z. B.

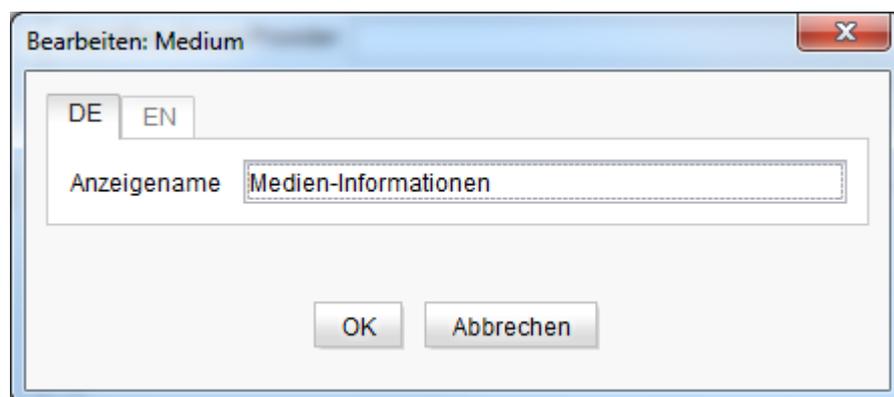


Abbildung 7-102: Projekteigenschaften – WebClient-Metadaten

Der im Beispiel für Medien gewählte Anzeigename "Medien-Informationen" wird dem Redakteur im WebClient z. B. beim Hochladen von Medien in das Projekt angezeigt:



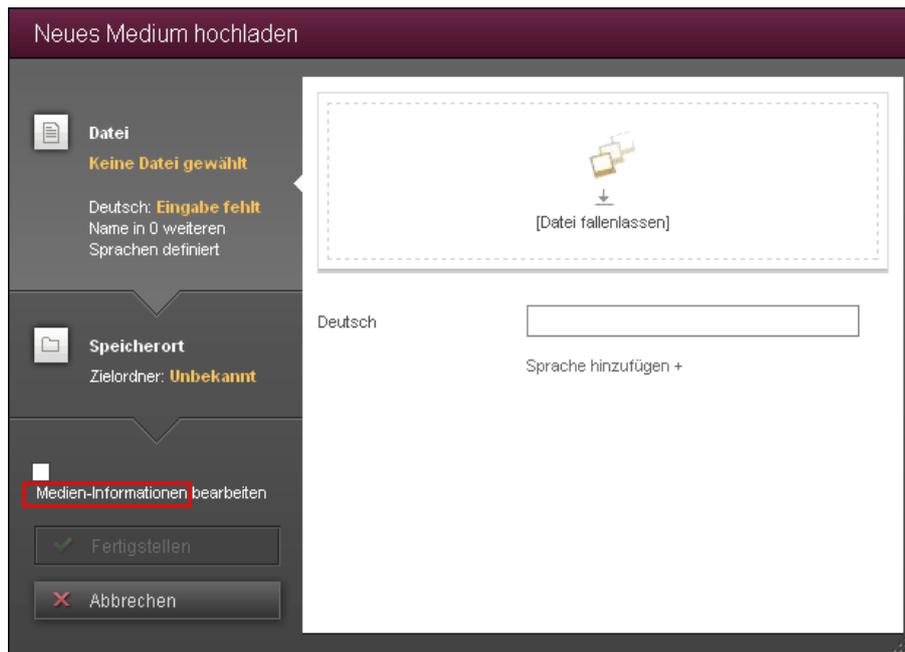


Abbildung 7-103: Metadaten beim Hochladen von Medien

Jeder Typ kann nur einmal ausgewählt werden.

Bearbeiten: Sollen bereits gewählte Anzeigenamen in einer oder allen Sprachen geändert werden, kann dies über die Schaltfläche "Bearbeiten" bewerkstelligt werden. Es öffnet sich der Dialog aus Abbildung 7-102. Es wird dann nur der Menüname im WebClient geändert, auf bereits erfasste Metadaten wirkt sich diese Änderung nicht aus.

Entfernen: Über diese Schaltfläche kann das Erfassen von Metadaten im WebClient unterbunden werden. Sollen über den WebClient keine Metadaten mehr oder nur noch zu bestimmten FirstSpirit-Objekttypen erfasst werden können, können die Typen, die nicht mehr erforderlich sind, über die Schaltfläche "Entfernen" aus der Liste entfernt werden. Die entsprechende Menüfunktion (siehe z. B. Abbildung 7-103) wird dann entfernt, auf bereits erfasste Metadaten wirkt sich diese Änderung nicht aus.

Report Plug-ins: Hier werden auf dem Server vorhandene Report Plug-ins angezeigt (siehe dazu FirstSpirit Online Dokumentation, Kapitel "Plugin-Entwicklung" / "WebClient-Erweiterungen" / "Interaktive Features" / "Reports"). Wird die Checkbox aktiviert, kann das Plug-in im jeweiligen Projekt verwendet werden. Wird die Checkbox deaktiviert, kann das Plug-in im jeweiligen Projekt nicht mehr verwendet werden.





Die Verwendung von selbstimplementierten Reports im WebClient erfordert eine Applikationsintegrations-Lizenz. Der zugehörige Lizenzparameter `license.APPTAB_SLOTS` gilt für alle FirstSpirit Applikationsintegrationen im WebClient und im JavaClient und gibt an, wie viele Reports bzw. AppCenter-Anwendungen verwendet werden können (siehe dazu auch Kapitel 4.3.5 Seite 94 und Kapitel 8.6.2.6 Seite 464). Wird im Bereich "Report Plug-ins" ein WebClient-Report aktiviert, wird geprüft, ob die aktuelle Lizenz-Konfiguration in Verbindung mit der aktuellen Belegung das Aktivieren des Reports zulässt. Ist dies nicht der Fall, wird folgende Warnung angezeigt: "Das überschreitet die Anzahl der lizenzierten Applikationsintegrationen!" Im WebClient wird das zugehörige Report-Icon zwar angezeigt, es ist aber inaktiv.



Sollen in einer Projektkonfiguration Workflow Group Provider-, Store Mapping- oder Report-Plugins verwendet werden, die mittels FirstSpirit-Modulen (FSM) im FirstSpirit-Server installiert worden sind, muss eine projektspezifische WebEdit-Web-Applikation installiert und aktiviert sein (siehe Kapitel 7.4.18 Seite 327). Zu dieser Web-Applikation müssen Web-Komponenten hinzugefügt werden, die die von diesen Plugins verwendeten Java-Klassen und benötigte Ressourcen beinhalten. Die Konfiguration solcher Web-Komponenten in einem FSM sowie die projektspezifische Bereitstellung von WebClient-Plugins per Web-Komponente werden in der FirstSpirit Online Dokumentation, Bereich "Plugin-Entwicklung" / "WebClient-Erweiterungen" / "Entwicklung und Bereitstellung" (EN) beschrieben.



7.4.15 Quoten

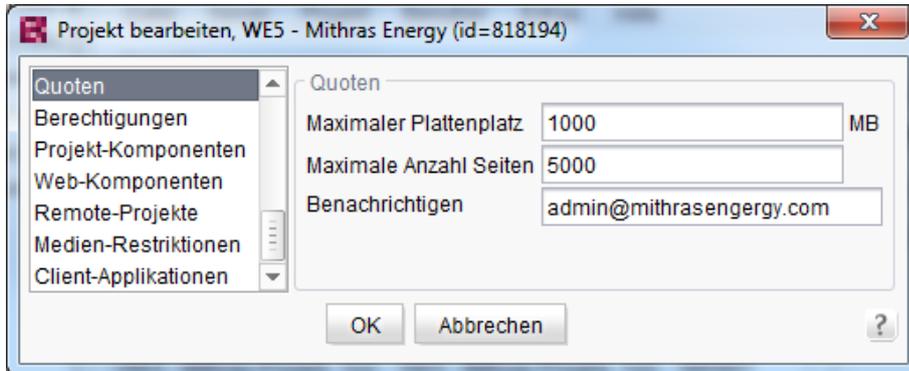


Abbildung 7-104: Projekteigenschaften – Quoten

Maximaler Plattenplatz: Gibt an, welcher maximale Speicherplatz dem Projekt auf dem CMS-Server zur Verfügung gestellt werden soll.

Maximale Anzahl Seiten: Gibt an, welche Anzahl von Seiten das Projekt maximal haben darf.

Benachrichtigen: In diesem Feld wird angegeben, wer benachrichtigt werden soll, falls die Speicher- oder Seitengrenze erreicht wurde.

7.4.16 Berechtigungen

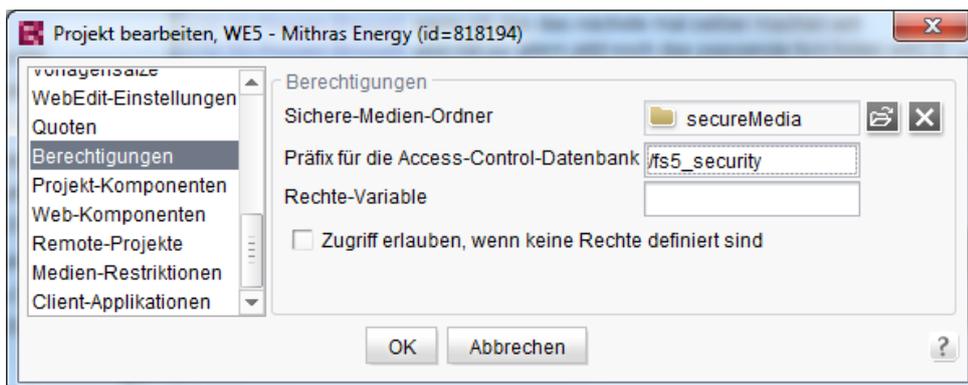


Abbildung 7-105: Projekteigenschaften – Berechtigungen

Sichere Medien Ordner: Über das Ordner-Icon öffnet sich ein neuer Dialog "Sichere Medien Ordner" mit der Baumstruktur der Medien-Verwaltung. Es kann ein Ordner aus der Medien-Verwaltung ausgewählt werden, der die sicheren Medien für das Projekt enthalten soll. Der "Sichere-Medien-Ordner" wird im JavaClient gesondert gekennzeichnet . Über das Löschen-Icon kann eine bereits getroffene



Auswahl wieder entfernt werden (vgl. Kapitel 11.3 Seite 523).



Die hier definierten Einstellungen für "Sichere Medien" wirken sich nur auf die FirstSpirit-Vorschauerzeugung aus. Ein Schutz der generierten bzw. veröffentlichten Inhalte vor unberechtigtem Zugriff ist nur über das Modul FirstSpirit Security möglich³¹ (vgl. Kapitel 11.3 Seite 523).

Präfix für die Access-Control-Datenbank: Angabe eines Präfixes zur Vervollständigung der ACL-Datenbank-Informationen einer Datei. Der vollständige Pfad zu einer Datei in FirstSpirit besteht immer aus drei Teilen:

- JURL der Webanwendung (z. B. http://meinServer.de)
- Präfix für die Access-Control-Datenbank (/fs5_security)
- Pfad zu einer Datei (/de/index.html)

Der vollständige Pfad innerhalb der ACL-Datenbank setzt sich aus dem Präfix und dem Pfad zur Datei zusammen, z. B. /fs5_security/de/index.html

In Abbildung 7-105 lautet das Präfix "/fs5_security" und entspricht dem direkten Unterverzeichnis "fs5_security" der Webanwendung, beispielsweise "live":
"~Webserver/webapps/live/fs5_security".

Die Angabe des Präfixes ist absolut zur Webanwendung. Zusätzlich entspricht das Präfix in diesem Fall dem letzten Teil des Feldwertes "Pfad auf Live-Server" im Dialog des Veröffentlichungs-Servlets (siehe Kapitel 7.5.10.6.3 Seite 399).

Weitere Informationen siehe Dokumentation zum Modul FirstSpirit Security.

Rechte-Variable: Die Zugriffsrechte auf Objekte lassen sich über die Metadaten in der Medien-Verwaltung definieren. Dafür wird in einer Seitenvorlage eine Eingabekomponente definiert (CMS_INPUT_PERMISSION). Der für diese Eingabekomponente vergebene Name ist hier anzugeben. FirstSpirit entscheidet dann anhand des Wertes der Eingabekomponente im Register Metadaten, ob ein Benutzer Zugriff auf ein Objekt hat oder nicht.

Weitere Informationen siehe Dokumentation zum Modul FirstSpirit Security.

Zugriff erlauben, wenn keine Rechte definiert sind: Ist die Checkbox aktiviert, wird der Zugriff auf "Sichere Medien" für alle Gruppenmitglieder (aus der Datei

³¹ Siehe Dokumentation: SECU50DE_FirstSpirit_Modules_Security.pdf



groups.xml) erlaubt, wenn keine anderslautende Definition über die Rechte-Variable definiert wurde. Ist die Checkbox deaktiviert, werden nur die über die Rechte-Variable gesetzten Werte ausgewertet.

Weitere Informationen zu Benutzerrechten siehe Kapitel 11 Seite 512.

Weitere Informationen zum Konzept "Sichere Medien" siehe Kapitel 11.3 Seite 523.

7.4.17 Projekt-Komponenten



Abbildung 7-106: Projekteigenschaften – Projektkomponenten

In diesem Bereich werden Komponenten gespeichert, die für das gesamte Projekt zur Verfügung stehen sollen (nicht nur für einzelne Bereiche des Projekts – vgl. Kapitel 7.4.18). Zur Installation einer Projektanwendung sind bestimmte Schritte erforderlich. So muss beispielsweise zur Installation der Portalkomponente in einem Projekt zuerst das entsprechende Modul auf dem Server installiert werden (siehe 7.3.11 Seite 254). Ein Beispiel für einen Projekt-Komponente ist die FirstSpirit-Portal-Komponente, die Bestandteil des Moduls "FS Portal" ist (siehe Dokumentation FirstSpirit Portal).

Mit einem Klick auf den Button "Hinzufügen" können anschließend die Komponenten (vom Typ "Projekt-Applikationen") einem Projekt hinzugefügt werden. Danach stehen im Projekt alle Funktionalitäten der installierten Komponente zur Verfügung. (Im Fall der Portalkomponente werden beispielsweise Portal-Ordner innerhalb der Vorlagen-Verwaltung des Projekts angezeigt.)



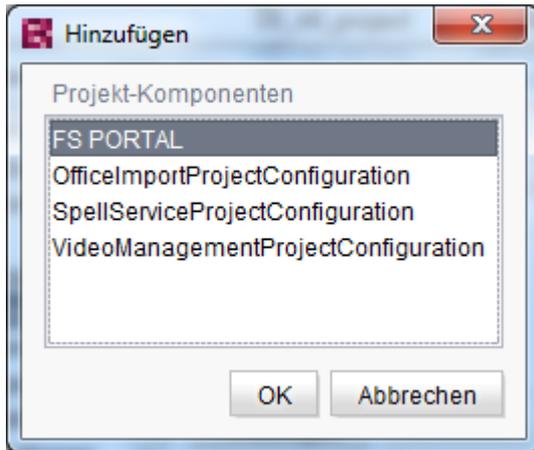


Abbildung 7-107: Hinzufügen einer Projekt-Komponente

Mit einem Klick auf den Löschen-Button können bereits zu einem Projekt hinzugefügte Komponenten (vom Typ "Projekt-Applikationen") entfernt werden.



Beim Löschen einer Projekt-Komponente gehen sämtliche damit verbundene Inhalte sowie die Konfiguration der Komponente im Projekt verloren.

Mit einem Klick auf den Button "Konfigurieren" kann die Konfiguration für eine zuvor hinzugefügte Projekt-Komponente bearbeitet werden (vgl. Kapitel 7.3.11 Seite 254). Die Konfiguration erfolgt abhängig von der Komponente entweder über eine von der Komponente erzeugte oder eine generische GUI.

Mit einem Klick auf den Button "Aktualisieren" kann die Komponente aktualisiert werden. Die Aktualisierung erfolgt gegen die aktuelle Version der Komponente auf dem FirstSpirit-Server. Ist dort eine aktuellere Version verfügbar, kann eine Aktualisierung für das jeweilige Projekt angestoßen werden. Aktualisierungen der Projekt-Komponenten können Anpassungen innerhalb des Projekts nach sich ziehen (z. B. Anpassung der Konfiguration).



7.4.18 Web-Komponenten

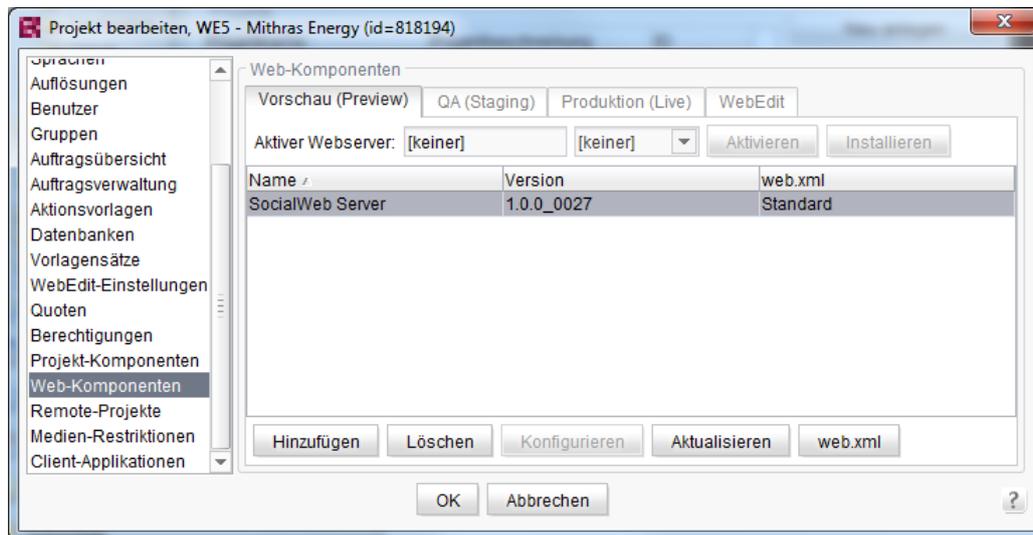


Abbildung 7-108: Projekteigenschaften – Web-Komponenten

In diesem Bereich können die Web-Komponenten für ein Projekt aktiviert werden.

Web-Bereiche:



Abbildung 7-109: Web-Bereiche innerhalb eines Projekts

Es existieren für jedes Projekt drei unterschiedliche Web-Bereiche. Über die jeweilige Registerkarte können die Web-Komponenten für jeden Bereich einzeln aktiviert und konfiguriert werden:

- Vorschau (Preview): Ort für die Projektinhalte für die eine Vorschau angefordert wurde.
- QA (Staging): Ort für die generierten Projektinhalte
- Produktion (Live): Ort für die veröffentlichten Projektinhalte
- WebEdit: Konfiguration für eine projektlokale WebEdit-Instanz (siehe Kapitel 5.2.2 Seite 182).

Bereich Webserver:

In jedem Web-Bereich können die Web-Komponenten für jedes Projekt konfiguriert werden. Im Dialog wird der momentan **aktive Webserver** für jeden Bereich angezeigt. Standardmäßig ist ein Eintrag für den "internalJetty" vorhanden. Es können aber auch weitere Webserver hinzugefügt werden. Zur Auswahl stehen alle Webserver, die im Bereich "Webserver" konfiguriert wurden (vgl. Kapitel 7.3.12 Seite



261). Über die danebenliegende Combobox kann ein anderer Webserver ausgewählt werden. Abhängig von der Art des Webserver sind zur Umstellung unterschiedliche Schritte erforderlich:

- Interner Webserver (siehe Kapitel 7.4.18.1 Seite 329)
- Generischer Webserver (siehe Kapitel 7.4.18.2 Seite 329)
- Externer Webserver (siehe Kapitel 7.4.18.3 Seite 330)
- Tomcat (siehe Kapitel 7.4.18.4 Seite 331)

Mit einem Klick auf den Button "Installieren" werden alle Web-Komponenten im jeweiligen Web-Bereich des Projekts in einer WAR-Datei zusammengefasst und abhängig vom konfigurierten Webserver installiert. Der Button ist aktiviert, wenn die Web-Komponente noch nicht installiert wurde und zur Installation bereit ist. Ist der Button deaktiviert, wurden die Web-Komponenten bereits installiert.

Handelt es sich beim ausgewählten Webserver, um einen externen Webserver oder um einen generischen Webserver (ohne die erforderliche Skript-Funktionalität), wird stattdessen der **Button "Download"** angezeigt (vgl. Kapitel 7.4.18.3).

Wurde die Webanwendung bereits installiert, anschließend aber die Konfiguration geändert (siehe Kapitel 7.4.18.7) oder eine Komponente hinzugefügt (siehe Kapitel 7.4.18.5) oder gelöscht (siehe Kapitel 7.4.18.6), wird statt des Buttons "Installieren", der **Button "Aktualisieren"** angezeigt.

Mit einem Klick auf den Button "Deinstallieren" werden die Web-Komponenten im jeweiligen Web-Bereich des Projekts wieder vom Webserver gelöscht. Diese Aktion wird für alle Web-Komponenten des jeweiligen Bereichs ausgeführt. Abhängig vom verwendeten Webserver wird die Deinstallation analog zur Installation einer Web-Komponente ausgeführt. Ist der Button deaktiviert, wurden die Web-Komponenten noch nicht installiert.

Mit einem Klick auf den Button "Download" kann eine WAR-Datei der Anwendung heruntergeladen werden, die manuell auf dem Webserver installiert werden muss (siehe Kapitel 7.3.13.5 Seite 273). Der Button wird nur für die Konfiguration externer Webserver oder die Konfiguration generischer Webserver (ohne die erforderliche Skript-Funktionalität) angezeigt.

Mit einem Klick auf den Button "Aktivieren" wird die Konfiguration des projektspezifischen Webbereichs auf den selektierten Webserver umgestellt, der anschließend als aktiver Webserver für diesen Bereich angezeigt wird.



7.4.18.1 Internen Webserver für eine Web-Applikation konfigurieren

Die Steuerung für den internen Webserver Jetty ist standardmäßig vorhanden und kann nicht verändert werden.

Wurde ein anderer Webserver für einen projektspezifischen Webbereich aktiviert, kann die Konfiguration über die folgenden Schritte wieder auf den internen Webserver zurückgesetzt werden:

1. In der Combobox den Eintrag "internalJetty" auswählen.
2. Der Button "Installieren" wird aktiv. Mit einem Klick auf den Button wird die WAR-Datei ins Zielverzeichnis des Webserver (Jetty) ausgepackt und anschließend werden die Web-Komponenten im Jetty direkt registriert.
3. Nach der Installation wird der Button "Aktivieren" aktiv. Mit einem Klick auf den Button wird die Konfiguration des Webbereichs auf den internen Webserver umgestellt, der nun als aktiver Webserver für den Bereich angezeigt wird.
4. Alle Änderungen der Konfiguration müssen mit einem Klick auf "OK" bestätigt und gespeichert werden.

7.4.18.2 Generischen Webserver für eine Web-Applikation konfigurieren

Die Auswahl eines generischen Webserver ist nur möglich, wenn eine entsprechende Webserver-Instanz zuvor auf dem Server hinzugefügt wurde (siehe Kapitel 7.3.12.1 Seite 263). Die Steuerung für den generischen Webserver ist nicht standardmäßig vorhanden, kann aber über Skripte realisiert werden (siehe Kapitel 7.3.12.2 Seite 263). Sind diese Skripte nicht vorhanden, ist das Vorgehen identisch zum externen Webserver (siehe Kapitel 7.4.18.3).

Soll für einen projektspezifischen Webbereich ein generischer Webserver aktiviert werden, müssen die folgenden Schritte ausgeführt werden:

1. In der Combobox den Eintrag für den gewünschten generischen Webserver auswählen.
2. Der Button "Installieren" wird aktiv. Sofern die entsprechende Funktionalität über ein Skript zur Verfügung gestellt wurde, kann mit einem Klick auf den Button die WAR-Datei kopiert und vom jeweiligen Webserver automatisch ausgepackt und die einzelnen Komponenten auf dem Webserver registriert werden (die entsprechende skriptbasierte Webserver-Steuerung muss zuvor innerhalb der "Server-Eigenschaften" konfiguriert werden (siehe Kapitel 7.3.15.2 Seite 154)).



3. Nach der Installation wird der Button "Aktivieren" aktiv. Mit einem Klick auf den Button wird die Konfiguration des Webbereichs auf den generischen Webserver umgestellt, der nun als aktiver Webserver für diesen Bereich angezeigt wird.
4. Alle Änderungen der Konfiguration müssen mit einem Klick auf "OK" bestätigt und gespeichert werden.

7.4.18.3 Externen Webserver für eine Web-Applikation konfigurieren

Die Auswahl eines externen Webservers ist nur möglich, wenn eine entsprechende Webserver-Instanz zuvor auf dem Server hinzugefügt wurde (siehe Kapitel 7.3.12.3 Seite 265). Die Steuerung für einen externen Webserver wird über FirstSpirit nicht unterstützt und muss manuell erfolgen (siehe Kapitel 7.3.12.4 Seite 266). Bei externen Webservern wird nur ein Download der WAR-Datei angeboten. Die Installation auf dem Webserver und die Registrierung der Web-Komponenten müssen manuell erfolgen.

Soll für einen projektspezifischen Webbereich ein externer Webserver aktiviert werden, müssen die folgenden Schritte ausgeführt werden:

1. In der Combobox den Eintrag für den gewünschten externen Webserver auswählen.
2. Der Button "Download" wird aktiv. Mit einem Klick auf den Button kann eine WAR-Datei der Anwendung heruntergeladen werden.
3. Die WAR-Datei muss manuell auf dem externen Webserver installiert werden. Die Installation erfolgt entweder manuell über die Verwaltungsoberfläche des externen Webservers oder automatisch aus dem Dateisystem des Webservers heraus
4. Nach der Installation kann mit einem Klick auf "Aktivieren" die Konfiguration des Webbereichs auf den externen Webserver umgestellt werden. Der externe Webserver wird nun als aktiver Webserver für die Anwendung angezeigt.
5. Alle Änderungen der Konfiguration müssen mit einem Klick auf "OK" bestätigt und gespeichert werden.



7.4.18.4 Tomcat-Webserver für eine Web-Applikation konfigurieren

Die Auswahl eines Tomcat-Webserver ist nur möglich, wenn eine entsprechende Webserver-Instanz zuvor auf dem Server hinzugefügt wurde (siehe Kapitel 7.3.12.5 Seite 267). Die Steuerung für einen Tomcat-Webserver wird von FirstSpirit unterstützt und erfolgt automatisch (siehe Kapitel 7.3.12.6 Seite 268). Die Installation auf dem Webserver und die Registrierung der Web-Komponenten erfolgt durch FirstSpirit.

Soll für einen projektspezifischen Webbereich ein Tomcat-Webserver aktiviert werden, müssen die folgenden Schritte ausgeführt werden:

1. In der Combobox den Eintrag für den gewünschten Tomcat-Webserver auswählen.
2. Der Button "Installieren" wird aktiv. Mit einem Klick auf den Button wird die WAR-Datei kopiert und vom Webserver automatisch ausgepackt, bevor die Web-Komponenten registriert werden.
3. Nach der Installation wird der Button "Aktivieren" aktiv. Mit einem Klick auf diesen Button wird die Konfiguration des Webbereichs auf den Tomcat-Webserver umgestellt, der nun als aktiver Webserver für diesen Bereich angezeigt wird.
4. Alle Änderungen der Konfiguration müssen mit einem Klick auf "OK" bestätigt und gespeichert werden.



Wurde die `fs-server.jar`-Datei aktualisiert, müssen die zuvor beschriebenen Schritte erneut für alle Web-Applikationen in allen Projekten, für die ein Webserver des Typs "Tomcat" ausgewählt wurde, erneut manuell ausgeführt werden. Es erfolgt kein automatisches Update.



7.4.18.5 Hinzufügen einer Web-Komponente

Hinzufügen: Mit einem Klick auf den Button öffnet sich der Dialog "Hinzufügen". In der Liste werden alle Web-Komponenten angezeigt, die auf dem Server installiert sind (siehe Kapitel 7.3.11 Seite 254).

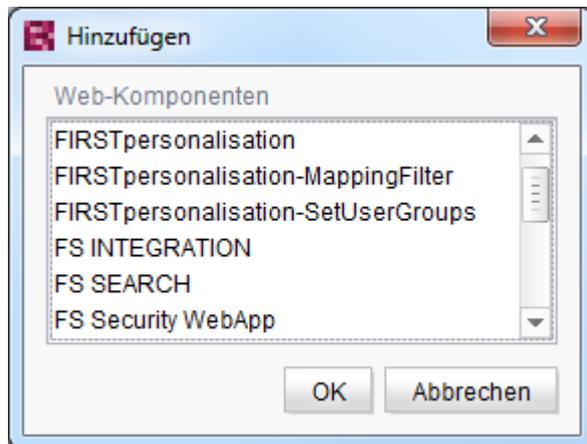


Abbildung 7-110: Hinzufügen einer Web-Komponente

Diese Web-Komponenten können den einzelnen Web-Bereichen (Preview, Staging, Live) innerhalb des gewünschten Projekts zugefügt werden. Anschließend besteht die Möglichkeit, diese Komponenten zu konfigurieren, entweder mit einer von der Komponente erzeugten oder einer generischen GUI (siehe Button "Konfigurieren"). Nach der Konfiguration müssen die Komponenten noch aktiviert werden. Dabei kann eine Komponente innerhalb eines Projekts nur für bestimmte Bereiche aktiviert bzw. deaktiviert werden (siehe "Installation").

7.4.18.6 Löschen einer Web-Komponente

Mit einem Klick auf den Button "Löschen" können zuvor hinzugefügte Komponenten wieder entfernt werden. Diese Komponenten werden in der Tabelle nicht mehr angezeigt und sind bei der nächsten Veröffentlichung kein Bestandteil der WAR-Datei mehr. Damit die Änderungen wirksam werden, muss eine Veröffentlichung auf dem Webserver erfolgen (siehe "Installation").



7.4.18.7 Konfigurieren einer Web-Komponente

Mit einem Klick auf den Button "Konfigurieren" kann die Konfiguration für eine zuvor hinzugefügte Komponente bearbeitet werden (vgl. Kapitel 7.3.11 Seite 254). Die Konfiguration erfolgt abhängig von der Komponente entweder über eine von der Komponente erzeugte oder eine generische GUI. Für FirstSpirit Personalisation sieht der Konfigurations-Dialog folgendermaßen aus:

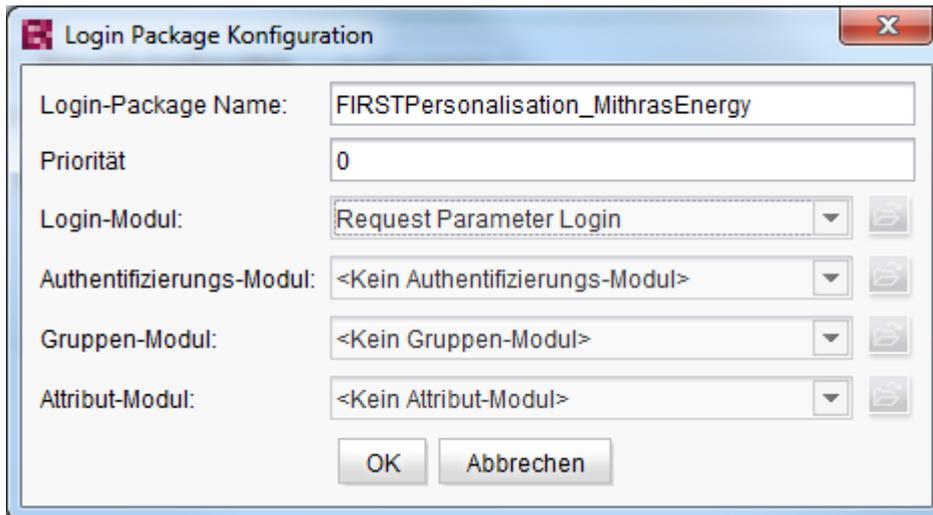


Abbildung 7-111: Konfiguration einer weblokalen Komponente (Beispiel)

7.4.18.8 Aktualisieren einer Web-Komponente

Aktualisieren: Mit einem Klick auf den Button kann die Komponente aktualisiert werden. Die Aktualisierung erfolgt gegen die aktuelle Version der Komponente auf dem FirstSpirit-Server. Ist dort eine aktuellere Version verfügbar wird die Aktualisierung für das jeweilige Projekt angestoßen. Aktualisierungen der Web-Komponenten für ein Projekt können Anpassungen innerhalb des Projekts nach sich ziehen. So kann es beispielsweise notwendig werden, die Projektkonfiguration anzupassen.

7.4.18.9 web.xml bearbeiten

Werden ein oder mehrere Web-Komponenten für einen Web-Bereich konfiguriert, wird automatisch eine Datei web.xml erstellt, die aus den einzelnen web.xmls der jeweiligen Komponenten besteht. Die Datei web.xml kann manuell editiert werden. Mit einem Klick auf den Button "web.xml" öffnet sich das Dialog-Fenster zur manuellen Konfiguration der Datei. Nach dem Speichern der Änderungen wird in der



Übersicht in der Spalte "web.xml" nicht mehr der Wert "Standard", sondern der Wert "Editiert" angezeigt.



Abbildung 7-112: Nach manuellem Editieren der web.xml

7.4.19 Remote-Projekte

Hier werden alle Remote-Projekte aufgelistet, die für das Projekt konfiguriert worden sind. Remote-Projekte sind Projekte, aus denen Medien bzw. Daten referenziert oder auch geladen werden können.

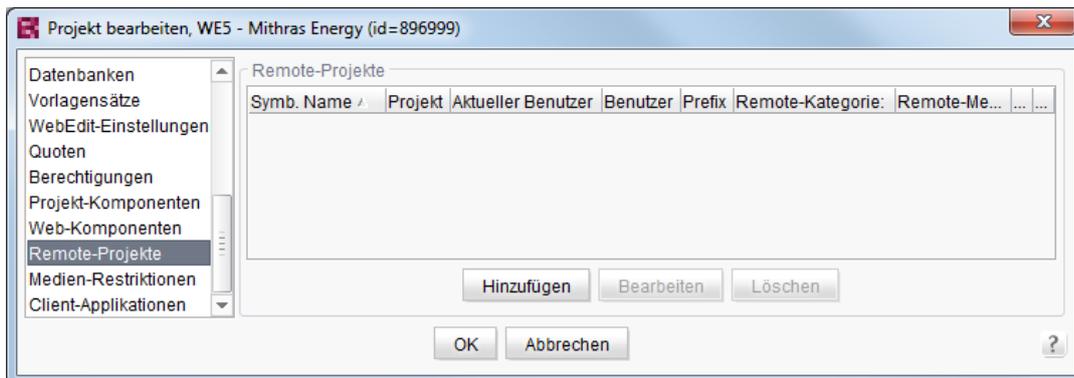


Abbildung 7-113: Projekteigenschaften – Remote-Projekte

Der Button **Hinzufügen** öffnet ein Fenster, in dem ein neues Remote-Projekt konfiguriert werden kann. Bei einem bestehenden Remote-Projekt öffnet der Button **Bearbeiten** dasselbe Fenster.



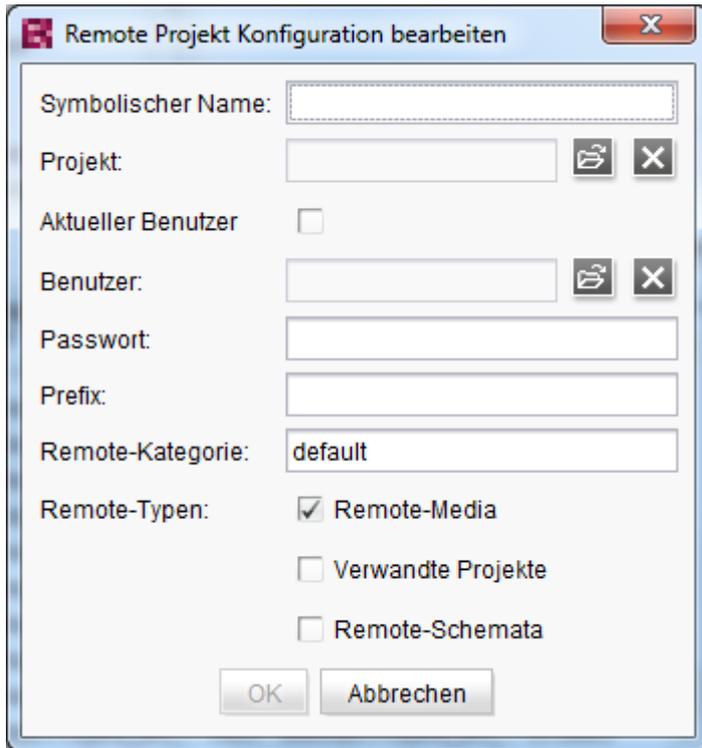


Abbildung 7-114: Konfiguration eines Remote-Projektes

Symbolischer Name: In diesem Feld wird ein eindeutiger Name für das Remote-Projekt festgelegt. Über diesen Namen kann im Zielprojekt das gewünschte Remote-Projekt referenziert werden.



Als Remote-Projekte können nur Projekte konfiguriert werden, die auf einem Server mit dem Zielprojekt liegen.

Projekt: Über das Dateisymbol kann aus der Projektauswahlliste des Servers das gewünschte Projekt ausgewählt werden, aus dem Medien/Daten referenziert werden sollen.

Aktueller Benutzer: Ist diese Option aktiviert, dann wird beim Zugriff auf das Remote-Projekt überprüft, ob der aktuelle Benutzer die nötigen Rechte in diesem Projekt besitzt.

Benutzer: Über das Dateisymbol kann ein Benutzer aus der Benutzerliste des Servers ausgewählt werden, über dessen Rechte der Zugriff auf das Remote-Projekt geregelt werden soll.

Passwort: In diesem Feld muss das Passwort des ausgewählten Benutzers



angegeben werden.



Der technische Benutzer muss im Zielprojekt als Benutzer bekannt sein und innerhalb der Medien-Verwaltung des Remote-Projekts mindestens das Recht "sichtbar" auf der gesamten Medien-Verwaltung oder auf einzelnen Ordnern besitzen.

Prefix: In diesem Feld muss noch einmal der Name des Remote-Projektes eingetragen werden. Das Präfix wird bei der Generierung des Zielprojekts für die URL-Erzeugung benötigt.

Remote-Kategorie: In diesem Feld wird eine Kategoriebezeichnung für eine Remote-Projektconfiguration vergeben. Dadurch können mehrere Remote-Projekte zu einer Gruppe zusammengefasst werden. Die Kategorie kann in einer Verweiskonfiguration verwendet werden, um aus der festgelegten Gruppe von Remote-Projekten auswählen zu können. Standardmäßig ist das Feld mit dem Wert "default" (Standardkategorie) vorbelegt.

Remote-Typen: Es sind drei unterschiedliche Arten des Remote-Zugriffs möglich:

Remote-Media	(siehe Kapitel 7.4.19.1 Seite 336)
Verwandte Projekte	(siehe Kapitel 7.4.19.2 Seite 337)
Remote-Schemata	(siehe Kapitel 7.4.19.3 Seite 337)



Bei allen Arten von Remote-Zugriffen handelt es sich um lizenzabhängige Zusatzfunktionen.

7.4.19.1 Konzept Remote-Media

Ziel des Konzepts "Remote Media" ist, alle Medien in einem getrennten Medienprojekt anzulegen und dort zentral zu verwalten. Über den Remote-Media-Zugriff können anschließend alle beteiligten FirstSpirit-Projekte auf den Medienbestand (Bilder und Dateien) dieses Projekts zugreifen.

Im Unterschied zur Verteilung der Medien über die lizenzabhängige Funktion "CorporateContent" müssen die Medien nicht in die beteiligten Projekte importiert werden, sondern können direkt über den Remote-Media-Zugriff referenziert werden. Die Objekte verbleiben dabei physikalisch im Medienprojekt, können aber in allen



gewünschten Projekten verwendet werden.

Sofern eine gültige Lizenz für die Funktionalität besteht, kann der Remote-Media-Zugriff über die FirstSpirit Server- und Projektkonfiguration aktiviert werden.

Weitere Informationen zur Konfiguration und Verwendung von Remote-Medien siehe Dokumentation zur Funktionalität "FirstSpirit Corporate-Media", FirstSpirit Handbuch für Redakteure und FirstSpirit Online Dokumentation.

7.4.19.2 Konzept Verwandte Projekte

Das Konzept "Verwandte Projekte" behandelt Verknüpfungen von einem Projekt in ein weiteres FirstSpirit-Projekt. Diese Verknüpfungen können indirekt über die Definition von Verweiszielen in ein FirstSpirit-Projekt realisiert werden (über speziell konfigurierte Eingabekomponenten), als auch über eine direkte Definition innerhalb der Struktur-Verwaltung.

Sofern eine gültige Lizenz für die Funktionalität besteht, kann der "Verwandte Projekte"-Zugriff über die FirstSpirit Server- und Projektkonfiguration aktiviert werden.

Weitere Informationen zur Konfiguration und Verwendung von verwandten Projekten siehe FirstSpirit Handbuch für Redakteure und FirstSpirit Online Dokumentation.

7.4.19.3 Konzept Remote-Schemata

Das Konzept "Remote-Schemata" wird für das lizenzabhängige FirstSpirit Modul "FirstSpirit DynamicDatabaseAccess" benötigt, das zur Anbindung unterschiedlicher Datenbank-Technologien an FirstSpirit dient. Mithilfe des Moduls können Inhalte aus einer Datenbank über eine Webanwendung dargestellt und editiert werden. FirstSpirit DynamicDatabaseAccess verwendet Datenbank-Schemata aus der FirstSpirit Vorlagenverwaltung. Diese Schemata können innerhalb eines Projekts neu definiert werden. Für die Bearbeitung eines Datenbank-Schemas steht ein grafischer Editor zur Verfügung, mit dessen Hilfe das gewünschte Datenbankschema erstellt werden kann. Jedes Schema kann dabei auf bestehende Datenbankstrukturen zurückgreifen oder neue Tabellenstrukturen in einer bestehenden Datenbank anlegen (weiterführende Informationen zur Verwendung von Datenbank-Schemata siehe "FirstSpirit Handbuch für Entwickler (Grundlagen)" oder "FirstSpirit Online Dokumentation").

Der Remote-Zugriff auf Datenbank-Schemata aus weiteren FirstSpirit-Projekten ist ebenfalls möglich (Remote-Projekte), wenn innerhalb der Remote-Projekt-



Konfiguration der Remote-Typ "Remote-Schemata" aktiviert wurde.

Weitere Informationen zu FirstSpirit DynamicDatabaseAccess siehe die entsprechende Modul-Dokumentation.

7.4.20 Medien-Restriktionen

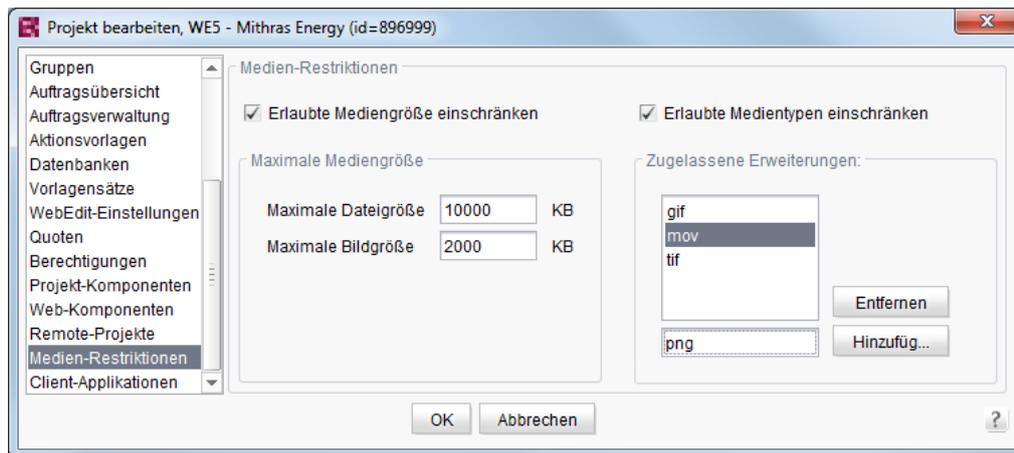


Abbildung 7-115: Projekteigenschaften – Medien-Restriktionen

Der Upload von Medien in die Medien-Verwaltung kann über die Optionen "Erlaubte Mediengröße einschränken" und "Erlaubte Medientypen einschränken" auf bestimmte Dateigrößen und/oder -formate begrenzt werden. Beide Optionen können jeweils einzeln aktiviert oder miteinander kombiniert werden.



Die Restriktionen werden nur beim Upload neuer Medien angewendet, nachträgliche Restriktionen wirken sich nicht auf bereits angelegte Medien aus.

7.4.20.1 Erlaubte Mediengröße einschränken

Erlaubte Mediengröße einschränken: Wird die Checkbox aktiviert, werden die Eingabefelder zur Angabe einer maximalen Dateigröße bzw. einer maximalen Bildgröße aktiv.

Maximale Dateigröße: In diesem Feld kann die maximale Dateigröße (in KB) für den Upload von Medien vom Typ "Datei" in die Medien-Verwaltung definiert werden. Dateien, die größer sind als der angegebene Maximalwert, können nicht mehr angelegt werden.



Maximale Bildgröße: In diesem Feld kann die maximale Dateigröße (in KB) für den Upload von Medien vom Typ "Bild" in die Medien-Verwaltung definiert werden. Bilder, die größer sind als der angegebene Maximalwert, können nicht mehr angelegt werden.

Für beide Felder sind nur ganzzahlige Werte zulässig. Die Eingabe von anderen Werten wie z. B. Buchstaben oder Zeichen wird direkt unterbunden und das jeweilige Feld blinkt rot auf.

7.4.20.2 Erlaubte Medientypen einschränken

In der rechten Hälfte der Medien-Beschränkungen unter der Auswahlbox **Erlaubte Medientypen einschränken** der Konfigurationsoberfläche gibt es die Option, das Dateiformat für hochladbare Medien einzuschränken. Dies geschieht über die Eingabe von Dateinamen-Erweiterungen.

Zugelassene Erweiterungen: Hier können die zulässigen Dateinamen-Erweiterungen hinzugefügt bzw. entfernt werden. Die Überprüfung des Dateiformates erfolgt nur über die Endung des Dateinamens und nicht über eine Analyse des Dateiinhaltes. Alle Erweiterungen müssen hier ohne "." angegeben werden. Dabei wird Groß- und Kleinschreibung berücksichtigt, d.h. bei einer Einschränkung auf den Medientyp DOC wird später nur der Upload von Dateien mit der Endung .DOC, nicht aber mit der Endung .doc unterbunden. Im Zweifelsfall sollten verschiedene Schreibweisen angegeben werden, um alle Dateien des gewünschten Formates abdecken zu können.

Über die Schaltfläche "**Hinzufügen**" wird eine zuvor in dem Eingabefeld links neben der Schaltfläche "Hinzufügen" definierte Erweiterung der Liste für die zugelassenen Erweiterungen hinzugefügt und gespeichert. Die Erweiterungen werden in alphabetischer Sortierung aufgelistet.

Die Schaltfläche "**Entfernen**" löscht eine zuvor aus der Liste selektierte Dateinamen-Erweiterung.

7.4.20.3 Darstellung der Restriktionen im JavaClient

Für den Redakteur werden die Medien-Beschränkungen, sofern in der Projektkonfiguration aktiviert, im Dateiauswahldialog des JavaClients in Form einer Filter-Beschränkung dargestellt. Dies gilt auch für den Medien-Assistenten. Medien, die größer als die definierte maximale Größe sind und/oder eine nicht erlaubte Dateinamen-Erweiterung haben, stehen im Dateiauswahldialog nicht zur Auswahl.



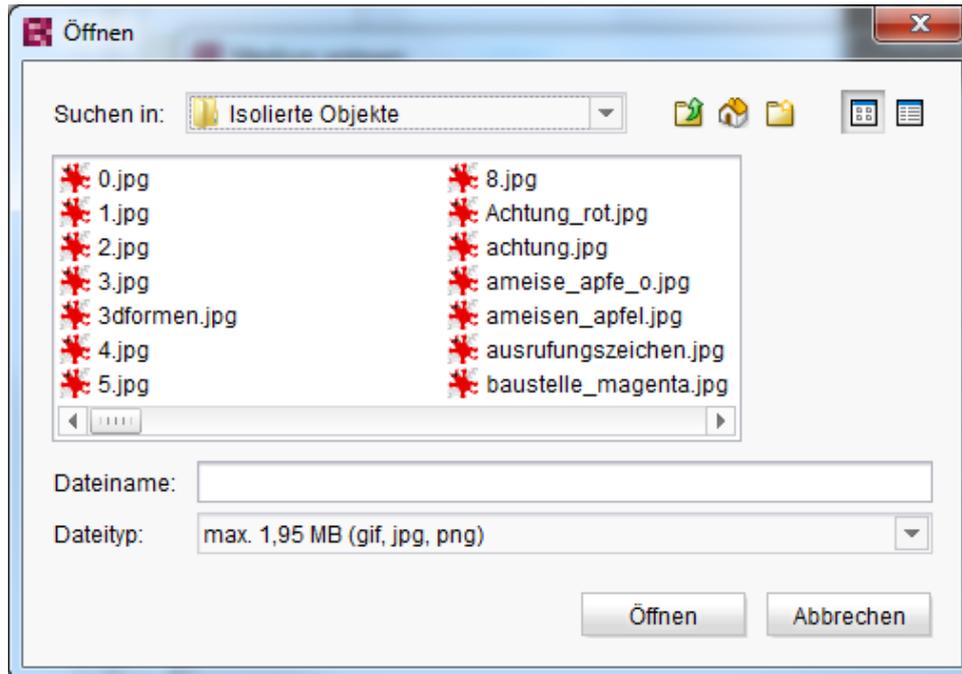


Abbildung 7-116: Aktivierte Beschränkung auf Dateigrößen und Dateinamen-Erweiterungen

Rechts neben "Dateityp:" werden die Restriktionen noch einmal angezeigt. Im Beispiel in Abbildung 7-116 dürfen Dateien mit den Dateinamen-Erweiterungen "jpg", "png" und "gif" bis zu einer Größe von 1,95 MB ausgewählt und hochgeladen werden.



Sollen Dateien per ‚Drag & Drop‘ aus der Verzeichnisstruktur des Arbeitsplatzrechners in die Medien-Verwaltung eingefügt werden, die aufgrund der Medien-Beschränkungen nicht zugelassen sind, wird ein entsprechender Warnhinweis angezeigt. Die Dateien werden nicht in der Medien-Verwaltung angelegt.

7.4.20.4 Aktivierte Beschränkungen im WebClient

Im WebClient ist die technische Umsetzung der Filterregeln nicht wie im JavaClient möglich. Da der im Browser zur Verfügung stehende Dateiauswahldialog zum Upload von Medien keine FirstSpirit-eigene Implementierung ist, sondern in jedem Browser (z. B. Firefox, Mozilla, Internet Explorer, Opera) fest integriert ist, ist eine Filterung wie im JavaClient technisch nicht möglich. Somit werden die Dateien erst nach dem Upload gefiltert und ggf. eine Fehlermeldung bei Überschreitung der in der Projektkonfiguration definierten Medien-Restriktionen an den Benutzer ausgegeben.



7.4.21 Client-Applikationen

Die hier vorgenommenen Einstellungen haben Einfluss auf das Menü "Ansicht" im JavaClient. Damit die Änderungen übernommen werden, muss das betreffende Projekt neu gestartet werden.

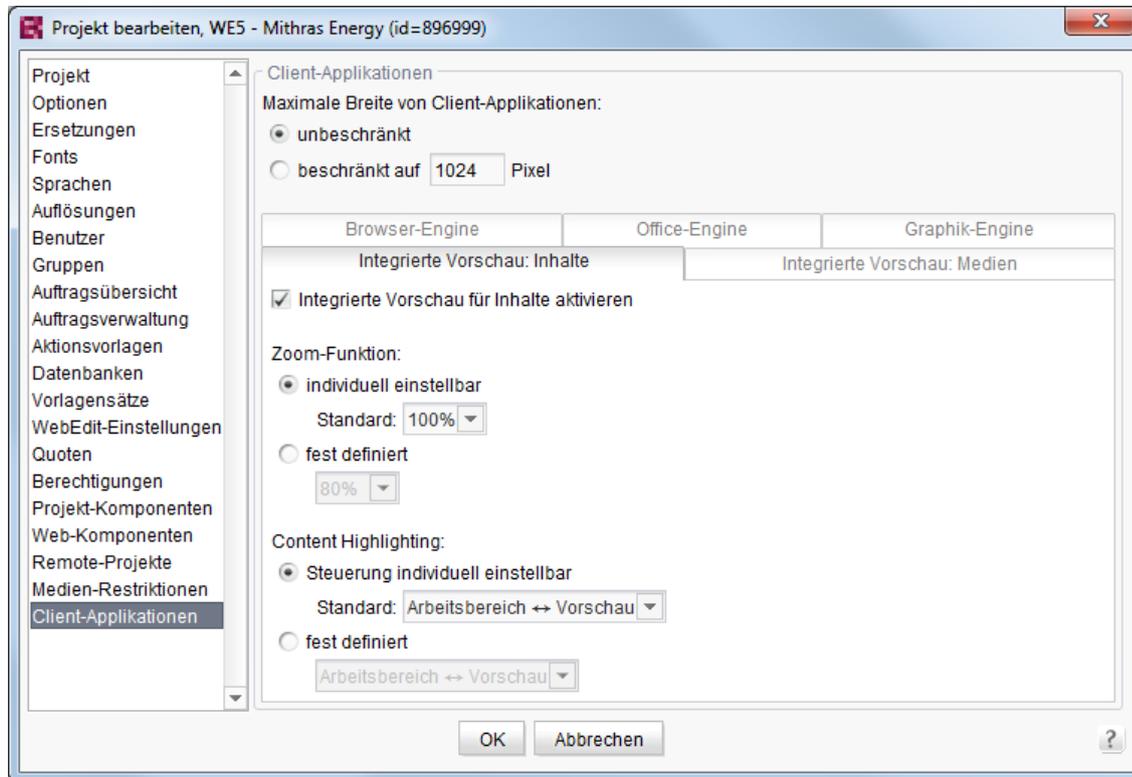


Abbildung 7-117: Projekteigenschaften – Client-Applikationen

Maximale Breite von Client-Applikationen: Hier kann festgelegt werden, ob die Breite auf eine bestimmte Pixel-Anzahl beschränkt werden, oder vom Benutzer beliebig wählbar sein soll.

7.4.21.1 Integrierte Vorschau: Inhalte

Integrierte Vorschau für Inhalte aktivieren: Diese Option hängt mit der Option "Browser Engine aktivieren" auf dem Register "Browser-Engine" zusammen: Es können nur beide Optionen gleichzeitig aktiviert oder deaktiviert sein. Ist diese Option deaktiviert, können auf diesem Register keine Änderungen vorgenommen werden.

Ist diese Option aktiviert, kann im JavaClient für das gewählte Projekt die integrierte Vorschau für Inhalte genutzt werden, das heißt die Redakteure können individuell über die Menüfunktion "Ansicht" / "Integrierte Vorschau – für Inhalte verwenden"



(siehe *FirstSpirit Handbuch für Redakteure (JavaClient)*) auswählen, ob sie mit der integrierten Vorschau arbeiten möchten oder nicht. Ist die Option deaktiviert, steht im JavaClient projektweit die Menüfunktion "Ansicht" / "Integrierte Vorschau – für Inhalte verwenden" nicht zur Auswahl zur Verfügung. Standardmäßig ist diese Option aktiviert.

Zoom-Funktion

individuell einstellbar: Ist diese Option aktiviert, können Benutzer im betreffenden Projekt die Skalierungsstufe selbst auswählen. Über die Combobox "Standard" kann vorgegeben werden, welche Skalierungsstufe vorausgewählt sein soll. Standardmäßig ist dies 80%.

fest definiert: Ist diese Option aktiviert, wird die gewählte Skalierungsstufe projektweit verwendet. Die Skalierung kann nicht durch die Redakteure geändert werden.

Content Highlighting

Steuerung individuell einstellbar: Ist diese Option aktiviert, können die Redakteure das Verhalten des Content Highlighting über den Menüpunkt "Ansicht" / "Steuerung Content Highlighting" (siehe *FirstSpirit Handbuch für Redakteure (JavaClient)*) individuell einstellen. Über die Combobox "Standard" kann ausgewählt werden, welches Verhalten vorausgewählt sein soll. Standardmäßig ist dies "Arbeitsbereich ↔ Vorschau"

fest definiert: Ist diese Option aktiviert, wird das gewählte Verhalten projektweit verwendet. Das Verhalten kann nicht durch die Redakteure geändert werden.

7.4.21.2 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:



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

Integrierte Vorschau für Medien aktivieren

Office-Engine verwenden für folgende Datei-Endungen (kommasepariert):

Browser-Engine verwenden für folgende Datei-Endungen (kommasepariert):

Integrierten Text-Editor verwenden für folgende Datei-Endungen (kommasepariert):

Integrierte Bild-Ansicht verwenden für folgende Datei-Endungen (kommasepariert):

Windows Media Player (nur Windows) verwenden für folgende Datei-Endungen (kommasepariert):

Default-Werte wiederherstellen

Abbildung 7-118: Voreinstellung Integrierte Vorschau/Medien (Default-Werte)

Integrierte Vorschau für Medien aktivieren: Diese Option hängt mit den Optionen "Office Engine aktivieren" auf dem Register "Office-Engine" und "Grafik-Engine aktivieren" auf dem Register "Grafik-Engine" zusammen: Wird eine der anderen Optionen aktiviert, dann wird diese Option ebenfalls aktiv. Ist diese Option deaktiviert, können auf diesem Register keine Änderungen vorgenommen werden. Ist diese Option aktiviert, kann im JavaClient für das gewählte Projekt die integrierte Vorschau für Medien genutzt werden, das heißt die Redakteure können individuell über die Menüfunktion "Ansicht" / "Integrierte Vorschau – für Medien verwenden" (siehe *FirstSpirit Handbuch für Redakteure (JavaClient)*) auswählen, ob sie mit der integrierten Vorschau arbeiten möchten oder nicht. Ist die Option deaktiviert, steht im JavaClient projektweit die Menüfunktion "Ansicht" / "Integrierte Vorschau anzeigen – für Medien" nicht zur Auswahl zur Verfügung.

In neuen Projekten 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.

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 Webbrowsern bzw. von Plug-ins 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.

Default-Werte wiederherstellen: Mit einem Klick auf diese Schaltfläche kann die Standard-Einstellung wiederhergestellt werden.

7.4.21.3 Browser-Engine



Abbildung 7-119: Projekteigenschaften – Voreinstellung Integrierte Vorschau/Browser

Browser-Engine aktivieren: Diese Option hängt mit der Option "Integrierte Vorschau für Inhalte aktivieren" auf dem Register "Integrierte Vorschau: Inhalte" zusammen: Es können nur beide Optionen gleichzeitig aktiviert oder deaktiviert sein. Ist diese Option deaktiviert, können auf diesem Register keine Änderungen vorgenommen werden.

Die Dateiformate, die auf dem Register "Integrierte Vorschau: Medien" im Feld



"Browser-Engine" (siehe Kapitel 7.4.21.2 Seite 342) 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 *FirstSpirit TechnischesDatenblatt Version 5.0*, zum Ausrollprozess der nativen Bestandteile der Browserintegration siehe Kapitel 4.9 Seite 175.

7.4.21.4 Office-Engine



Abbildung 7-120: 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. 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 7.4.21.2 Seite 342) 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 7.4.21.2 Seite 342) 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.

Hinweis zur Verwendung von OpenOffice: Um OpenOffice verwenden zu können, muss OpenOffice direkt in den Standard-Programme-Verzeichnissen des jeweiligen Betriebssystems installiert sein (nicht in einem Unterverzeichnis!), unter Microsoft Windows z. B. unter "C:\Programme" oder "C:\Programme (x86)" bzw. "C:\Program Files" oder "C:\Program Files (x86)", unter Linux z. B. "/usr/lib", "/usr/local/lib" oder "/opt". Ansonsten öffnen sich OpenOffice-Dokumente möglicherweise nicht in der Integrierten Vorschau sondern in einem separaten Fenster.



Für die Verwendung von Anwendungen in der Integrierten Vorschau ist grundsätzlich zu beachten, dass FirstSpirit die erforderlichen Schnittstellen für die Applikationsintegration 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.





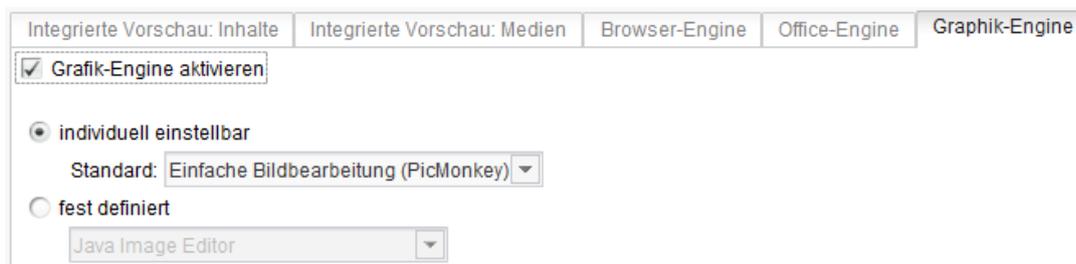
*Die gleichzeitige Verwendung der Integrierten Vorschau für Medien mit Microsoft Office und extern als Microsoft Office-Desktop-Anwendung (z.B. gestartet über Windows oder aus dem JavaClient heraus) kann unter bestimmten Umständen zu Problemen führen und ist keine zugesicherte Produkteigenschaft der 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.*

Hinweise zur Verwendung von 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 nicht freigegebenen API führen dazu, dass die Google Docs-Integration in FirstSpirit Version 5.0 auch weiterhin keine zugesicherte Produkteigenschaft ist 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. Aktuelle kann Google Docs im JavaClient nur mit der Browser Engine "Internet Explorer" verwendet werden.



7.4.21.5 Graphik-Engine

**Abbildung 7-121: 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 7.4.21.2 Seite 341) 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.



Für die Verwendung von Anwendungen in der Integrierten Vorschau ist grundsätzlich zu beachten, dass FirstSpirit die erforderlichen Schnittstellen für die Applikationsintegration 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.





Für die Nutzung des **Java Image Editors** erfolgt die Freigabe 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". Wenn die Nutzung von Bildbearbeitungsfunktionen eine produktionskritische Funktion ist, sollte eine externe Bildbearbeitungssoftware mit dem entsprechenden Hersteller-Support verwendet werden. Unter MacOS steht der Java Image Editor aktuell nur sehr eingeschränkt zur Verfügung.



Zur Verwendung der Optionen **PicMonkey** oder **Pixlr** ist eine aktive Verbindung zum Internet erforderlich. Die zu bearbeitenden Bilder werden faktisch auf dem Server des jeweiligen Anbieters hochgeladen und dort bearbeitet. Dies sollte bei der Verwendung im Hinblick auf den Datenschutz berücksichtigt werden

7.5 Auftragsplanung

Über die FirstSpirit Auftragsplanung können zusammengehörige Aktionen zu einem Auftrag zusammengefasst und zu definierten geplanten Zeitpunkten gestartet werden.

Die FirstSpirit Server- und Projektkonfiguration bietet die Möglichkeit bestehende Aufträge und die zugehörigen Informationen anzuzeigen. Innerhalb der Server- oder den Projekteigenschaften kann dazu die tabellarische "Auftragsübersicht" aufgerufen werden (siehe Kapitel 7.5.1).

Über den Menüpunkt "Auftragsverwaltung" können neue Aufträge hinzugefügt und bestehende verwaltet werden (siehe Kapitel 7.5.2). Das Anlegen neuer Aufträge ist unterteilt in die Konfiguration der Auftragseigenschaften (siehe Kapitel 7.5.4), also beispielsweise die Angabe eines Ausführungszeitpunkts, und das Hinzufügen der gewünschten Aktionen, die im Rahmen des Auftrags durchlaufen werden sollen (siehe Kapitel 7.5.5).

Die Aktionen müssen zuvor über die Server- und Projektkonfiguration angelegt werden (siehe Kapitel 7.5.6). Serverbezogene Aktionen werden innerhalb der Servereigenschaften definiert (siehe Kapitel 7.5.6.2), projektbezogene Aktionen innerhalb der Projekteigenschaften der einzelnen Projekte (siehe Kapitel 7.5.6.1).



7.5.1 Auftragsübersicht

Startzeitpunkt	Typ	Auftrag	Status	Letzte Dau...	Dauer	Auftr...	Ausf...	Benut...
30.08.2012...	Regelmäßig	Teil-Deployment ...	Erfolgreich	0:03:28.614	0:01:13.536	31557	90501	SYSTE
30.08.2012...	Regelmäßig	Teil-Deployment ...	Erfolgreich	0:03:08.261	0:01:13.536	31557	90499	SYSTE
30.08.2012...	Regelmäßig	Teil-Deployment ...	Erfolgreich	0:02:48.688	0:01:13.536	31557	90497	SYSTE
30.08.2012...	Regelmäßig	Teil-Deployment ...	Erfolgreich	0:02:48.824	0:01:13.536	31557	90493	SYSTE
30.08.2012...	Regelmäßig	Teil-Deployment ...	Erfolgreich	0:03:49.282	0:01:13.536	31557	90486	SYSTE
30.08.2012...	Einmalig	hotdeployment	Erfolgreich	0:00:10.528	0:00:01.131	31555	90488	deplo
30.08.2012...	Einmalig	hotdeployment	Erfolgreich	0:00:10.076	0:00:01.131	31555	90487	deplo
30.08.2012...	Regelmäßig	Teil-Deployment ...	Erfolgreich	0:01:33.779	0:01:13.536	31557	90478	SYSTE
30.08.2012...	Einmalig	hotdeployment	Erfolgreich	0:00:16.251	0:00:01.131	31555	90485	deplo
30.08.2012...	Einmalig	hotdeployment	Erfolgreich	0:00:14.375	0:00:01.131	31555	90482	deplo

Abbildung 7-122: Auftragsübersicht

Die tabellarische Auftragsübersicht enthält historische Daten bis zum aktuellen Datum und berechnete Daten für zukünftig anstehende, aktive (nicht manuelle) Aufträge (siehe Kapitel 7.5.1.2 Seite 352). Je nach dem, ob man sich in den Servereigenschaften (siehe Kapitel 7.3.9) oder den Projekteigenschaften (siehe Kapitel 7.4.9) befindet, werden entweder nur server- oder nur projektabhängige Aufträge angezeigt.

Standardmäßig werden Aufträge im Zeitraum von 5 Tagen vor dem aktuellen Datum bis 5 Tagen nach dem aktuellen Datum angezeigt. Es besteht die Möglichkeit, sich alle Daten der laufenden Woche, des laufenden Monats oder eines definierten Zeitraums anzeigen zu lassen (siehe Kapitel 7.5.1.1 Seite 351).

Durch einen Doppelklick auf den gewünschten Auftrag wird der Dialog mit den Auftrags-Details geöffnet (siehe Kapitel 7.5.1.3 Seite 353).

7.5.1.1 Übersichtszeitraum definieren

Optionen – Woche: Ist diese Option *aktiviert*, werden alle Aufträge aufgelistet, welche in der aktuellen Woche (Montag – Sonntag) ausgeführt wurden bzw. noch ausgeführt werden sollen.



Optionen – Monat: Ist diese Option *aktiviert*, werden alle Aufträge aufgelistet, welche im laufenden Monat ausgeführt wurden bzw. noch ausgeführt werden sollen.

Optionen – von ... bis: Ist diese Option *aktiviert*, werden alle Aufträge aufgelistet, welche innerhalb des definierten Zeitraums ausgeführt wurden bzw. noch ausgeführt werden sollen. Das Anfangs- und Enddatum kann jeweils durch einen Klick auf das Kalendersymbol ausgewählt werden.

Wurde eine neue Option bzw. ein anderes Anfangs- oder Enddatum ausgewählt, muss die Auswahl per Klick auf den Button "Anzeigen" bestätigt werden. Erst dann wird die Tabelle mit den angeforderten Daten aktualisiert.

7.5.1.2 Tabellarische Auftragsübersicht

Die Tabelle verfügt über folgende Spalten:

Startzeitpunkt: Der Zeitpunkt, zu dem die Ausführung des Auftrags gestartet wurde bzw. die Ausführung geplant ist.

Typ: Es existieren unterschiedliche Ausführungsarten für Aufträge (siehe Seite 360).

- manuell: manuell auszuführender Auftrag
- Automatisch auszuführende Aufträge unterscheidet man wie folgt:
 - Einmalig
 - Regelmäßig

Auftrag: Eindeutiger Name des Auftrags. Der Name wird beim Hinzufügen eines neuen Auftrags definiert.

Status: Diese Spalte beschreibt den Ausführungsstatus des Auftrags. Es existieren folgende Status:

- Nicht gestartet
- Wird ausgeführt...
- Abgebrochen
- Fehler
- Beendet, mit Fehlern
- Erfolgreich

Abhängig vom Status werden die Zeilen der Tabelle mit unterschiedlichen Farben hinterlegt:

- grün: Erfolgreich
- rot: Fehler
- gelb: Zeitkonflikt



Letzte Dauer: Hier wird der Zeitraum der letzten Auftragsausführung angezeigt.

Ø Dauer: Die durchschnittliche Dauer gehört, wie auch die "letzte Dauer" zu den statistischen Informationen, die zur zukünftigen Zeitplanung der Aufträge sinnvoll sein können.

7.5.1.3 Details zum Auftrag



Abbildung 7-123: Details zum Auftrag

Durch einen Doppelklick auf einen Auftrag aus der Auftragsübersicht (siehe Abbildung 7-122) wird ein Dialog mit Detailinformationen geöffnet. Dieser Dialog zeigt den Status des ausgewählten Auftrags, sowie die zugehörigen Aktionen.

Je nach Ausführungsstatus werden die Zeilen der Tabelle mit unterschiedlichen Farben hinterlegt (Farb-Legende: siehe Kapitel 7.5.1.2 Seite 352).

Der Button "Bearbeiten" ist nur aktiviert, wenn der Auftrag den Status "Nicht gestartet" hat und der eingeloggte Benutzer Zugriffsrechte auf das entsprechende Projekt besitzt. Durch einen Klick auf diesen Button öffnet sich der Bearbeitungsdialog zum Auftrag (siehe Kapitel 7.5.4 Seite 360).

Durch einen Klick auf den Button "Auftrags-Logfile anzeigen" öffnet sich die Log-Datei dieses Auftrags.

Aktion: Der Name der ausgeführten bzw. auszuführenden Aktion.

Dauer: Zeitraum der letzten Ausführung der Aktion. Wurde die Aktion noch nicht ausgeführt, bleibt die Zelle leer.



Fehler / Warnungen: Hier wird die Anzahl der Fehler und Warnungen angezeigt, die bei der Ausführung aufgetreten sind.

Status: Diese Spalte beschreibt den Ausführungsstatus des Auftrags (vgl. Kapitel 7.5.1.2 Seite 352).

Durch Klick auf den Button "Aktions-Logfile" öffnet sich die Log-Datei der ausgewählten Aktion. Die Log-Datei kann alternativ auch per Doppelklick auf der gewünschten Aktion aufgerufen werden.



Aufträge oder zum Auftrag zugehörige Aktionen, die sich im Status "Nicht gestartet" befinden, besitzen noch keine Log-Datei.

7.5.2 Auftragsverwaltung

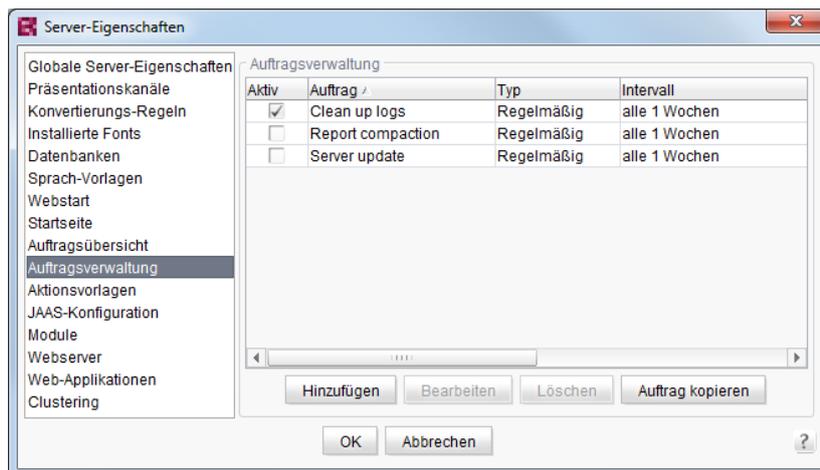


Abbildung 7-124: Auftragsverwaltung

Die Auftragsverwaltung enthält alle Aufträge, die (je nach dem, ob man sich in den Server- oder Projekteigenschaften befindet) entweder server- (siehe Kapitel 7.3.9) oder projektabhängig (siehe Kapitel 7.4.9) angelegt wurden.



Einige Änderungen an Aufträgen werden sofort gespeichert, ohne dass sie mit OK bestätigt werden müssen. Eine solche Speicherung wird auch dann nicht zurückgesetzt, wenn der Dialog über "Abbrechen" geschlossen wird.



Aktiv: Ist diese Option *deaktiviert*, wird die automatische Ausführung dieses Auftrags ausgesetzt. Dies gilt auch für Systemaufträge (gelb hinterlegt). 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.

Auftrag: Eindeutiger Name des Auftrags. Der Name wird beim Hinzufügen eines neuen Auftrags definiert.

Typ: Es existieren unterschiedliche Ausführungsarten für Aufträge (vgl. Seite 360).

- manuell: manuell auszuführender Auftrag
- Automatisch auszuführende Aufträge unterscheidet man wie folgt:
 - Einmalig
 - Regelmäßig

Intervall: Diese Spalte zeigt an, in welchem Intervall ein Auftrag ausgeführt wird. Diese Information wird nur für regelmäßig, automatisch ausgeführte Aufträge angezeigt:

- Täglich
- Alle n Minuten
- Wöchentlich, n Mal
- Monatlich, am n-ten Tag



Ein Auftrag kann erst dann gespeichert werden, wenn ein korrekter Startzeitpunkt ermittelt werden kann. Ist also eine wöchentliche Ausführung konfiguriert, muss auch ein Wochentag angegeben werden.

Letzte Ausführung: Diese Spalte zeigt Datum und Zeit der letzten Ausführung des Auftrags an.

Letzte Dauer: Hier wird der Zeitraum der letzten Auftragsausführung angezeigt.

Ø Dauer: Die durchschnittliche Dauer gehört, wie auch die "letzte Dauer", zu den statistischen Informationen, die zur Zeitplanung der Aufträge sinnvoll sein können.

Auftrags-ID: Eindeutige ID des Auftrags. Die ID wird automatisch beim Anlegen eines neuen Auftrags vergeben.



Durch Klick auf den Button "Hinzufügen" wird ein neuer Auftrag für dieses Projekt bzw. den Server angelegt (siehe Kapitel 7.5.4 Seite 360).

Durch Klick auf den Button "Bearbeiten" (oder per Doppelklick auf den gewünschten Auftrag) kann der selektierte Auftrag bearbeitet werden. Es öffnet sich ein neuer Dialog mit der entsprechenden Eingabemaske (siehe Kapitel 7.5.4 Seite 360).

Durch Klick auf den Button "Löschen" wird der selektierte Auftrag aus diesem Projekt bzw. vom Server gelöscht. Das Löschen muss durch eine Sicherheitsabfrage bestätigt werden. Die (gelb-hinterlegten) Systemaufträge können nicht gelöscht werden.

Durch einen Klick auf den Button "Ausführen" kann der gewünschte Auftrag, unabhängig vom eingestellten Ausführungsintervall direkt manuell ausgeführt werden.

Durch Klick auf den Button "Auftrag kopieren" besteht die Möglichkeit, einen Auftrag aus einem anderen Projekt heraus zu kopieren und dem aktuellen Projekt bzw. den serverweiten Aufträgen hinzuzufügen. Im ersten Schritt wird das Projekt ausgewählt, aus dem ein bestehender Auftrag kopiert werden soll. Der Auswahldialog zeigt alle auf dem Server vorhandenen Projekte an, für die der eingeloggte Benutzer als **Projekt-Administrator** eingetragen ist.

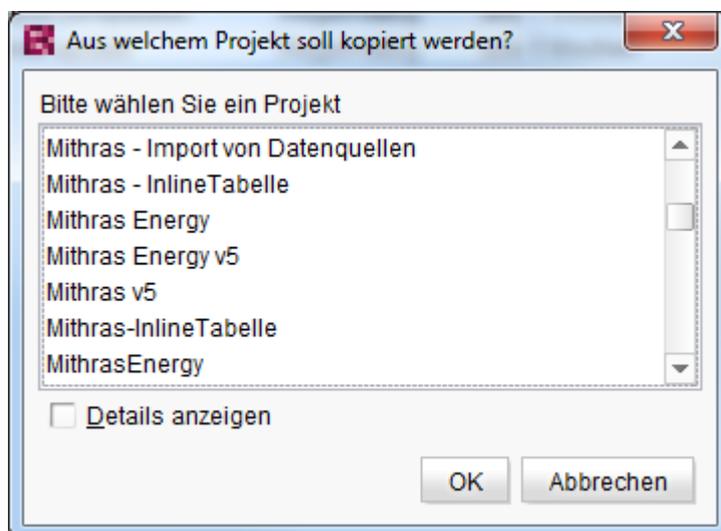


Abbildung 7-125: Auftrag kopieren – Auswahl eines Projekts

Im nächsten Schritt öffnet sich ein neuer Dialog, in dem alle Aufträge des ausgewählten Projekts aufgelistet werden.



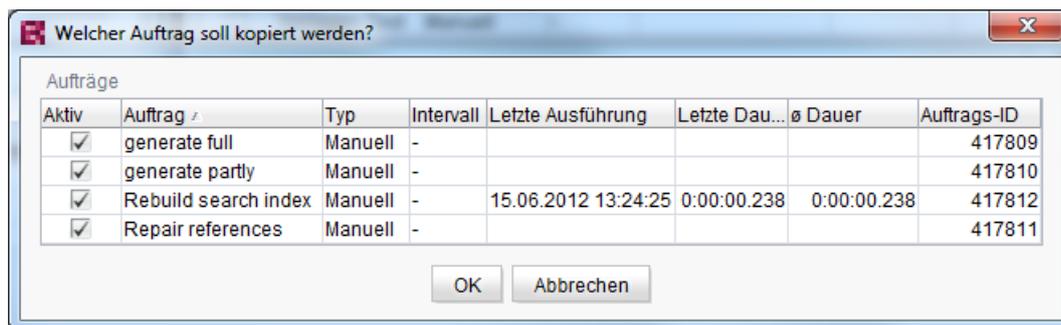


Abbildung 7-126: Auftrag kopieren – Auswahl des zu kopierenden Auftrags

Hier lässt sich nun ein einzelner Auftrag per Doppelklick in den bearbeiteten Auftrag kopieren.



7.5.3 Aktionsvorlagen

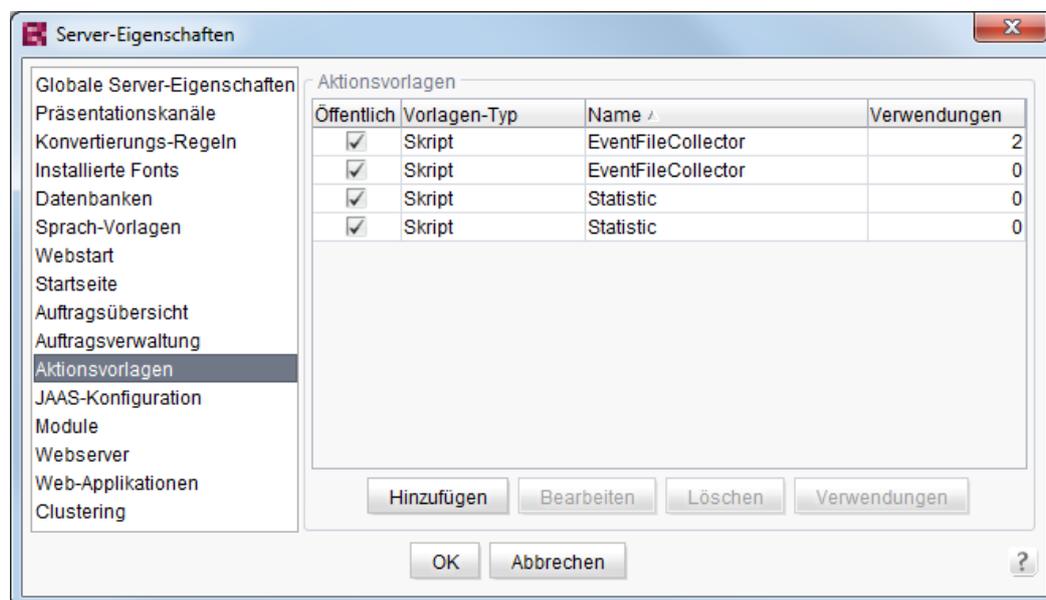


Abbildung 7-127: Aktionsvorlagen



Einige Änderungen an Aktionsvorlagen werden sofort gespeichert, ohne dass sie mit OK bestätigt werden müssen. Eine solche Speicherung wird auch dann nicht zurückgesetzt, wenn der Dialog über "Abbrechen" geschlossen wird.

Aktionsvorlagen ermöglichen die einfache Verwaltung von Aktionen, die in mehreren Aufträgen – auch in anderen Projekten – in identischer Konfiguration genutzt werden sollen. Aktionsvorlagen können nur an der Stelle editiert werden, an der sie angelegt wurden. Um Aktionsvorlagen auch anderen Projekten bzw. serverseitigen Aufträgen zur Verfügung stellen zu können, müssen diese als "öffentlich" gekennzeichnet werden.

Öffentlich: Aktionsvorlagen stehen grundsätzlich immer dem Projekt zur Verfügung, in dem sie erstellt wurden. Ist diese Option *aktiviert*, wird die Aktion zusätzlich auch noch allen anderen Projekten zur Verwendung bereitgestellt.

Vorlagen-Typ: Zeigt an, um welche Aktion es sich handelt. Möglich sind:

- Veröffentlichung ausführen: siehe Kapitel 7.5.10.6 Seite 393
- Skript ausführen: siehe Kapitel 7.5.9.4 Seite 373
- Mail verschicken: siehe Kapitel 7.5.9.1 Seite 371



Name: Name der Aktion. Abhängig von der gewählten Aktion wird der Name automatisch vergeben (vgl. Vorlagen-Typ):

- Deployment
- Script
- Mail

Verwendungen: Anzahl der Aufträge, die diese Aktionsvorlage verwenden.

Durch Klick auf den Button "Hinzufügen" wird eine neue Aktionsvorlage erstellt. Dafür muss zunächst der Aktions-Typ ausgewählt werden (siehe Kapitel 7.5.6 Seite 365).

Durch Klick auf den Button "Bearbeiten" kann die selektierte Aktionsvorlage bearbeitet werden. Es öffnet sich ein neuer Dialog mit der entsprechenden Eingabemaske (siehe Kapitel 7.5.10 Seite 379 und Kapitel 7.5.9 Seite 370).



Änderungen, die an einer Aktionsvorlage vorgenommen werden, wirken sich auf alle Aufträge aus, die diese Vorlage verwenden.

Durch Klick auf den Button "Löschen" kann die selektierte Aktionsvorlage gelöscht werden. Das Löschen muss durch eine Sicherheitsabfrage bestätigt werden.



Das Löschen einer Aktionsvorlage ist nur dann möglich, wenn diese in keinem Auftrag mehr verwendet wird.

Durch Klick auf den Button "Verwendungen" öffnet sich ein neuer Dialog, der alle Aufträge anzeigt, in denen diese Aktionsvorlage verwendet wird:

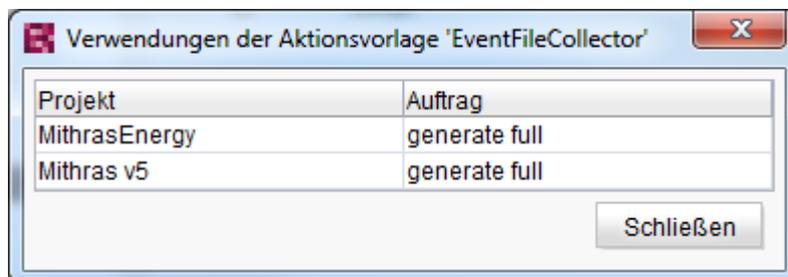


Abbildung 7-128: Anzeige der Verwendungen einer selektierten Aktionsvorlage



Projekt: Der Name des Projekts, zu dem der Auftrag gehört.

Auftrag: Der Name des Auftrags, der die Aktionsvorlage verwendet.

7.5.4 Hinzufügen / Bearbeiten eines Auftrags (Register Eigenschaften)

Durch einen Klick auf den Button "Hinzufügen" bzw. "Bearbeiten" innerhalb der Auftragsverwaltung (oder durch einen Doppelklick auf den gewünschten Tabelleneintrag) öffnet sich das Dialogfenster "Auftrag bearbeiten". Im Register "Eigenschaften" können die folgenden Auftragseigenschaften definiert werden.

Eigenschaften | Aktionen

Auftrags-ID: 820914

Auftragsname: generate full

eMail-Verteiler:

Verzeichnis: mithras_full

Manuell

Einmalig Ausführung am: 05.09.2012 um 09:07

Regelmäßig

 Erste Ausführung am: 05.09.2012 um 09:07

Täglich Ausführungsregel:

Wöchentlich

Monatlich

Intervall 0 Wochen 0 Tage 1 Stunden 0 Minuten

Interaktive Ausführung

Interaktive Ausführung erlaubt für:

Benutzer | Gruppen

ID	Name	Login	Kürzel	E-Mail
1	Admin	Admin	A	
124	chief editor	chief	chiefeditor	
125	editor	editor	editor	

Parallele Ausführung: Nicht erlaubt (abbrechen)

OK Abbrechen

Abbildung 7-129: Auftrag bearbeiten – Eigenschaften

Auftragsname: Eindeutiger Name des Auftrags. Unter diesem Namen wird der Auftrag sowohl in der Server- und Projektkonfiguration (siehe Abbildung 7-122 und Abbildung 7-124) als auch im JavaClient angezeigt.



E-Mail-Verteiler: Hier können E-Mail-Adressen angegeben werden, die unter anderem auch an eingebundene "Mail-Aktionen" (siehe Kapitel 7.5.9.1 Seite 371) vererbt und von diesen genutzt werden können.

Verzeichnis: Hier kann der Name des zugehörigen Generierungszielverzeichnisses angegeben werden. Durch die Angabe eines solchen Verzeichnisses werden im Fall von mehreren Generierungstasks Überschreibungen vermieden.

Ausführungstyp: Über die Radiobuttons kann einer der folgenden Ausführungstypen selektiert werden. Standardmäßig ist der Ausführungstyp "Manuell" selektiert.

- **Manuell:** Ist dieser Ausführungstyp ausgewählt, kann der Auftrag nur manuell gestartet werden. Die Angabe eines Ausführungszeitpunkts ist daher nicht möglich. Die entsprechenden Felder werden deaktiviert
- **Einmalig:** Im Gegensatz zur manuellen Ausführung, wird die einmalige Ausführung eines Auftrags automatisch zu einem bestimmten Zeitpunkt gestartet. Der Ausführungszeitpunkt kann per Klick auf das Kalendersymbol definiert werden.
- **Regelmäßig:** Ist dieser Ausführungstyp ausgewählt, wird der Auftrag automatisch in regelmäßigen Abständen ausgeführt. Dazu müssen per Klick auf das Kalendersymbol der erste Ausführungszeitpunkt sowie die Ausführungsregel eingestellt werden. Ein Auftrag kann erst dann gespeichert werden, wenn ein gültiger Startzeitpunkt ermittelt werden konnte (aus diesem Grund werden Standardwerte vorgegeben, die aber beliebig geändert werden können). Regelmäßige Aufträge können nach folgenden Regeln ausgeführt werden:
 - **Täglich**
Ein täglich auszuführender Auftrag bedarf keiner weiteren Ausführungsregel, da dieser immer zur eingestellten Uhrzeit ausgeführt wird.
 - **Wöchentlich**
Bei einem wöchentlich auszuführenden Auftrag muss definiert werden, an welchen Wochentagen dieser ausgeführt werden soll (Standardwert: Montag).



Abbildung 7-130: Ausführungsregel – Wöchentliche Ausführung

- **Monatlich**

Bei einem monatlich auszuführenden Auftrag muss bestimmt werden, an welchem Tag des Monats dieser ausgeführt werden soll (Standardwert: am ersten Montag des Monats).

Abbildung 7-131: Ausführungsregel – Monatliche Ausführung

- **Intervall**

Bei Auswahl des Radiobuttons "Intervall" muss lediglich das Intervall zwischen den Ausführungen angegeben werden (Standardwert: 1 Woche).

Abbildung 7-132: Ausführungsregel – Intervall-Ausführung

Abhängig vom selektierten Ausführungs-Typ können weitere Auftragseigenschaften editiert werden (siehe Abbildung 7-129):

Interaktive Ausführung: Ist diese Option *aktiviert*, dürfen alle selektierten Benutzer bzw. Benutzer, die zu einer der selektierten Gruppen gehören, diesen Auftrag auch interaktiv ausführen. Die interaktive Ausführung eines Auftrags ist nur aus dem JavaClient heraus möglich.

Benutzer

Diese Tabelle listet alle Benutzer auf, die berechtigt sind, diesen Auftrag interaktiv (aus dem JavaClient heraus) auszuführen.

Durch einen Klick auf den Ordner-Button öffnet sich ein Dialog, der alle Benutzer des aktuellen Projekts anzeigt. Bereits ausgewählte Benutzer werden durch Fettschrift hervorgehoben.



Durch einen Klick auf den Mülleimer-Button werden alle selektierten Benutzer aus der Tabelle gelöscht. Diese Benutzer besitzen anschließend kein Recht mehr, diesen Auftrag aus dem JavaClient heraus auszuführen.

Gruppen

Diese Tabelle listet alle Benutzergruppen auf, die berechtigt sind, diesen Auftrag interaktiv auszuführen.

Das Hinzufügen und Löschen einer Gruppe erfolgt analog zum Hinzufügen und Löschen von Benutzern.

Parallele Ausführung:

- Nicht erlaubt (abbrechen):
Ist diese Option ausgewählt, ist die parallele Ausführung dieses Auftrags nicht erlaubt. Läuft bereits ein Auftrag, wird der Versuch, die Ausführung zu starten, geblockt.
- Erlaubt (parallel ausführen):
Ist diese Option ausgewählt, wird die Ausführung des Auftrags unmittelbar nach Anfrage gestartet (auch, wenn parallel bereits ein Auftrag läuft).
- Nicht erlaubt (nacheinander ausführen):
Ist diese Option ausgewählt, ist die parallele Ausführung des Auftrags nicht erlaubt. Der Auftrag wird jedoch nach Beendigung der laufenden Ausführung (falls vorhanden) automatisch gestartet.



7.5.5 Hinzufügen / Bearbeiten eines Auftrags (Register Aktionen)

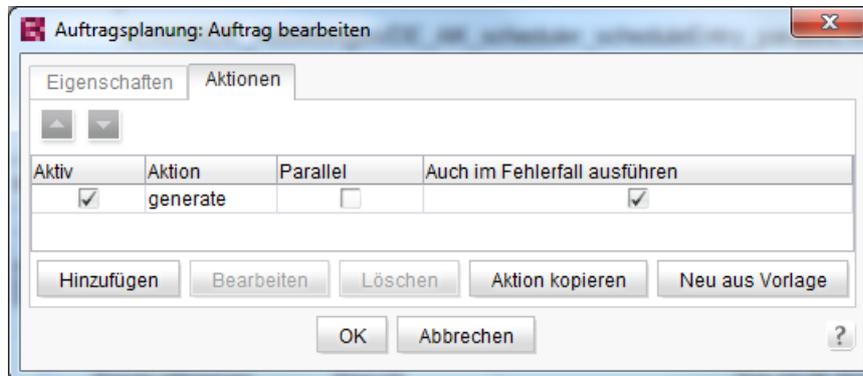


Abbildung 7-133: Auftrag bearbeiten – Aktionen

Durch einen Klick auf den Button "Hinzufügen" bzw. "Bearbeiten" innerhalb der Auftragsverwaltung (oder durch einen Doppelklick auf den gewünschten Tabelleneintrag) öffnet sich das Dialogfenster "Auftragsplanung". Im Register "Aktionen" können Aktionen zum Auftrag hinzugefügt, bearbeitet oder gelöscht werden.

Alle zugehörigen Aktionen werden in der Reihenfolge angezeigt, in der sie bei Ausführung des Auftrags durchlaufen werden. Diese Reihenfolge kann geändert werden, indem eine Aktion – in dem Fall eine Zeile der Tabelle – selektiert wird und per Klick auf den entsprechenden Button   entweder eine Position nach oben oder unten verschoben wird.

Die Tabelle enthält drei Spalten, welche den Namen sowie zwei – zur Ausführung wichtige – Eigenschaften der Aktion zeigen.

Aktiv: Ist die Option *aktiviert*, wird diese Aktion bei Ausführung des Auftrags ebenfalls ausgeführt, ist sie *deaktiviert*, wird die Aktion übersprungen.

Aktion: Name der Aktion.

Parallel: Ist die Option *aktiviert*, darf diese Aktion innerhalb eines Auftrags parallel zu einer anderen Aktion ausgeführt werden. Diese Einstellung ist nur sinnvoll, wenn mehrere aufeinander folgende Aktionen auch zur parallelen Ausführung freigegeben sind und wenn diese Aktionen logisch unabhängig voneinander sind. Wird die Option aktiviert, wird die Zeile innerhalb der Übersicht gelb hinterlegt.

Aktiv	Aktion	Parallel	Auch im Fehlerf...
<input checked="" type="checkbox"/>	generate	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>



Auch im Fehlerfall ausführen: Ist diese Option aktiviert, wird auch nach einer vorangegangenen fehlerhaften Aktion, die jeweils nachfolgende Aktion ausgeführt.

Durch einen Klick auf den Button "Hinzufügen" wird eine neue Aktion für diesen Auftrag angelegt (siehe Kapitel 7.5.6 Seite 365).

Durch einen Klick auf den Button "Bearbeiten" (oder einen Doppelklick auf den Tabelleneintrag) öffnet sich ein neuer Dialog zum Bearbeiten der selektierten Aktion (siehe Kapitel 7.5.10 Seite 379 und Kapitel 7.5.9 Seite 370).

Durch einen Klick auf den Button "Löschen" wird die selektierte Aktion aus diesem Auftrag gelöscht. Das Löschen muss durch eine Sicherheitsabfrage bestätigt werden.

Durch einen Klick auf den Button "Aktion kopieren" besteht die Möglichkeit, eine Aktion aus einem anderen Auftrag heraus zu kopieren und dem aktuellen Auftrag hinzuzufügen (siehe Kapitel 7.5.7 Seite 368).

Durch einen Klick auf den Button "Neu aus Vorlage" kann eine Aktion, die zuvor in den Aktionsvorlagen definiert wurde (siehe Kapitel 7.5.3 Seite 358), zum aktuellen Auftrag hinzugefügt werden (siehe Kapitel 7.5.8 Seite 369).



Aktionen, die dem Auftrag anhand einer Vorlage hinzugefügt werden, können an dieser Stelle nicht editiert werden. Änderungen an dieser Aktion können nur über die entsprechende Vorlage im Bereich "Aktionsvorlagen" durchgeführt werden (siehe Kapitel 7.5.3 Seite 358).

7.5.6 Hinzufügen von Aktionen zu einem Auftrag

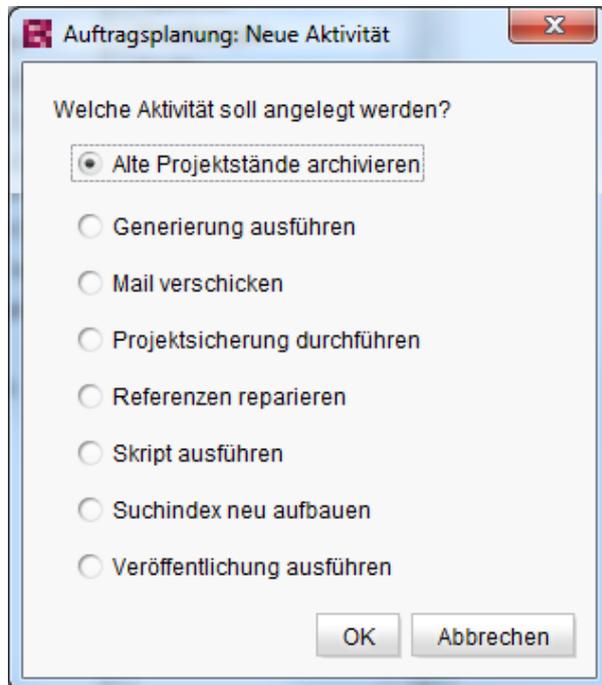
Durch einen Klick auf den Button "Hinzufügen" innerhalb der Auftragsverwaltung (oder durch einen Doppelklick auf den gewünschten Tabelleneintrag) öffnet sich das Dialogfenster "Neue Aktivität".

Wie bereits in der Einleitung zum Kapitel 7.4.21 beschrieben, wird beim Anlegen einer Aktion unterschieden, ob es sich um einen projektbezogenen oder serverbezogenen Auftrag handelt:

- Hinzufügen einer projektbezogenen Aktion (siehe Kapitel 7.5.6.1).
- Hinzufügen einer serverbezogenen Aktion (siehe Kapitel 7.5.6.2).



7.5.6.1 Anlegen einer projektbezogenen Aktion

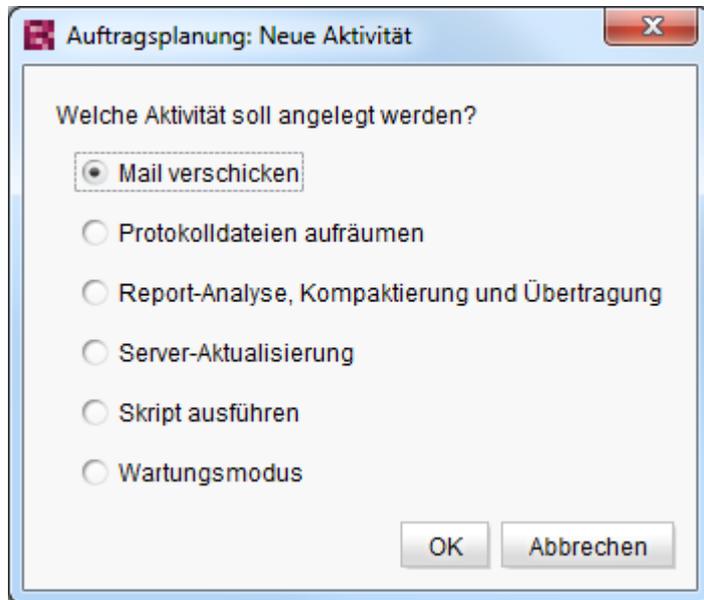
**Abbildung 7-134: Anlegen einer projektbezogenen Aktion**

Abhängig von der selektierten Aktion wird die Eingabemaske des ausgewählten Aktionstyps geöffnet:

- Alte Projektstände archivieren siehe Kapitel 7.5.10.1 Seite 380
- Generierung ausführen: siehe Kapitel 7.5.10.2 Seite 385
- Mail verschicken: siehe Kapitel 7.5.9.1 Seite 371
- Projektsicherung durchführen siehe Kapitel 7.5.10.3 Seite 391
- Referenzen reparieren siehe Kapitel 7.5.10.4 Seite 392
- Skript ausführen: siehe Kapitel 7.5.9.4 Seite 373
- Suchindex neu aufbauen siehe Kapitel 7.5.10.5 Seite 392
- Veröffentlichung ausführen: siehe Kapitel 7.5.10.6 Seite 393



7.5.6.2 Anlegen einer serverbezogenen Aktion

**Abbildung 7-135: Anlegen einer serverbezogenen Aktion**

Entsprechend der getroffenen Auswahl wird die Eingabemaske des ausgewählten Aktionstyps geöffnet:

- Mail verschicken: siehe Kapitel 7.5.9.1 Seite 371
- Protokolldateien aufräumen siehe Kapitel 7.5.9.2 Seite 372
- Report-Analyse siehe Kapitel 7.5.9.3 Seite 373
- Server-Aktualisierung siehe Kapitel 7.5.9.3 Seite 373
- Skript ausführen: siehe Kapitel 7.5.9.4 Seite 373
- Wartungsmodus siehe Kapitel 7.5.9.5 Seite 376



7.5.7 Kopieren von Aktionen aus einem anderen Auftrag

Über den Button "Aktion kopieren" im Dialogfenster "Auftrag bearbeiten" (siehe Kapitel 7.5.5 Seite 364) kann eine bereits erstellte Aktion aus einem anderen Auftrag in den aktuell bearbeiteten Auftrag kopiert werden. Es öffnet sich das folgende Dialogfenster:

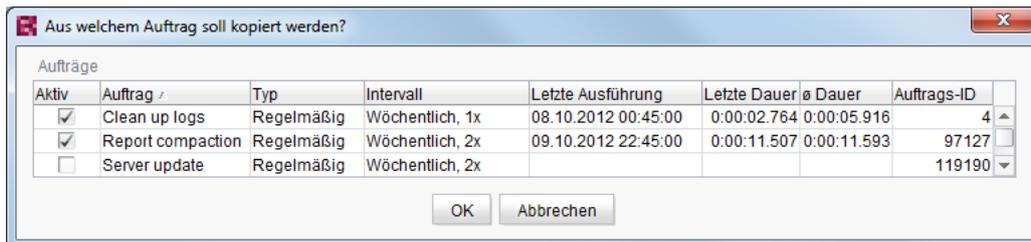


Abbildung 7-136: Aktion kopieren

Dieser Dialog zeigt eine Tabelle aller vorhandenen Aufträge:

- Innerhalb der Projekteigenschaften: alle projektbezogenen Aufträge.
- Innerhalb der Servereigenschaften: alle serverbezogenen Aufträge.

Per Doppelklick lässt sich der Auftrag, aus dem eine oder mehrere Aktionen kopiert werden sollen, auswählen. Es öffnet sich ein neuer Dialog, der alle Aktionen des ausgewählten Auftrags auflistet:

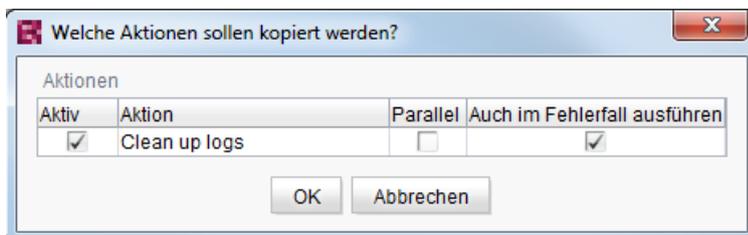


Abbildung 7-137: Aktion kopieren – Auswahl der zu kopierenden Aktion

Hier lässt sich nun ein einzelner Auftrag per Doppelklick in den bearbeiteten Auftrag kopieren.





Neben der Möglichkeit einzelne Aktionen zu kopieren, können alternativ auch mehrere Aktionen gleichzeitig kopiert werden. Dazu werden alle Aktionen bei gedrückter "STRG"-Taste selektiert. Nach Bestätigung per Klick auf den Button OK werden alle ausgewählten Aktionen kopiert und dem Auftrag hinzugefügt.

7.5.8 Hinzufügen von Aktionen über Aktionsvorlagen

Für Aktionen, die in gleicher Konfiguration in mehreren Aufträgen wiederverwendet werden sollen, können analog zum Vorlagenkonzept von FirstSpirit "Aktionsvorlagen" erstellt werden (siehe Kapitel 7.5.3). Über die Aktionsvorlage kann eine Aktion beliebig vielen Aufträgen hinzugefügt werden.

Über den Button "Neu aus Vorlage" im Dialogfenster "Auftrag bearbeiten" (siehe Kapitel 7.5.5 Seite 364) kann eine Aktion auf Basis einer bereits erstellten Aktionsvorlage erstellt und dem aktuell bearbeiteten Auftrag zugefügt werden. Es öffnet sich das folgende Dialogfenster:

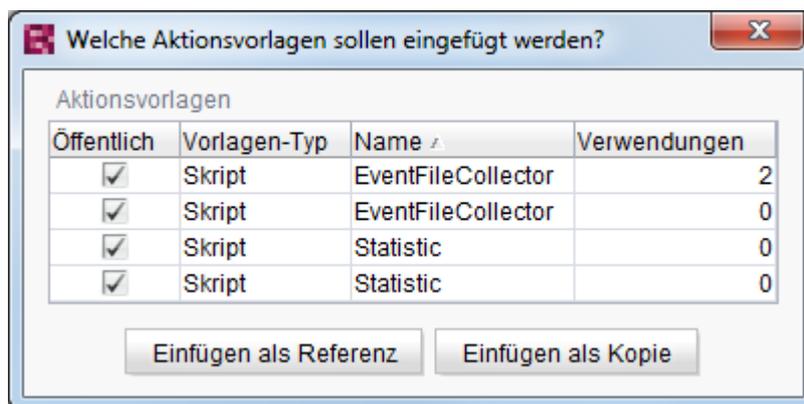


Abbildung 7-138: Aktion hinzufügen – Auswahl einer Aktionsvorlage

Abbildung 7-138 zeigt die zur Verfügung stehenden Aktionsvorlagen.

Es sind immer nur die Aktionsvorlagen verfügbar, die entweder in der lokalen Aktionsvorlagenverwaltung oder beispielsweise in der Aktionsvorlagenverwaltung eines anderen Projekts definiert und als "öffentlich" markiert wurden (vgl. Kapitel 7.5.3 Seite 358). Per Doppelklick lassen sich nun einzelne Aktionen zum Auftrag hinzufügen.





Neben der Möglichkeit, einzelne Aktionen anhand der Aktionsvorlagen anzulegen, können alternativ auch mehrere Aktionen gleichzeitig eingefügt werden. Dazu werden alle gewünschten Vorlagen bei gedrückter "STRG"-Taste selektiert.

Mit einem Klick auf den Button "Einfügen als Referenz" wird eine Aktion als Referenz auf eine Aktionsvorlage dem Auftrag hinzugefügt. Aktionen, die als Referenz auf eine Aktionsvorlage eingefügt wurden, können nur in der Vorlagenverwaltung selbst, jedoch in keinem Auftrag editiert werden. Referenzierte Aktionen werden in der Tabelle mit kursiver Schrift dargestellt.

Mit einem Klick auf den Button "Einfügen als Kopie" wird eine Aktion als Kopie der selektierten Aktionsvorlage dem Auftrag hinzugefügt. Kopien einer Aktion können normal bearbeitet werden.

Mit einem Klick auf den Button "Abbrechen" wird das Dialogfenster geschlossen. Es wird keine Aktion zum Auftrag hinzugefügt.

7.5.9 Serverbezogene Aktionen

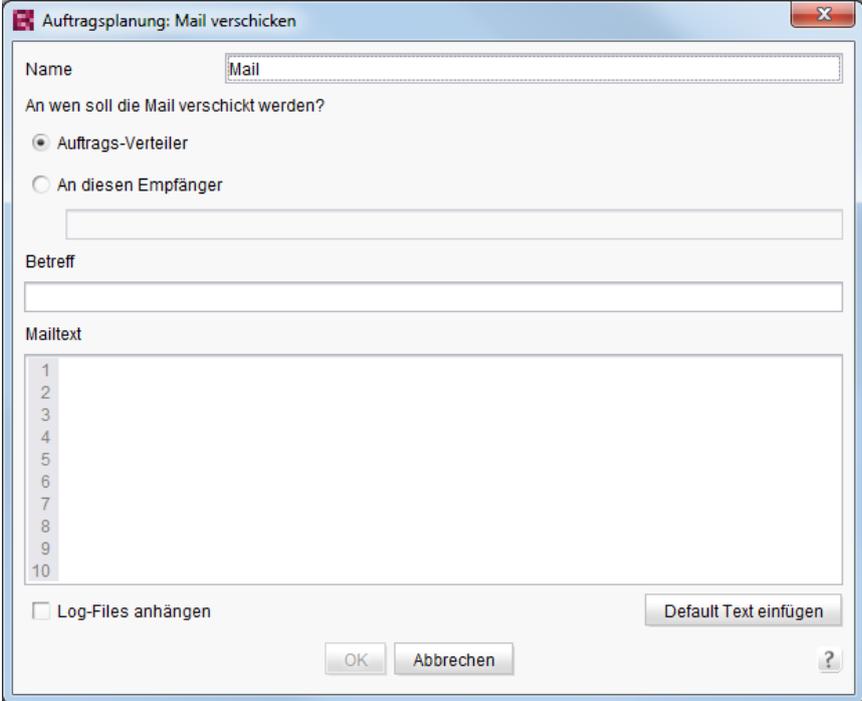
Serverbezogene Aktionen werden innerhalb der Servereigenschaften angelegt und zu serverbezogenen Aufträgen hinzugefügt.

Folgende Aktionen stehen zur Verfügung:

- Mail verschicken: Analog zu Kapitel 7.5.9.1 (siehe Seite 371)
- Skript ausführen: Analog zu Kapitel 7.5.9.4 (siehe Seite 373)



7.5.9.1 Mail verschicken

**Abbildung 7-139: Aktion anlegen – Mail verschicken**

Diese Aktion bietet die Möglichkeit E-Mails zu versenden. Dabei kann entweder der E-Mail-Verteiler des zugehörigen Auftrags verwendet oder ein eigener definiert werden. Die Besonderheit liegt darin, dass es möglich ist, die Log-Files von zuvor ausgeführten Aktionen im selben Auftrag im Anhang mit zu versenden.

Auftrags-Verteiler: Ist diese Option *aktiviert*, wird die E-Mail an alle Empfänger versendet, die im Verteiler des zugehörigen Auftrags definiert sind.

An diesen Empfänger: Ist diese Option *aktiviert*, wird die E-Mail an alle im folgenden Text-Feld definierten E-Mail-Adressen versendet.

Betreff: Der Betreff der versendeten E-Mails.

Mailtext: Hier ist der Mailtext einzugeben. Dieser Text darf Template-Syntax enthalten, welcher dann vor dem Versenden der E-Mails geparkt wird. Über die Variable `#context` können Informationen zum aktuellen Auftrag (inklusive der in diesem Auftrag ausgeführten Aktionen) und über die Variable `#task` Informationen



zur Aktion "Mail verschicken" selbst ausgegeben werden³².

Beispiele:

`$CMS_VALUE(#context.getStartTime())$`: gibt den Start-Zeitpunkt des Auftrags aus

`$CMS_VALUE(#task.getSubject())$`: gibt den Betreff der versendeten E-Mail aus

Log-Files anhängen: Ist diese Option *aktiviert* werden die Log-Files – sofern vorhanden – der vorher ausgeführten Aktionen im selben Auftrag im Anhang der Mail mit versendet.

Bei Klick auf den Button "Default Text einfügen" wird dem vorhandenen Mailtext der Inhalt der Konfigurationsdatei für die Standardvorlage (%Serververzeichnis%/conf/server/DefaultMailText.txt) angehängt.

7.5.9.2 Protokolldateien aufräumen

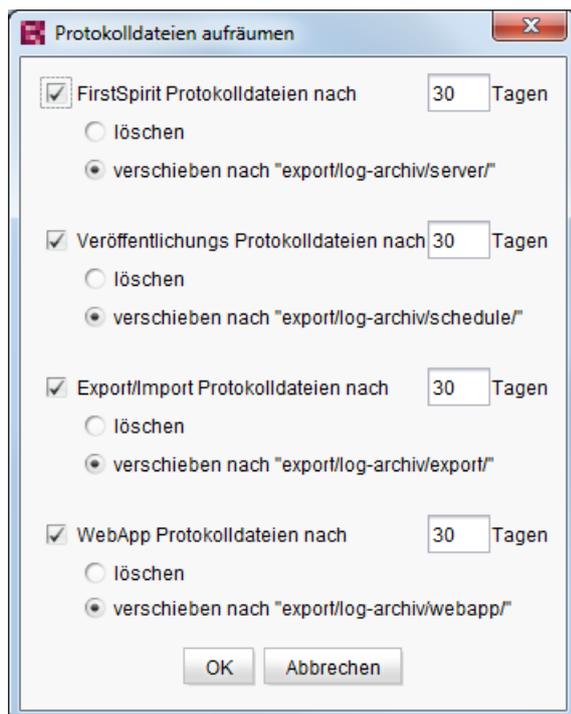


Abbildung 7-140: Aktion anlegen – Protokolldateien aufräumen

³² Methoden zu `#context` und `#task` können der FirstSpirit Access-API entnommen werden (`de.espirit.firstspirit.access.schedule.ScheduleContext` bzw. `de.espirit.firstspirit.access.schedule.MailTask`).



Um die Übersichtlichkeit der Server-Protokolldateien zu bewahren, besteht die Möglichkeit, sie über diese Funktion aufzuräumen. Dabei kann angegeben werden, wie alt die Protokolldateien sein sollen, die bei der Durchführung der Aktion berücksichtigt werden. Je nach Auswahl werden sie dabei gelöscht oder in ein entsprechendes Archiv-Verzeichnis verschoben. Die Auswahl der aufzuräumenden Protokolldateien lässt sich über die jeweiligen Checkboxen zum De-/Aktivieren steuern.

Standardmäßig besteht nach der Server-Installation bereits ein aktivierter Auftrag mit dieser Aktion, der automatisch einmal wöchentlich ausgeführt wird.

7.5.9.3 Report-Analyse, Kompaktierung und Übertragung / Server-Aktualisierung

Diese beiden Features sind für die Benutzung durch Endkunden in der FirstSpirit Version 5.0 noch nicht freigegeben und stehen bisher nur Partnern für Beta-Tests zur Verfügung.

Eine manuelle Server-Aktualisierung ist jedoch bereits über das Server-Monitoring möglich.

Siehe dazu Kapitel 8.6.2.3 Seite 460.

7.5.9.4 Skript ausführen

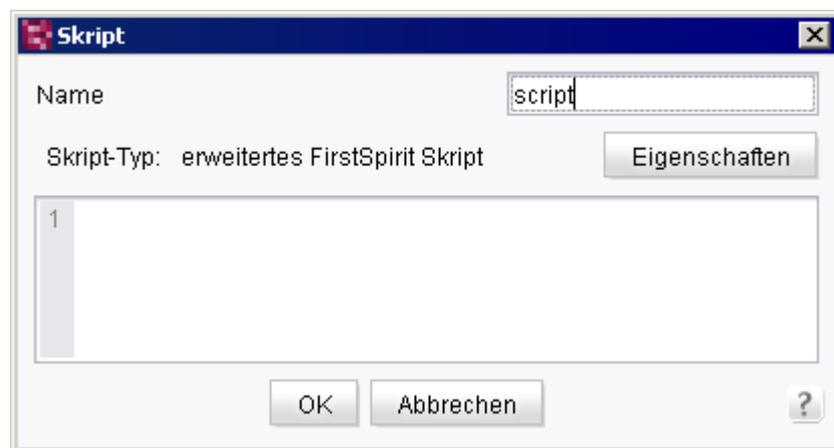


Abbildung 7-141: Aktion anlegen – Skript ausführen

Name: In diesem Feld kann ein Name für das Skript vergeben werden.

Texteingabefeld: In diesem Feld ist der Skript-Code einzutragen, der in dieser



Aktion ausgeführt werden soll.

Per Klick auf den Button "Eigenschaften" öffnet sich ein neuer Dialog, in dem die Eigenschaften dieses Skripts editiert werden können (siehe Kapitel 7.5.9.4.1 Seite 374). Dies ist beispielsweise dann notwendig, wenn das Skript ändernde Operationen im Projekt durchführen soll (z. B. Objekte anlegen oder ändern).

7.5.9.4.1 Skript Eigenschaften

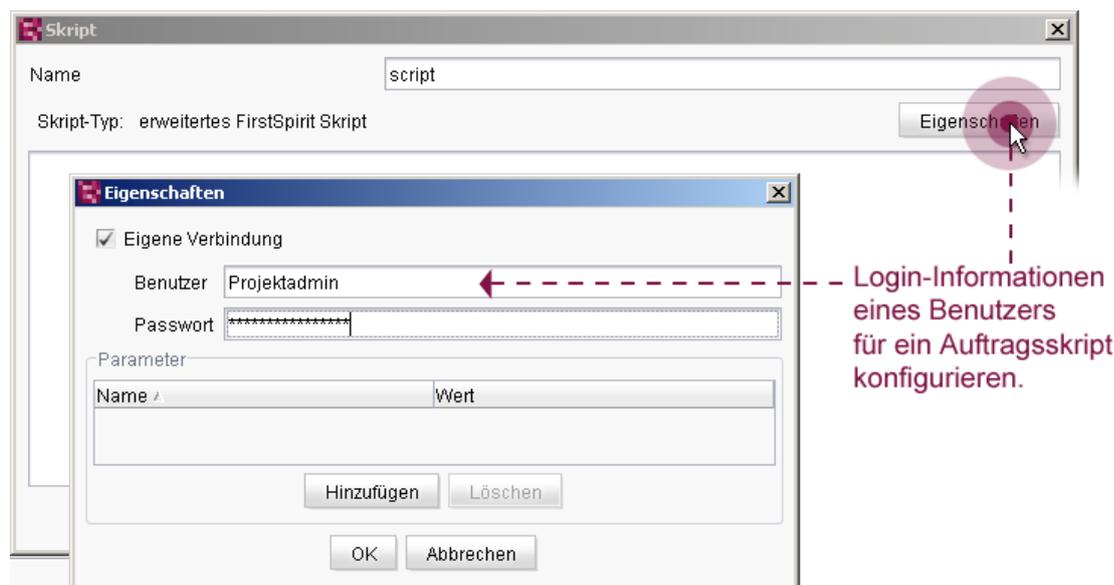


Abbildung 7-142: Skript editieren – Skript Eigenschaften

Über diesen Dialog kann – auf Wunsch – eine eigene Verbindung zum Server aufgebaut werden, anstatt die laufende Sitzung zur Ausführung des Skripts zu nutzen.

Eigene Verbindung: Ist diese Option *aktiviert*, wird anhand folgender Daten zur Ausführung des Skripts eine neue Serververbindung aufgebaut.

Benutzer: Der Benutzername, der zur Anmeldung am Server verwendet werden soll.

Passwort: Das entsprechende Passwort zum oben angegebenen Benutzer.





Zum Aufbau einer neuen Serververbindung werden die Daten eines internen FirstSpirit-Benutzers benötigt. Die Verwendung eines externen Benutzers ist nicht möglich.

Parameter: Hier werden alle Parameter eingegeben, die bei der Ausführung des Skripts berücksichtigt werden sollen.

Ist die Option **Eigene Verbindung** aktiviert, können bei der Ausführung des Skripts die Redaktionsrechte des hier angegebenen Benutzers ausgewertet und berücksichtigt werden.

Es ist möglich, Auftragskripte über die Systemverbindung auszuführen, z. B.:

```
context.getUserService();
```

Die Variable `context` stellt eine spezielle Verbindung bereit, die (über die Rechte `CAN_SEE`, `CAN_READ`, `CAN_META_SEE`) nur lesenden Zugriff besitzt. Solch eine Verbindung wird von Skripten, die ohne spezielle Benutzerinformationen konfiguriert wurden, genutzt. Mit ihr können lediglich nicht verändernde Operationen ausgeführt werden. Andernfalls wird die Ausführung des Skripts mit einer Security-Exception abgebrochen.

Sollen verändernde Operationen im Projekt über ein Auftragskript durchgeführt werden, muss das Auftragskript die Login-Informationen des angegebenen Benutzers verwenden. Dieser muss die entsprechenden Rechte besitzen (z. B. `CAN_CHANGE`, `APPEND`, `DELETE`, etc.) und zunächst über die Skripteigenschaften definiert werden (s.o.). Innerhalb des Skripts muss dann über die Variable `connection` eine benutzerspezifische Verbindung (basierend auf den hinterlegten Benutzerinformationen) geholt werden, z. B.:

```
connection.getProjectById(context.getProject().getId()).getUserService();
```

Das Auftragskript läuft anschließend im Rechtekontext des angegebenen Benutzers und wertet bei jeder verändernden Operation dessen Redaktionsrechte aus (z. B. beim Anlegen).



Werden in einem Skript (innerhalb eines Auftrags) verändernde Aktionen an den Projekteigenschaften ausgeführt, so muss dieses Skript entweder im Benutzerkontext eines Server- oder eines Projektadministrators ausgeführt werden.



7.5.9.5 Wartungsmodus

Der Wartungsmodus dient folgenden Zwecken:

- zur Aktualisierung des FirstSpirit-Servers. Dafür muss der Server heruntergefahren werden. (Dies wird allerdings nicht automatisch von der Wartungsmodus-Funktionalität durchgeführt, sondern muss manuell während des Wartungsmodus vorgenommen werden)
- um den Zugriff, ggf. auch für bestimmte Benutzergruppen, auf Projekte zu beschränken (z. B. bei Umbauarbeiten, größeren Updates).

Wird der Wartungsmodus aktiviert, können sich je nach Konfiguration keine Benutzer am FirstSpirit-Server bzw. an ausgewählten Projekten anmelden, geöffnete Clients werden je nach Konfiguration automatisch beendet. Eingeloggte FirstSpirit-Benutzer werden benachrichtigt werden, dass der Server heruntergefahren wird, und erhalten so die Gelegenheit, aktuelle Änderungen zu speichern und sich abzumelden.

Damit der Wartungsmodus aktiviert wird, ist ein entsprechender Server-Auftrag erforderlich.

Wartungsmodus

Hinweinsnachricht

Hinweis auf der Webseite für alle Projekte anzeigen

(Die Zeitpunkte sind jeweils relativ zum Zeitpunkt davor)

1. Warnung anzeigen nach 30 Minuten

2. Sitzungsendewarnung anzeigen nach 15 Minuten

3. Neue Sitzungen ablehnen nach 5 Minuten

4. Wartungsmodus starten nach 5 Minuten

5. Geplante Dauer 15 Minuten

Sitzungen während der Wartung beenden, neue Sitzungen ablehnen

Aufträge nicht annehmen

Gültig für alle Projekte

Gültig für Projekte

Nicht gültig für Benutzer

Nicht gültig für Gruppen

Abbildung 7-143: Aktion anlegen – Wartungsmodus



Bis auf den Startzeitpunkt (Punkt 4.) und die geplante Dauer (Punkt 5.) des Wartungsmodus sind alle Angaben innerhalb des Dialogs optional. Sind Angaben vorhanden, sollte darauf geachtet werden, dass sie zueinander passen.

Hinweisnachricht: In diesem Feld kann ein Text eingegeben werden, der zusätzlich zu den Systemmeldungen ausgegeben wird.

Hinweis auf der Webseite für alle Projekte anzeigen: Diese Option ist standardmäßig aktiviert. Wird sie deaktiviert, werden die Systemmeldungen nur in den Clients, jedoch nicht auf der FirstSpirit-Startseite angezeigt.

Der Wartungsmodus besteht aus mehreren Stufen. Die Punkte 1. bis 5. werden automatisch nacheinander abgearbeitet, sobald der Auftrag gestartet ist. Die Minutenangaben geben die Startzeitpunkte der einzelnen Stufen in Relation zur jeweils vorherigen Stufe an. Wird als Minutenangabe der Wert 0 eingegeben oder das Feld leer gelassen, wird die betreffende Stufe übersprungen. Die Abbildung 7-143 zeigt die standardmäßig voreingestellten Werte. Die ausgegebenen Systemmeldungen richten sich danach, ob der Auftrag für den gesamten Server (Option: "Gültig für alle Projekte", siehe unten) oder nur für ein bzw. mehrere Projekte (Option: "Gültig für Projekte", siehe unten) gilt.

1. Warnung anzeigen nach: Nach der hier angegebenen Zahl an Minuten wird als Vorwarnung, dass Wartungsarbeiten geplant sind, die Systemmeldung "In x Minuten werden Wartungsarbeiten am FirstSpirit-Server | an diesem Projekt durchgeführt. Geplante Dauer: ca. y Minuten." plus ggf. der im Feld "Hinweisnachricht" eingegebene Text ausgegeben. x ist dabei die Zahl, die in diesem Feld eingetragen wird, y die Zahl, die bei der Option "Geplante Dauer" eingetragen wird.

2. Sitzungsendewarnung anzeigen nach: Nach der hier angegebenen Minutenanzahl wird die Systemmeldung "Der FirstSpirit-Server | Das Projekt ist wegen Wartungsarbeiten ab 21.08.2012 15:15:00 nicht verfügbar. Geplante Dauer: ca. y Minuten. Bitten beenden Sie Ihre Sitzung." plus ggf. der im Feld "Hinweisnachricht" eingegebene Text ausgegeben. Eingeloggte Benutzer haben nun die Gelegenheit, ihre Arbeiten zu speichern und sich abzumelden. Datum und Uhrzeit sind dabei der Startzeitpunkt des Wartungsmodus-Auftrags plus die Angaben der Stufen 1. bis 4.

3. Neue Sitzungen ablehnen nach: Ist die Option "Sitzungen beenden, neue Sitzungen ablehnen" (siehe unten) aktiviert, sind nach der hier angegebenen Zeit in Minuten keine neuen Anmeldungen am Server bzw. in den ausgewählten Projekten mehr möglich.



4. Wartungsmodus starten nach: Nach der hier angegebenen Zahl an Minuten beginnt der Wartungszeitraum. Ist die Option "Sitzungen beenden, neue Sitzungen ablehnen" (siehe unten) aktiviert, werden noch angemeldete Clients mit der Meldung "Der FirstSpirit-Server | Das Projekt ist wegen Wartungsarbeiten nicht verfügbar. Ihre Sitzung wurde beendet." beendet. Je nach Einstellung wird auf der Startseite und im Client die Systemmeldung "Der FirstSpirit-Server ist seit 21.08.2012 15:15:00 im Wartungsmodus. Geplante Dauer: ca. y Minuten." ausgegeben. Nun können die Wartungsarbeiten vorgenommen und bei Bedarf der FirstSpirit-Server heruntergefahren werden.

5. Geplante Dauer: Hier kann angegeben werden, wie viele Minuten der Wartungszeitraum dauern soll. Nach Ablauf dieser Zeitspanne wird automatisch folgende Systemmeldung ausgegeben: "Die Wartungsarbeiten am FirstSpirit-Server | am Projekt wurden beendet." Ein Anmelden am Server ist wieder möglich. Sollte sich während eines laufenden Auftrags herausstellen, dass doch ein längerer Wartungsmodus erforderlich ist, als hier eingestellt wurde, sollte der Auftrag (vor Ablauf dieser Stufe) manuell beendet werden.

In welcher Stufe sich ein gestarteter Wartungsmodus-Auftrag befindet, kann im Server-Monitoring eingesehen werden (siehe Kapitel 8.6.2.1 Seite 457).

Sitzungen beenden, neue Sitzungen ablehnen: Wird diese Option aktiviert, werden alle Sitzungen des Servers bzw. der ausgewählten Projekte während des Wartungszeitraums beendet. Wird die Option deaktiviert, werden keine Sitzungen beendet. Um neue Sitzungen zuzulassen, muss zusätzlich bei Stufe 3. "Neue Sitzung ablehnen nach" der Wert gesetzt werden.

Aufträge nicht annehmen: Wird diese Option aktiviert, werden Aufträge, die in den Wartungszeitraum fallen, abgebrochen.

Gültig für alle Projekte: Ist diese Option aktiviert, gilt der Auftrag für alle Projekte auf dem Server. Wird die Option deaktiviert, können unter "Gültig für Projekte" das oder die gewünschten Projekte ausgewählt werden.

Gültig für Projekte: Hier kann das oder die Projekte ausgewählt werden, für die der Auftrag gelten soll.

Nicht gültig für Benutzer: Hier kann ein Benutzer ausgewählt werden, der sich im Wartungsmodus am FirstSpirit-Server bzw. an den ausgewählten Projekten anmelden kann. Der Serveradministrator kann sich in jedem Fall anmelden.

Nicht gültig für Gruppen: Hier können Gruppen ausgewählt werden, die sich im Wartungsmodus am FirstSpirit-Server bzw. an den ausgewählten Projekten



7.5.10.1 Alte Projektstände archivieren



Zur Verwendung der Archiv-Funktion im Vergleich zur Verwendung des Moduls "FirstSpirit EnterpriseBackup" siehe auch FirstSpirit Release Notes 4.2, Kap. "Langzeit-Archivierung und Backup in FirstSpirit".

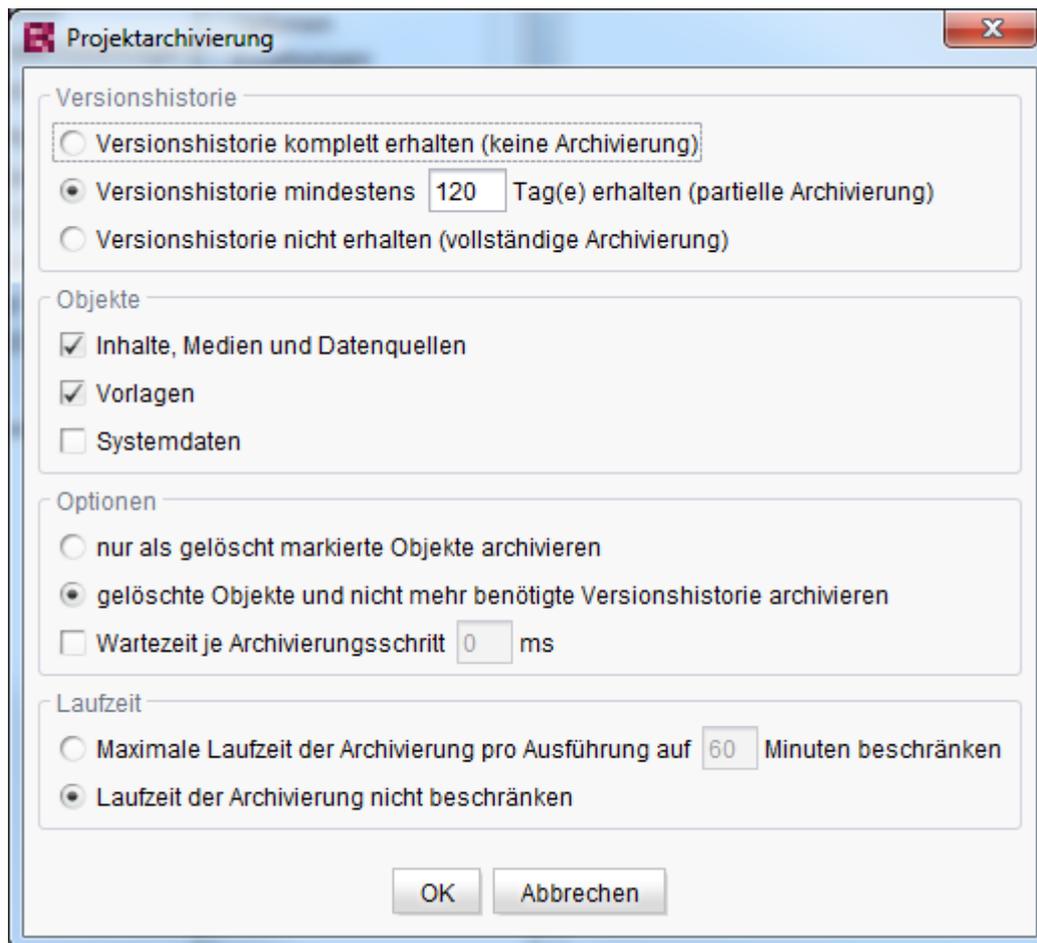


Abbildung 7-144: Auftrag anlegen – Projektarchivierung

FirstSpirit setzt zur Archivierung und Versionierung von Projektdaten Repositories ein. Für jedes Projekt existiert ein Repository im Server-Verzeichnis `data\projects\`. Bei jeder Aktion, die im JavaClient vorgenommen wird, werden Daten in das Repository geschrieben. Das gilt sowohl für Aktionen, die neue Elemente erstellen, als auch für Aktionen, die Elemente löschen. Darüber hinaus werden auch gelöschte Elemente nicht aus dem Repository entfernt. Da so immer neue Daten hinzukommen, wird das Repository größer und es wird immer mehr



Festplattenplatz benötigt.

Der Auftrag "Alte Projektstände archivieren" dient dazu, eine Archivierung des gewählten Projekts durchzuführen, um so nicht mehr benötigte Daten aus dem Projekt auszulagern und damit Ladezeiten zu reduzieren und die Performance des FirstSpirit-Servers zu erhöhen. Dazu werden Daten aus den Repositories in Archivdateien verschoben. Nicht mehr benötigte Archivdateien können später gelöscht werden, um Speicherplatz auf der Festplatte endgültig freizugeben.

Für jedes Projekt wird bei der Archivierung ein eigener Ordner im Server-Verzeichnis `archive` angelegt und die zu archivierenden Daten in eine Archivdatei (Format `tar.gz`) im entsprechenden Projektordner verschoben. Sie können jedoch später über die Funktion "Archiv" in den Server- und Projekteigenschaften angezeigt und bei Bedarf installiert werden (siehe Kapitel 7.2.3.8 Seite 220).

Weitere Informationen zum FirstSpirit Archivierungs-Konzept befinden sich in Kapitel 7.9 ab Seite 423.



Die Archivierungsfunktion sollte mit Bedacht verwendet werden. Denn je nach Einstellung liegt die Versionshistorie nach einer Archivierung nicht mehr komplett vor. Somit können auch ältere Revisionen möglicherweise nicht mehr fehlerfrei wiederhergestellt werden. Bei einem Export (siehe Kapitel 7.2.3.3 Seite 216) wird nach einer Archivierung ebenfalls nur der aktuell verfügbare Projektstand exportiert, ohne eventuell vorliegende Archive bzw. archivierte Projektstände.

Über die folgenden Optionen können die Archivierungskriterien festgelegt werden:

Versionshistorie:

In diesem Bereich kann bestimmt werden, für welchen Zeitraum eine Archivierung durchgeführt werden soll bzw. für welchen Zeitraum die Versionshistorie erhalten bleiben soll.

Versionshistorie komplett erhalten (keine Archivierung): Wird diese Option aktiviert, wird keine Archivierung durchgeführt und keine Archivdatei angelegt. Die Versionshistorie bleibt komplett erhalten.

Versionshistorie mindestens 120 Tag(e) erhalten (partielle Archivierung): Über diese Option kann der Zeitraum festgelegt werden, für den die gesamte Versionshistorie mit allen Revisionen erhalten bleiben soll. Sind z. B. 120 Tage eingestellt, berücksichtigt die Archivierung nur Revisionen, die zu Beginn des Archivierungsauftrags **älter als** 120 Tage sind. Alle Änderungen, die zu Beginn des



Archivierungsauftrags **jünger als 120 Tage** sind, können auch nach der Archivierung noch lückenlos nachvollzogen werden.

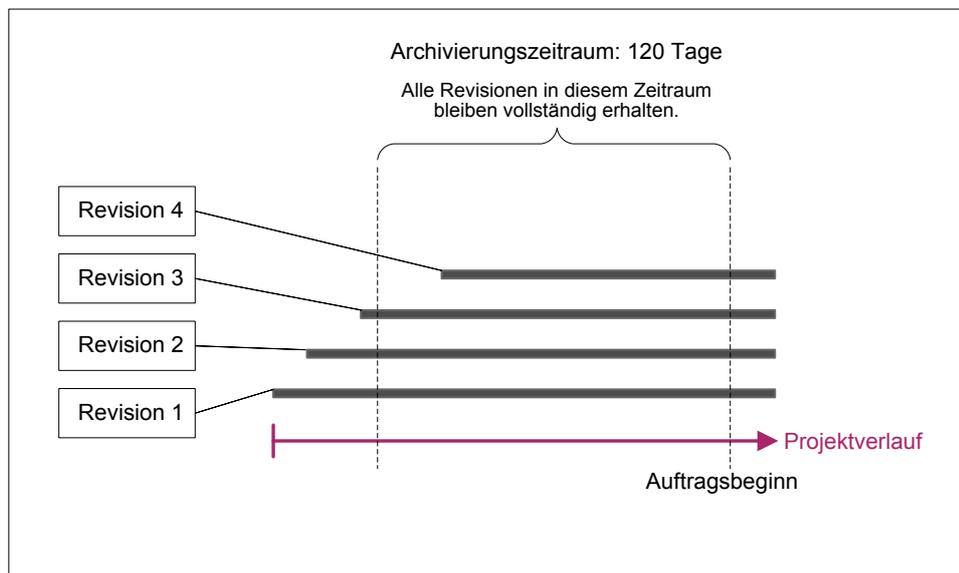


Abbildung 7-145: Partielle Archivierung über 120 Tage

Versionshistorie nicht erhalten (vollständige Archivierung): Wird diese Option aktiviert, wird die gesamte Versionshistorie bei der Ausführung des Auftrags berücksichtigt. Alle nicht mehr benötigten Daten (siehe dazu Kapitel 7.9.3 Seite 424) werden in die Archiv-Datei verschoben.

Objekte:

In diesem Bereich kann festgelegt werden, welche Art von Daten archiviert werden sollen.

Inhalte, Medien und Datenquellen: Ist diese Option aktiviert, werden alle Inhalte der Inhalte-, Medien- und Datenquellen-Verwaltung eines Projektes archiviert (also alles bis auf Vorlagen).





Es gibt eine Größenbeschränkung bei der Archivierung von Datenbankinhalten: Ist die Summe aller Datenbankeinträge größer als 90% von 8 GB, wird die Archivierung bezüglich der Datenbankeinträge abgebrochen.

In diesem Fall wird ein WARN (in der Datei fs-server.log) geloggt, z. B.

```
WARN 09.08.2010          12:50:42.080          {seID=369117}
(de.espirit.or.impl.AbstractSessionHandler): cleanup for entityType='cases'
aborted, tar entry size limit reached! [schema=P222005_222001]
```

Um die dann noch übrigen Datenbankeinträge zu archivieren, müssen weitere Archivierungsaufträge durchgeführt werden.

Vorlagen: Ist diese Option aktiviert, werden Vorlagen archiviert. Die Option kann nur in Verbindung mit "Inhalte, Medien und Datenquellen" gewählt werden.

Systemdaten: Systemdaten sind Informationen, die zu jeder Aktion im JavaClient (z. B. Anlegen oder Löschen von Objekten, Freigaben etc.) vom System erzeugt werden (siehe auch Kapitel 7.9.1 Seite 423 und Kapitel 7.9.2 Seite 423). Ist diese Option aktiviert, werden neben den nicht mehr verwendeten Systemdaten auch die Daten geschlossener Workflows und interne Revisions-Metadaten archiviert. Revisionen, zu denen keine Nutzerdaten mehr existieren, werden komplett (d.h. alle internen, zu dieser Revision gehörenden Dateien) gelöscht. Dazu gehören auch die UI-Dateien. Archivierte Revisionen werden später über die Funktion "Archiv" auf dem Register "Revisionen" angezeigt (siehe Abbildung 7-18).

Sollen auch geschlossene Aufgaben archiviert werden, müssen die Checkboxen "Inhalte, Medien und Datenquellen" **und** "Systemdaten" aktiviert sein. In diesem Fall werden alle zu einer Aufgabe gehörenden Dateien archiviert, die zu einem Zeitpunkt geschlossen wurde, der innerhalb des zu archivierenden Zeitraums liegt.

Optionen:

In diesem Bereich kann festgelegt werden, ob nur Objekte oder auch nicht mehr benötigte Einträge der Versionshistorie archiviert werden sollen.

nur als gelöscht markierte Objekte archivieren: Wird diese Option aktiviert, werden nur Objekte, die gelöscht wurden, archiviert.

gelöschte Objekte und nicht mehr benötigte Versionshistorie archivieren: Wird diese Option aktiviert, werden zusätzlich zu gelöschten Objekten auch nicht mehr benötigte Einträge der Versionshistorie archiviert. Das bedeutet, dass die Versionshistorie der Objekte reduziert wird. In jedem Fall bleiben allerdings die komplette Versionshistorie aus dem Zeitraum der partiellen Archivierung erhalten



(sofern diese Option ausgewählt ist), die Revisionen des letzten Freigabestandes (sofern vorhanden) sowie des aktuellen Bearbeitungsstandes.

Wartezeit je Archivierungsschritt: Bei umfangreichen Archivierungen kann über diesen Wert eine Pause (in Millisekunden) eingestellt werden, die jeweils zwischen den Archivierungsschritten eingelegt werden soll. Auf diese Weise kann die Belastung des Servers während der Archivierung reduziert werden.

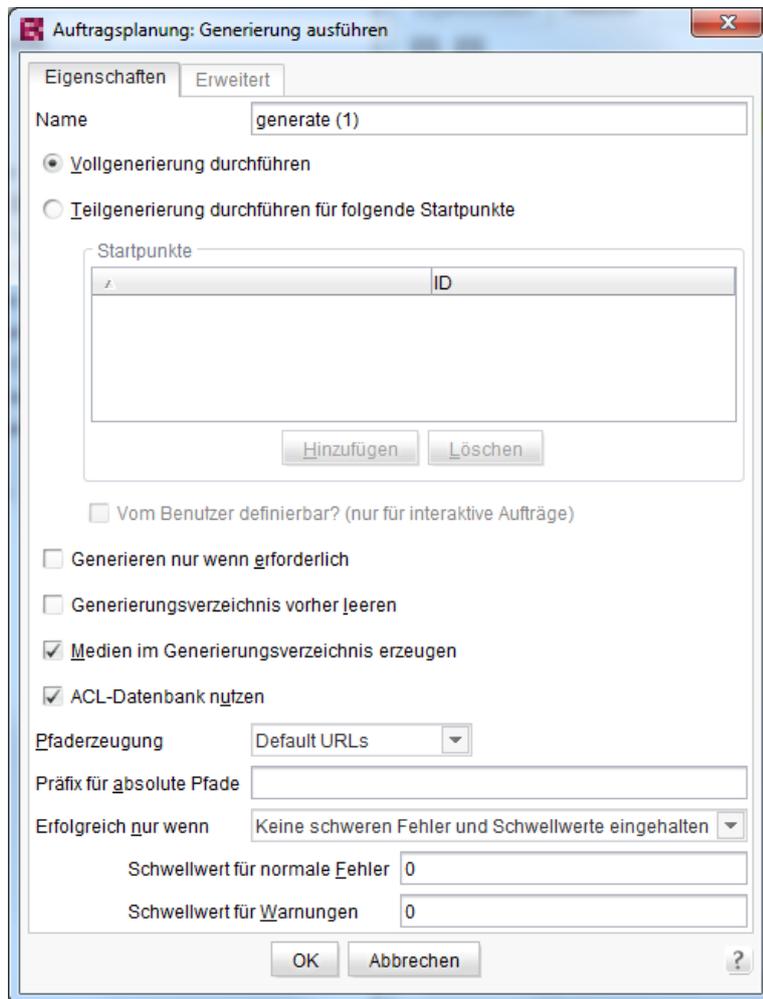
Laufzeit: Da umfangreiche Archivierungen einige Zeit in Anspruch nehmen können, kann die maximale Laufzeit einer Projektarchivierung eingeschränkt werden. Im Bereich "Laufzeit" kann festgelegt werden, wie viel Zeit die Archivierung in Anspruch nehmen darf.

Maximale Laufzeit der Archivierung pro Ausführung auf 60 Minuten beschränken: Hier kann eingestellt werden, nach wie vielen Minuten die Archivierung gestoppt werden soll. Standardmäßig sind 60 Minuten voreingestellt. Beim nächsten (manuellen oder automatischen) Starten der Aktion beginnt die Archivierung bei dem Stand, der beim Stoppen erreicht wurde. Für jede dieser "Teil-Archivierungen" wird dabei eine eigene Archivdatei erstellt.

Laufzeit der Archivierung nicht beschränken: Ist diese Option aktiviert, wird die Archivierung ohne Zeitlimit bis zum Ende durchgeführt.



7.5.10.2 Generierung ausführen

**Abbildung 7-146: Aktion anlegen – Generierung ausführen**

Die Aktion "Generierung ausführen" dient dazu, eine Voll- bzw. Teilgenerierung des gewählten Projekts durchzuführen (siehe auch Kapitel 8.4 Seite 446).

Eine Vollgenerierung generiert alle Inhalte des Projekts, eine Teilgenerierung nur die ausgewählten "Startpunkte" und deren Kinder.

Zu weiteren Informationen über die Generierung in FirstSpirit siehe auch siehe dazu FirstSpirit Online Dokumentation, "Weiterführende Themen" / "Generierung".



Register "Eigenschaften"

Vollgenerierung durchführen: Ist diese Option *aktiviert*, wird das Projekt bei der Auftragsausführung vollständig generiert.



Ergänzend zur zeitintensiven Vollgenerierung eines Projektes besteht die Möglichkeit, nur die geänderten Inhalte eines Projektes (bezogen auf die letzte erfolgreiche Generierung) über die FirstSpirit-Access-API zu ermitteln und nur diese zu generieren ("Delta-Generierung").

Eine Gegenüberstellung der einzelnen Generierungsarten sowie ein Konfigurationsbeispiel für eine Delta-Generierung findet sich in der FirstSpirit Online Dokumentation (unter Weiterführende Themen/Generierung).

Teilgenerierung durchführen für folgende Startpunkte: Ist diese Option *aktiviert*, werden nur die in der nachfolgenden Tabelle "Startpunkte" angezeigten Knoten generiert (einschließlich aller untergeordneten Startpunkte):

- Teilgenerierung durchführen für folgende Startpunkte

Startpunkte	
<i>x</i>	ID
 Homepage	819283
 header_home	818372
 Über uns	818327

Abbildung 7-147: Aktion anlegen – Teilgenerierung ausführen

Startpunkte: Diese Tabelle zeigt alle Startpunkte, die bei Ausführung der Aktion generiert werden sollen. Die ausgewählten Elemente werden rekursiv durchlaufen, wird also beispielsweise ein Medien-Ordner hinzugefügt, sind auch alle untergeordneten Elemente (Medien und weitere Ordner) ein Bestandteil des Generierungsauftrags.

Durch einen Klick auf den Button "Hinzufügen" öffnet sich ein Dialog, der alle verfügbaren und noch nicht ausgewählten Startpunkte des Projekts anzeigt. Dabei werden ausschließlich freigegebene Objekte eingeblendet.

Durch einen Klick auf den Button "Löschen" wird der selektierte Startpunkt gelöscht.



Vom Benutzer definierbar? (nur für interaktive Aufträge): Für eine Teilgenerierung kann die Option "Vom Benutzer definierbar" aktiviert werden. Durch das Aktivieren der Option kann der Projektadministrator dem Benutzer gestatten, eigene Startpunkte für die Generierung beim Ausführen des Auftrags festzulegen. Diese Änderungen wirken sich nur auf die aktuelle Generierung aus.

Innerhalb eines Generierungsauftrags können auch Startpunkte für die Generierung von Medien definiert werden.

Hintergrund: Beim Veröffentlichen von Projekthinhalten werden referenzierte Medien, beispielsweise innerhalb einer Bildeingabekomponente auf einer Seite, ebenfalls generiert und veröffentlicht. Medien, die nicht explizit referenziert werden, werden bei der Generierung nicht automatisch berücksichtigt. In einigen Anwendungsfällen, beispielsweise wenn Medien innerhalb eines Skriptes verwendet werden, müssen auch diese Medien in der Generierung enthalten sein.



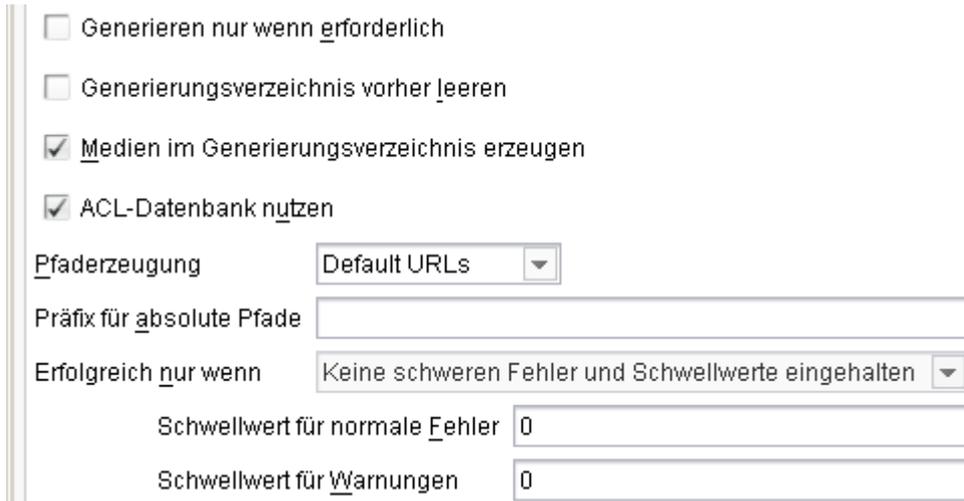
Medien, für die die Option "Datei parsen" aktiviert ist (siehe FirstSpirit Handbuch für Redakteure), werden nicht generiert/veröffentlicht, wenn für sie die Option "Alle Medien in diesem Ordner bei der Generierung kopieren" aktiviert ist.

Die Generierung von Medien können einzelnen Generierungsaufträgen zugeordnet werden. Zielsetzung der neuen Option ist es, eine schnellere Veröffentlichung einzelner Teilbereiche zu ermöglichen. Das Auslagern der rechenzeitintensiven Generierung von Medien in separate Generierungsaufträge bietet, insbesondere beim Ausführen mehrerer kleiner Teilgenerierungen, einen hohen Performancegewinn, da für jede Generierung ausschließlich die gewünschten Medien berücksichtigt werden. Diese Option eignet sich aber auch ideal, um beispielsweise Medien in einem Remote-Projekt zu veröffentlichen ("Remote-Medien").



Medien, für die die Option "Datei parsen" aktiviert ist (siehe FirstSpirit Handbuch für Redakteure), werden nicht generiert/veröffentlicht, wenn sie unterhalb eines gewählten Startpunkts liegen und auf keiner Seite der Generierung referenziert werden.





Generieren nur wenn erforderlich

Generierungsverzeichnis vorher leeren

Medien im Generierungsverzeichnis erzeugen

ACL-Datenbank nutzen

Pfaderzeugung

Präfix für absolute Pfade

Erfolgreich nur wenn

Schwellwert für normale Fehler

Schwellwert für Warnungen

Abbildung 7-148: Ausschnitt (siehe Abbildung 7-146)

Generieren nur wenn erforderlich: Ist diese Option *aktiviert*, wird vor der Generierung geprüft, ob es seit der letzten Ausführung Änderungen an dem Projekt gab. Die Generierung wird nur durchgeführt, wenn im Projekt eine neue Revision erzeugt wurde (vgl. auch Delta-Generierung).

Bitte beachten Sie, dass die Verwendung dieser Option zusammen mit dem manuellen Erzeugen einer neuen Revision vor der Generierung - was in manchen Konstellationen notwendig ist - keinen Sinn macht!

Generierungsverzeichnis vorher leeren: Ist diese Option *aktiviert*, wird das Generierungsverzeichnis vor Beginn der Generierung geleert.

Medien im Generierungsverzeichnis erzeugen:

Ist diese Option *aktiviert*, werden alle referenzierten Medien bei der Generierung automatisch im Generierungsverzeichnis erzeugt.

Ist die Option *deaktiviert*, werden bei der Generierung keine Medien im Generierungsverzeichnis erzeugt (Ausnahme: geparste Medien). Damit kann die I/O-Last und der beanspruchte Plattenplatz im Generierungsverzeichnis reduziert werden. Eine Übertragung der Medien wird durch diese Option *nicht* unterbunden, da bei einer Veröffentlichung über FirstSpirit-interne Veröffentlichungsmechanismen (in Dateisystem, per FTP oder per CRC, siehe Kapitel 7.5.10.6 Seite 393) alle referenzierte Medien, die nicht im Generierungsverzeichnis vorliegen, über das BerkeleyDB-Backend gestreamt werden. Dies gilt nicht für andere, externe Übertragungsmechanismen (z. B. via rsync, siehe Kapitel 10, Seite 506). (Hinweis: Ist die Option deaktiviert, ist das Generierungsverzeichnis (fs5staging) nicht "browserbar".)



ACL-Datenbank nutzen: Ist diese Option *aktiviert*, werden bei einer Generierung Informationen – für jede Seitenreferenz in der Struktur-Verwaltung und für jedes Medium in der Medien-Verwaltung – in einer lokalen Datenbank hinterlegt, der so genannten FirstSpirit-Access-Control-Datenbank, kurz ACL-Datenbank. Diese Datenbank dient zur Bereitstellungen von Informationen zu FirstSpirit-Objekten, beispielsweise zur Bereitstellung von Zugriffsrechten, die für ein Objekt gespeichert wurden. Die Synchronisation der Access-Control-Datenbank mit dem aktuell freigegebenen Projektstand erfolgt automatisch bei der Generierung der Inhalte.

Weiterführende Informationen zur ACL-Datenbank siehe "Dokumentation zum Modul FirstSpirit Security".

Pfaderzeugung: Aus dieser Liste wird eine Methode zur Pfaderzeugung ausgewählt. Aktuell gibt es vier verschiedene Arten der Erzeugung:

- **Advanced URLs:** Bei dieser Methode werden die Anzeigenamen der FirstSpirit-Objekte für die URL-Erzeugung zugrunde gelegt (siehe dazu FirstSpirit Online Dokumentation, "Weiterführende Themen" / "Generierung").
- **Default URLs:** Bei dieser Methode existiert für jede Sprache des Projekts ein eigener Unterordner auf dem Webserver (de, en, etc., siehe dazu FirstSpirit Online Dokumentation, "Weiterführende Themen" / "Generierung").
- **Multiview URLs:** Bei dieser Methode existieren keine sprachspezifischen Unterordner, stattdessen werden die Dateien für jede Sprache mit dem jeweiligen Sprachkürzel gekennzeichnet. Dabei werden die Sprachkürzel *nach* der Dateierweiterung eingefügt (z. B. index.html.de, index.html.en) (vgl. Kap. 7.4.2).
- **Infix URLs:** Bei dieser Methode existieren ebenfalls keine sprachspezifischen Unterordner, stattdessen werden die Dateien für jede Sprache mit dem jeweiligen Sprachkürzel gekennzeichnet. Dabei werden die Sprachkürzel *vor* der Dateierweiterung eingefügt (z. B. index.de.html, index.en.html).

Präfix für absolute Pfade: Der hier eingegebene Präfix wird allen Links vorangestellt, welchen in einem Template (im JavaClient) die Eigenschaft eines absoluten Links zugewiesen wird.

Erfolgreich nur wenn: Es gibt 2 Optionen zu entscheiden, wann eine Generierung erfolgreich ist und wann nicht:

- Keine schweren Fehler und Schwellwerte eingehalten
- Keine schweren Fehler

In beiden Fällen wird die Generierung abgebrochen, sobald ein schwerwiegender Fehler auftritt. Treten keine schweren Fehler auf, gilt die Generierung bei der Option



"Keine schweren Fehler" immer und bei der Option "Keine schweren Fehler und Schwellwerte eingehalten" nur als erfolgreich, wenn die folgenden Schwellwerte nicht überschritten werden:

- Schwellwert für normale Fehler
- Schwellwert für Warnungen

Register "Erweitert"

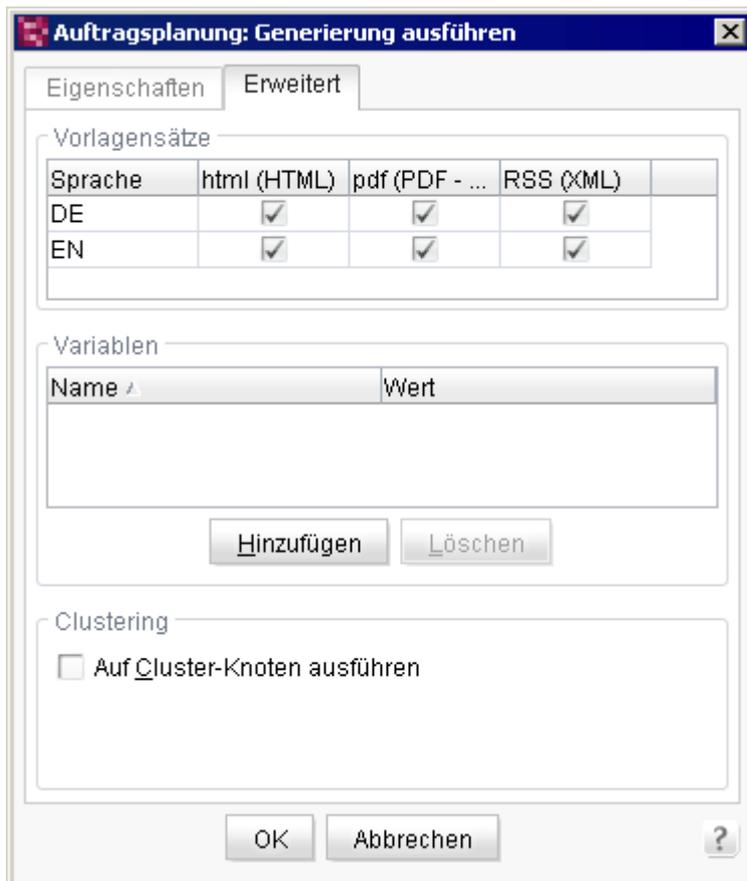


Abbildung 7-149: Generierung ausführen – Register "Erweitert"

Vorlagensätze: Über diesen Bereich kann definiert werden, für welchen Vorlagensatz in welcher Sprache die Generierung durchgeführt werden soll. Siehe dazu Kapitel 7.4.5 Seite 295 (Sprachen) und Kapitel 7.4.13 Seite 313 (Vorlagensätze).

Variablen: Über dieses Feld können Variablen im Generierungsauftrag gesetzt werden, die bei der Generierung eines Projekts ausgelesen werden. Sie sind im gesamten Projekt verfügbar, so dass z. B. in Seiten- und Absatzvorlagen auf diese Variablen zugegriffen werden kann. In Generierungsaufträgen gesetzte Variablen können darüber hinaus z. B. Variablen überschreiben, die im Projekt auf



Menüebenen definiert wurden. Siehe dazu auch *FirstSpirit Online Dokumentation*, "Vorlagenentwicklung" / "Variablen" / "Kontexte" / "projektbezogen".

Variablen für den Generierungsauftrag können mit Bezeichner und Wert über die Schaltfläche "Hinzufügen" definiert werden. Bereits vorhandene Variablen können über die Schaltfläche "Löschen" wieder entfernt werden. Zur Verwendung von Variablen in FirstSpirit (Definition, Ausgabe, Vererbung, Überschreiben) siehe auch *FirstSpirit Online Dokumentation*, "Vorlagenentwicklung" / "Variablen".

Clustering: siehe dazu Kapitel 7.6.4 Seite 406.

Über den Aufruf `#global.scheduleContext` kann auf den Ausführungskontext eines Generierungsauftrags zugegriffen werden und damit auf Informationen wie z. B. den Projektnamen, den Start-Zeitpunkt des Auftrags, den Pfad zum Generierungsverzeichnis, die Fehleranzahl bei der Generierung usw. Das Interface `de.espirit.firstspirit.access.schedule.ScheduleContext`, das die notwendigen Methoden bereitstellt, ist in der FirstSpirit Access-API verfügbar.

7.5.10.3 Projektsicherung durchführen

Die Projektsicherung dient dazu, einen aktuellen Stand des Projekts zu exportieren. Die Projektsicherung wird automatisch in dem Pfad angelegt, der in der Konfigurationsdatei `fs-server.conf` als Backup-Path definiert wurde. (Dabei ist es möglich, das Verzeichnis auf eine andere Festplatte auszulagern (siehe Parameter `BACKUP_PATH`, siehe Kapitel 4.3.1.15 Seite 58).) Das Löschen nicht mehr für die Projektsicherung benötigter Exportdateien erfolgt über die Funktion "Server aufräumen" (siehe Kapitel 7.2.2.1 Seite 206).

Das lizenzabhängige Modul "FirstSpirit Enterprise Backup" ermöglicht eine effiziente Datensicherung über inkrementelle und differentielle Backups. Alle Änderungen am Projekt werden dabei ab einem definierten Startpunkt ("SnapShot") nur noch separat gesichert. Bei Bedarf kann dann aus der initialen Backup-Datei ("SnapShot"-Export) und den jeweiligen Änderungs-Backup-Dateien ein vollständiges Backup erstellt werden.

Weiterführende Informationen siehe "Dokumentation zum Modul FirstSpirit Enterprise Backup".



7.5.10.4 Referenzen reparieren

Sofern in einem Projekt defekte Referenzen vorliegen, kann über diese Aktion eine Neuberechnung der Referenzen im Projekt gestartet werden. Nach dem Start des Projektauftrags erfolgt die Meldung: "Der Auftrag wurde erfolgreich ausgeführt!". Die Referenzberechnung wurde zu diesem Zeitpunkt zwar bereits gestartet, muss aber noch nicht abgeschlossen sein. Das Berechnen der Referenzen läuft im Hintergrund ab und kann abhängig von der Projektgröße eine gewisse Zeit andauern.

7.5.10.5 Suchindex neu aufbauen

Die Elemente aus den FirstSpirit-Verwaltungsbereichen (z. B. Seiten, Absätze, Mediendateien) werden beim Anlegen, Bearbeiten oder Löschen in den Suchindex geschrieben bzw. aus diesem entfernt. Über diese Aktion ist es möglich, den Suchindex neu zu erstellen. Je nach Projektgröße kann dies einige Zeit in Anspruch nehmen. Nach der Durchführung der Aktion erfolgt die Meldung: "Der Auftrag wurde erfolgreich ausgeführt!"

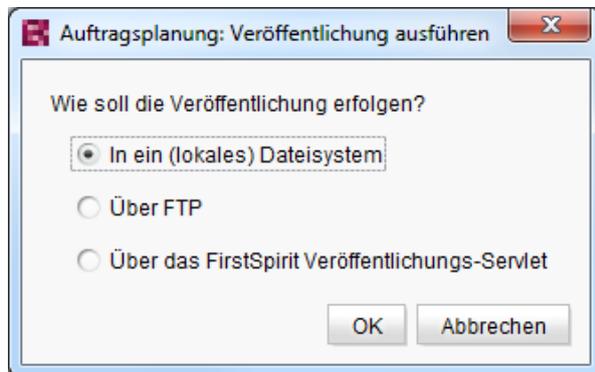
Die Neu-Indizierung mehrerer Projekte wird sequentiell abgearbeitet.

Für weitere Informationen siehe Kapitel 9.18 Seite 500

Hinweis: Neben der Möglichkeit den gesamten Suchindex für ein Projekt neu zu erstellen, kann der Suchindex ab FirstSpirit Version 5.0R3 nur für einzelne Teilbereiche eines Projekts aktualisiert werden. Die entsprechenden Methoden werden über die FirstSpirit Access-API bereitgestellt und können über ein Skript ausgeführt werden (Einstiegspunkt FirstSpirit Access-API: Interface ProjectStorage (Package: firstspirit.access.admin)).



7.5.10.6 Veröffentlichung ausführen

**Abbildung 7-150: Aktion anlegen – Veröffentlichung ausführen**

Damit eine neue Veröffentlichung erstellt werden kann, muss zunächst die Veröffentlichungsart ausgewählt werden. Aktuell gibt es drei verschiedene Möglichkeiten:

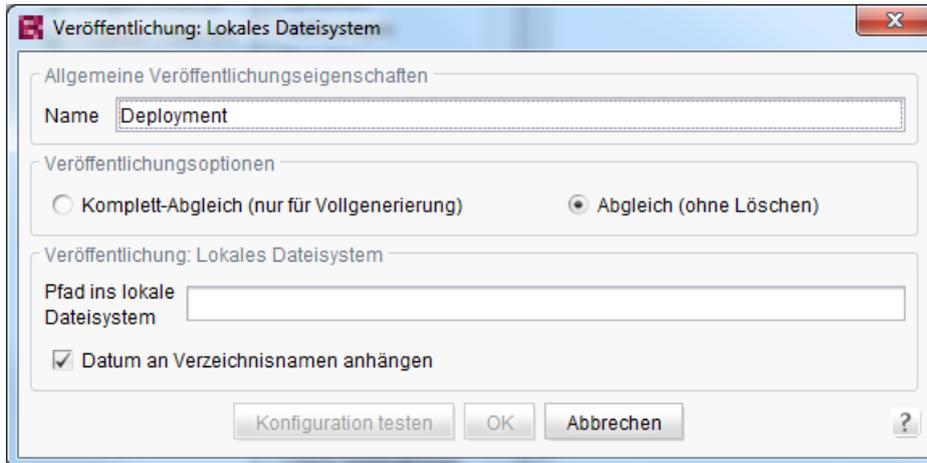
- Veröffentlichung in ein lokales Dateisystem (siehe Kapitel 7.5.10.6.1 Seite 394)
- Veröffentlichung über FTP (siehe Kapitel 7.5.10.6.2 Seite 396)
- Veröffentlichung über ein FirstSpirit Veröffentlichungs-Servlet (siehe 7.5.10.6.3 Seite 399)



Handelt es sich nicht um das Anlegen einer Veröffentlichung, sondern um das Bearbeiten einer vorhandenen Veröffentlichung, wird dieser Dialog übersprungen und direkt die entsprechende typenabhängige Eingabemaske geöffnet.



7.5.10.6.1 Veröffentlichung in ein lokales Dateisystem

**Abbildung 7-151: Aktion anlegen – Veröffentlichung in ein lokales Dateisystem**

Diese Aktion dient zur Projekt-Veröffentlichung in ein lokales Dateisystem und erfordert daher auch nicht viele Einstellungen.

Name: Der Name der Aktion, welcher in der Auftragsübersicht, der Auftragsverwaltung und den Aktionsvorlagen angezeigt wird.

Veröffentlichungsoptionen**Komplett-Abgleich (nur für Vollgenerierung)**

Bei dieser Methode wird auf dem Webserver ein mit dem Generierungsstand identischer Datenbestand hergestellt. Das heißt: Auf dem Entwicklungsserver nicht mehr vorhandene Dateien werden auch auf dem Webserver gelöscht, neue Dateien werden kopiert und alte, bereits vorhandene Dateien werden von den neuen überschrieben.

Abgleich (ohne Löschen)

Diese Veröffentlichungsoption arbeitet auf ähnliche Weise wie der Komplettabgleich, mit dem Unterschied, dass auf dem Webserver keine Dateien gelöscht werden, auch wenn sie auf dem Entwicklungsserver nicht mehr vorhanden sind.

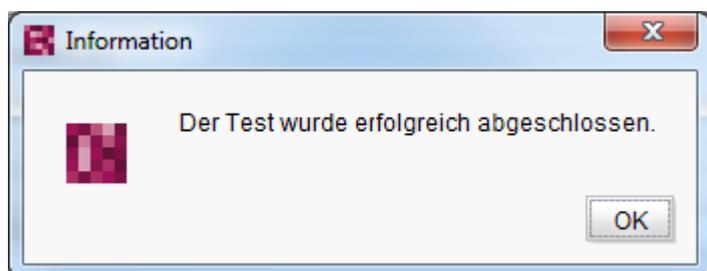
Veröffentlichung

Pfad ins lokale Dateisystem: Hier wird festgelegt, in welches lokale Verzeichnis veröffentlicht werden soll. Der Pfad kann absolut oder relativ (zum Arbeitsverzeichnis des Servers) angegeben werden. Die Angabe eines Pfades aktiviert außerdem den OK-Button und den Button "Konfiguration testen".



Datum an Verzeichnisnamen anhängen: Ist diese Option *aktiviert*, wird den oben angegebenen Pfaden das aktuelle Datum angehängen.

Per Klick auf den Button "Konfiguration testen", der durch die Angabe eines "Pfad ins lokale Dateisystem" aktiv wird, kann die eingestellte Konfiguration getestet werden. Erst wird überprüft ob alle benötigten Parameter angegeben sind. Im Anschluss versucht der Server in dem angegebenen "Pfad ins lokale Dateisystem" einen Ordner zu erstellen, eine Datei dort hinein zu schreiben, diese dann umzubenennen und im Anschluss alle Daten wieder zu löschen. Ist der Test erfolgreich, erscheint folgende Meldung:



Im Fehlerfall erscheint diese Meldung:



Per Klick auf den Button "Details" öffnet sich ein Dialog mit entsprechender Log-Datei, welche die aufgetretenen Fehler aufzeigt.

Zum Konzept "Sichere Medien" siehe "Dokumentation zum Modul FirstSpirit Security".



7.5.10.6.2 Veröffentlichung über FTP

The screenshot shows a Windows-style dialog box titled "Veröffentlichung: FTP". It is divided into several sections:

- Allgemeine Veröffentlichungseigenschaften:** A text field labeled "Name" contains the value "Deployment".
- Veröffentlichungsoptionen:** Two radio buttons are present. The first is "Komplett-Abgleich (nur für Vollgenerierung)" and the second is "Abgleich (ohne Löschen)", which is selected.
- Veröffentlichung: FTP:** This section contains fields for "FTP-Server", "Benutzer", "FTP-Server-Typ" (a dropdown menu currently showing "UNIX"), and "Passwort". There is also a checkbox for "Passiv Modus".
- Basis Pfad auf FTP-Server:** An empty text input field.
- FTP-Proxy-Einstellungen:** A checkbox for "FTP-Proxy verwenden?". Below it are fields for "Proxy-Server" and "Port". Two radio buttons are also present: "USER@SITE"-Protokoll and "USER with Login"-Protokoll (selected). Below these are fields for "Benutzer" and "Passwort".

At the bottom of the dialog, there are four buttons: "Konfiguration testen", "OK", "Abbrechen", and a help icon (question mark).

Abbildung 7-152: Aktion anlegen – Veröffentlichung per FTP

Diese Aktion dient zur Projekt-Veröffentlichung per FTP-Verbindung zu einem entfernten Server und erfordert daher eine Reihe von Einstellungen.

Allgemeine Veröffentlichungseigenschaften

Name: Der Name der Aktion, welcher in der Auftragsübersicht, der Auftragsverwaltung und den Aktionsvorlagen angezeigt wird.

Veröffentlichungsoptionen

Komplett-Abgleich (nur für Vollgenerierung)

Bei dieser Methode wird auf dem Webserver ein mit dem Generierungsstand identischer Datenbestand hergestellt. Das heißt: Auf dem Entwicklungsserver nicht mehr vorhandene Dateien werden auch auf dem Webserver gelöscht, neue Dateien werden kopiert und alte, bereits vorhandene Dateien werden von den neuen überschrieben.



Abgleich (ohne Löschen)

Diese Veröffentlichungsoption arbeitet auf ähnliche Weise wie der Komplettabgleich, mit dem Unterschied, dass auf dem Webserver keine Dateien gelöscht werden, auch wenn sie auf dem Entwicklungsserver nicht mehr vorhanden sind.

FTP-Server Einstellungen

FTP-Server: Die Adresse des FTP-Servers, auf den die Daten übertragen werden sollen. Diese Angabe ist zwingend erforderlich.

FTP-Server-Typ: In dieser Combobox wird das Betriebssystem des FTP-Servers ausgewählt.

Benutzer: Hier wird der Benutzer festgelegt, mit dem sich der Server auf dem FTP-Server anmelden soll.

Passwort: Das Passwort wird im Zusammenhang mit dem Benutzernamen zur Anmeldung auf dem FTP-Server verwendet. Wird ein Benutzername angegeben, ist auch diese Angabe zwingend erforderlich.

Passiv Modus: Ist diese Option *aktiviert* wird die Verbindung zu einem vom FTP-Server gewählten Port aufgebaut. Dieser Modus sollte eingesetzt werden, wenn sich der CMS-Server hinter einem Router befindet oder eine Firewall das Netzwerk vor Zugriffen von außen abschirmt.

Basis Pfad auf FTP-Server: Hier wird der Pfad zu dem Verzeichnis auf dem FTP-Server angegeben, in welches die generierten Daten übertragen werden sollen.

FTP-Proxy Einstellungen

FTP-Proxy verwenden?: Wird diese Option *aktiviert* wird die Verbindung zum FTP-Server über den, in diesem Bereich eingestellten, Proxy-Server hergestellt.

Proxy Server: Die Adresse des FTP-Proxy-Servers, über den die Verbindung hergestellt werden soll. Diese Angabe ist zwingend erforderlich, wenn ein FTP-Proxy verwendet wird.

Port: Der Port zum oben angegebenen Proxy-Server. Diese Angabe ist zwingend erforderlich, wenn ein FTP-Proxy verwendet wird.

"USER@SITE"-Protokoll: Ist diese Option *aktiviert* erfolgt die Authentifizierung beim Proxy per "USER@SITE"-Protokoll, also ohne gesonderte Benutzerkennung.

"User with Login"-Protokoll: Ist diese Option *aktiviert* erfolgt die Authentifizierung beim Proxy per "User with Login"-Protokoll mit folgender Benutzerkennung. Die

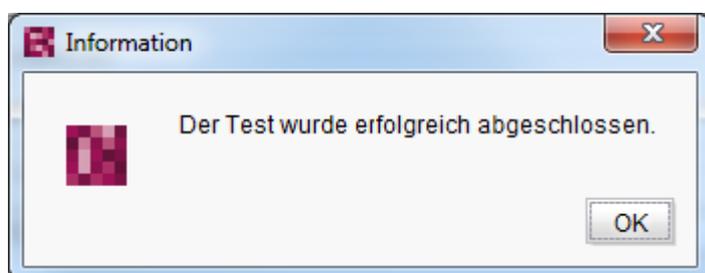


Angabe von Benutzername und Kennwort sind in diesem Fall zwingend erforderlich.

Benutzer: Der Benutzername, mit dem die Verbindung zum FTP-Proxy hergestellt werden soll. Diese Angabe ist zwingend erforderlich.

Passwort: Das Passwort wird im Zusammenhang mit dem Benutzernamen zur Anmeldung beim FTP-Proxy verwendet und ist daher auch zwingend erforderlich.

Per Klick auf den Button "Konfiguration testen" kann die eingestellte Konfiguration getestet werden. Erst wird überprüft, ob alle benötigten Parameter angegeben sind. Im Anschluss versucht der Server sich mit dem FTP-Server zu verbinden und im angegebenen "Basis Pfad auf FTP-Server" einen Ordner zu erstellen, eine Datei dort hinein zu schreiben, diese dann umzubenennen und im Anschluss alle Daten wieder zu löschen. Ist der Test erfolgreich, erscheint folgende Meldung:



Im Fehlerfall erscheint diese Meldung:



Per Klick auf den Button "Details" öffnet sich ein Dialog mit entsprechender Log-Datei, welche die aufgetretenen Fehler aufzeigt.

Zum Konzept "Sichere Medien" siehe "Dokumentation zum Modul FirstSpirit Security".



7.5.10.6.3 Veröffentlichung per FirstSpirit Veröffentlichungs-Servlet

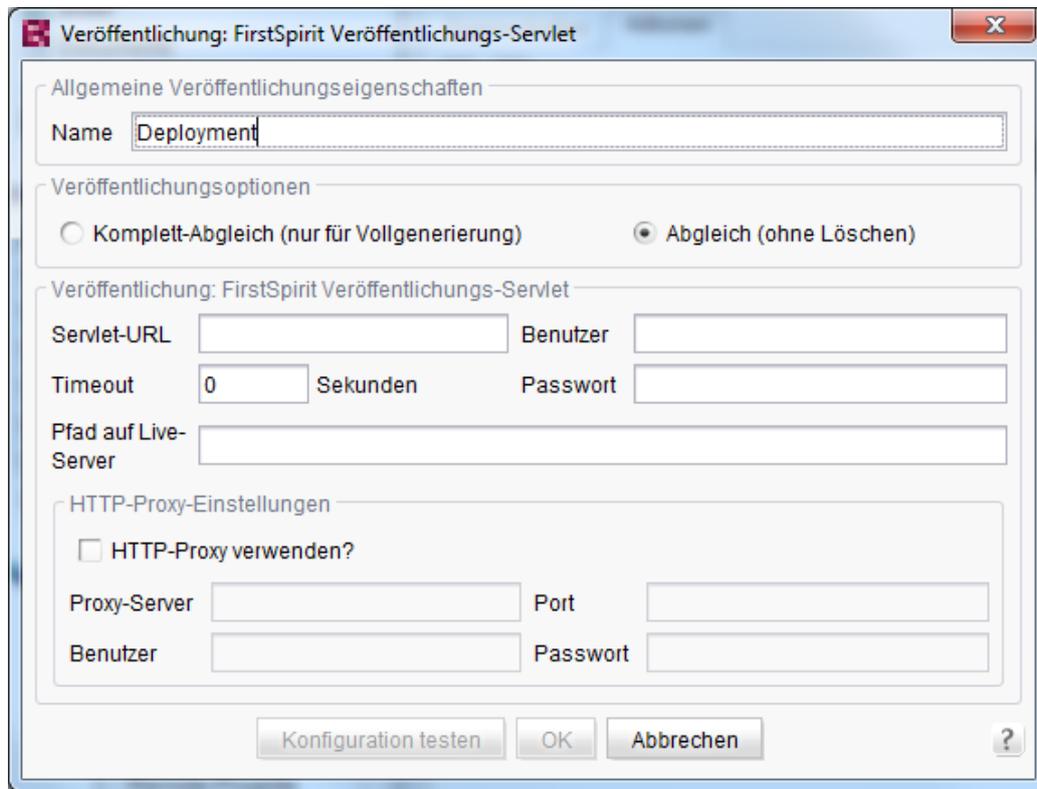


Abbildung 7-153: Aktion anlegen – Veröffentlichung per FirstSpirit Veröffentlichungs-Servlet

Diese Aktion dient zur Projekt-Veröffentlichung über ein FirstSpirit Veröffentlichungs-Servlet. Aufgabe des Servlets ist der Abgleich von Projekt-Dateien zwischen dem FirstSpirit-Server und dem Live-System. Anhand der CRC- Prüfsummenberechnung können neue, geänderte oder gelöschte Dateien ermittelt und nur diese Dateien aktualisiert werden. Dieser differentielle Upload beschleunigt den Aktualisierungsvorgang im Live-System. (Die dazu erforderlichen Informationen können u.a. aus einer Access-Control-Datenbank³³ ausgelesen werden, die die CRC-Prüfsumme zu allen Objekten verwaltet. Die Verwendung der Access-Control-Datenbank ist jedoch keine Voraussetzung für den Einsatz des CRC-Servlets).

³³ Das Servlet sowie die Funktionalität der Access-Control-Datenbank wird über das Modul FirstSpirit Security zur Verfügung gestellt und kann über den Konfigurationsdialog der zugehörigen Webanwendung "FS Security WebApp" projektspezifisch angepasst werden.



Allgemeine Veröffentlichungseigenschaften

Name: Der Name der Aktion, welcher in der Auftragsübersicht, der Auftragsverwaltung und den Aktionsvorlagen angezeigt wird.

Veröffentlichungsoptionen

Komplett-Abgleich (nur für Vollgenerierung)

Bei dieser Methode wird auf dem Webserver ein mit dem Generierungsstand identischer Datenbestand hergestellt. Das heißt: Auf dem Entwicklungsserver nicht mehr vorhandene Dateien werden auch auf dem Webserver gelöscht, neue Dateien werden kopiert und alte, bereits vorhandene Dateien werden von den neuen überschrieben.

Abgleich (ohne Löschen)

Diese Veröffentlichungsoption arbeitet auf ähnliche Weise wie der Komplettabgleich, mit dem Unterschied, dass auf dem Webserver keine Dateien gelöscht werden, auch wenn sie auf dem Entwicklungsserver nicht mehr vorhanden sind.

Servlet Einstellungen

Servlet-URL: Hier wird die vollständige Adresse zum FirstSpirit Veröffentlichungs-Servlet eingetragen. Diese Angabe ist zwingend erforderlich. Nach der Konfiguration der Datei `crcTransfer.ini` (über die Webanwendung FirstSpirit Security WebApp (Modul FirstSpirit Security)) wird das Servlet-Mapping in die Datei `web.xml` der Anwendung übernommen. Standardmäßig wird das Servlet auf `*.CRCTransfer` gemappt.

Zum Testen der Erreichbarkeit des Servlets, kann das Servlet im Webbrowser aufgerufen werden, z. B. über:

`http://www.mydomain.de/fs5staging_1921116/do.CRCTransfer`

Das Servlet wird eine Fehlermeldung anzeigen, da im Browser keine Anmeldedaten übermittelt wurden.

Benutzer: Hier wird der Benutzer festgelegt, mit dem sich der Server beim Servlet anmelden soll. Diese Angabe ist zwingend erforderlich.

Passwort: Das Passwort wird im Zusammenhang mit dem Benutzernamen zur Anmeldung beim Servlet verwendet. Diese Angabe ist zwingend erforderlich.

Timeout: Ist das Veröffentlichungsservlet nicht erreichbar, gibt dieser Wert an, nach wie viel Sekunden der Kommunikationsversuch abgebrochen wird.

Pfad auf Live-Server: Hier wird der Pfad zu dem Verzeichnis auf dem entfernten Server angegeben, in das die generierten Daten übertragen werden sollen. Das hier



angegebene Verzeichnis kann potentiell gelöscht werden ("Optionen – Generierungsverzeichnis vorher leeren" – siehe Kapitel 7.5.10.2 Seite 385). Es sollte daher nie das Web-Applikationsverzeichnis angegeben werden, sondern besser ein beliebiges Unterverzeichnis.

Um den Zugriffsschutz über das Modul FirstSpirit Security zu gewährleisten (siehe Kapitel 11.3.2 Seite 525), muss zusätzlich das Präfix für die Access-Control-Datenbank angegeben werden (siehe Kapitel 7.4.16 Seite 323).

HTTP-Proxy Einstellungen

HTTP-Proxy verwenden?: Wird diese Option *aktiviert*, wird die Verbindung zum Veröffentlichungs-Servlet über den in diesem Bereich eingestellten Proxy-Server hergestellt.

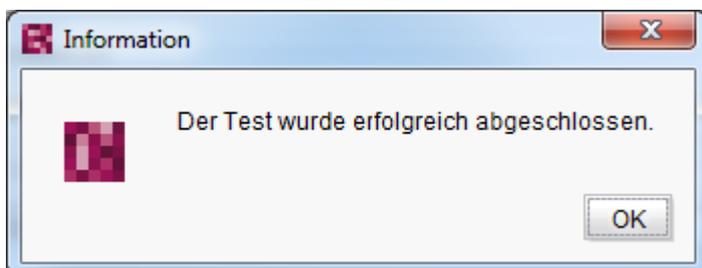
Proxy Server: Die Adresse des Proxy-Servers, über den die Verbindung hergestellt werden soll. Diese Angabe ist zwingend erforderlich.

Port: Der Port zum oben angegebenen Proxy-Server. Diese Angabe ist zwingend erforderlich.

Benutzer: Der Benutzername, mit dem die Verbindung zum HTTP-Proxy hergestellt werden soll.

Passwort: Das Passwort wird im Zusammenhang mit dem Benutzernamen zur Anmeldung beim HTTP-Proxy verwendet. Diese Angabe ist zwingend erforderlich sofern ein Benutzername angegeben wurde.

Per Klick auf den Button "Konfiguration testen" kann die eingestellte Konfiguration getestet werden. Erst wird überprüft, ob alle benötigten Parameter angegeben sind. Im Anschluss versucht der Server, sich mit dem Veröffentlichungs-Servlet zu verbinden und im angegebenen "Pfad auf Live-Server" einen Ordner zu erstellen, eine Datei dort hinein zu schreiben, diese dann umzubenennen und im Anschluss alle Daten wieder zu löschen. Ist der Test erfolgreich, erscheint folgende Meldung:



Im Fehlerfall erscheint diese Meldung:





Per Klick auf den Button "Details" öffnet sich ein Dialog mit entsprechender Log-Datei, welche die aufgetretenen Fehler aufzeigt.

7.6 Clustering: Lastverteilung bei der Generierung

FirstSpirit unterstützt einerseits eine "vertikale Skalierung", d.h. durch das Hinzufügen von Ressourcen, wie mehr CPUs oder eine Erhöhung des Hauptspeichers, ist eine Erhöhung der Systemleistung möglich, da in starkem Umfang Verfahren wie Multithreading und Caching eingesetzt werden.

Andererseits unterstützt FirstSpirit auch eine "horizontale Skalierbarkeit".

Ein Aspekt dieser horizontalen Skalierbarkeit ist die Lastverteilung bei der Generierung der FirstSpirit-Inhalte auf die Mitglieder des Cluster-Verbundes. Die Aufteilung der Generierung erfolgt auf Auftragsebene. Die Generierungsaktionen (innerhalb eines oder mehrerer Aufträge) können auf die Clusterknoten verteilt werden. Dabei wird eine Generierungsaktion jeweils vollständig auf einem Clusterknoten abgearbeitet. Weitere, parallel anstehende Generierungsaktionen können auf weitere Clusterknoten verteilt werden. Die Skalierung der Aktionen über die Clusterknoten erfolgt automatisch. Sofern eine gültige Lizenz für die Funktionalität besteht, können die benötigten Einstellungen über die FirstSpirit Server- und Projektkonfiguration definiert werden.

Neben der Vorschauerzeugung ist die Generierung von Inhalten eine der zeitkritischsten Operationen in einer FirstSpirit Umgebung. Dabei treffen hohe Anforderungen an die Rechenleistung und der Wunsch nach einer möglichst geringen Wartezeit zusammen. Zielsetzung der FirstSpirit-Funktionalität "Clustering" ist eine Erhöhung der Performance in Multi-User-Umgebungen. Zu diesem Zweck wird die rechenintensive Generierung vom Master-Server auf einen (oder mehrere) andere Server (Generation-Server) verlagert.

Die folgenden Kapitel beschreiben zunächst den Einsatz der Funktionalität bei der Generierung von FirstSpirit-Inhalten.



7.6.1 Konzept

Im Folgenden wird davon ausgegangen, dass die FirstSpirit-Serverdienste auf unterschiedlichen Systemen ablaufen sollen und im Bereich des Application-Servers eine Cluster-Lösung zum Einsatz kommt.

Es ergibt sich dann die in der folgenden Abbildung dargestellte Architektur:

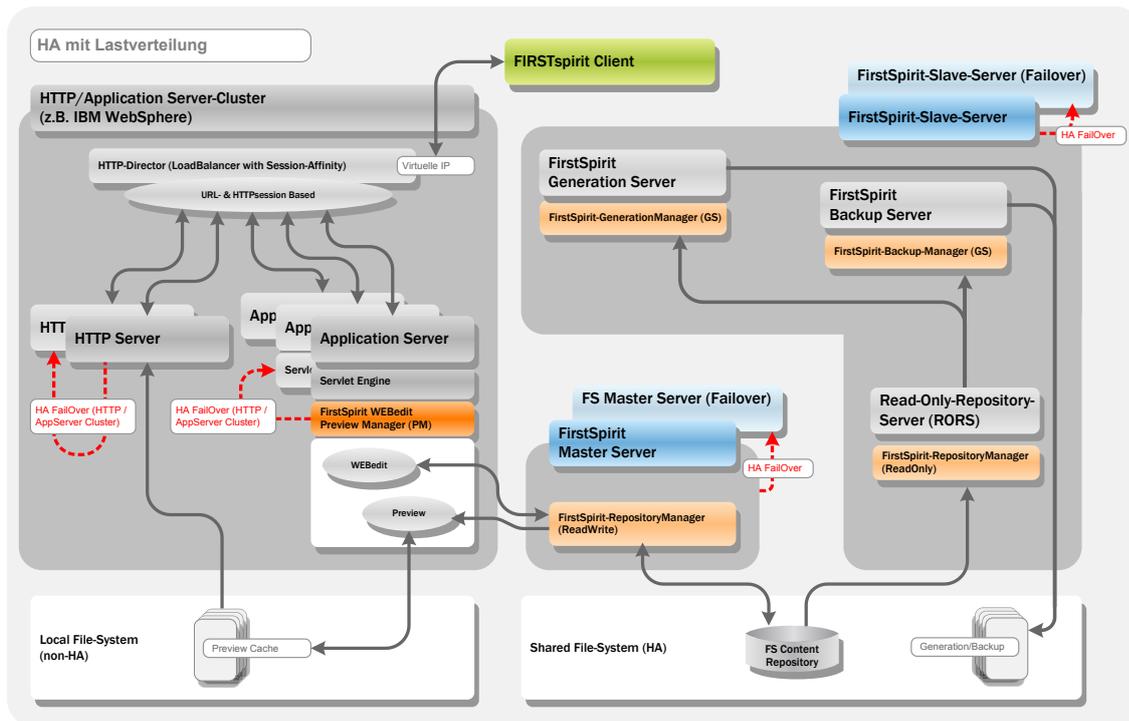


Abbildung 7-154: Hochverfügbarkeitscluster mit Lastverteilung

Der FirstSpirit-Client (sowohl der JavaClient als auch der WebClient) kommuniziert über HTTP(S) mit dem geclusterten Web-Application-Server. Über den HTTP(S)-Load-Balancer werden die FirstSpirit-Sitzungen auf die einzelnen Application-Server des Clusters verteilt. Eine Lastverteilung kann dabei auf URL-Basis und/oder auf HTTP-Session-Basis erfolgen.

Hinter dem Application-Server-Cluster wird eine Reihe von FirstSpirit-Diensten auf verschiedenen Systemen gestartet:

FirstSpirit-Master-Server: Der FirstSpirit-Master-Server verwaltet zentral alle FirstSpirit-Projekte und übernimmt die Abarbeitung der Anfragen/Änderungen der Benutzer und verteilt die Aufgaben, soweit möglich, auf andere FirstSpirit-Server.

FirstSpirit-Generation-Server:

Bei komplexen bzw. häufigen Veröffentlichungsvorgängen sollten mehrere



FirstSpirit-Generierungsserver zum Einsatz kommen, um die Belastung bei der Erzeugung der Web-Präsenzen vom Master zu verlagern. Bei Bedarf können so auch mehrere Veröffentlichungen auf verschiedene Server verteilt werden. Ein Generierungsserver enthält einen RORS.

ReadOnly-Repository-Server (RORS): Ein spezieller Repository-Manager bearbeitet die Anfragen von einem Generierungs-Server.

Die einzelnen FirstSpirit-Server entscheiden beim Start, für welchen Aufgabenbereich sie verantwortlich sind und welche Manager dazu gestartet werden müssen. Ein FirstSpirit-Generierungsserver entsteht beispielsweise aus einem "normalen" FirstSpirit-Server, der nur die zur Kommunikation benötigten Dienste und den Generation-Manager startet. Die FirstSpirit Softwarearchitektur ist so aufgebaut, dass (vereinfacht dargestellt) bei der Ausführung eines Auftrags ein Generation-Manager verwendet wird, dem man nicht "ansieht", ob er lokal oder remote läuft.

7.6.2 Lizenzdatei prüfen

Die Funktionalität "Clustering" ist eine lizenzabhängige Funktionalität. Der Menüpunkt "Clustering" wird in der Anwendung zur FirstSpirit Server- und Projektkonfiguration nur angezeigt, wenn eine gültige Lizenz für diese Funktionalität besteht.

Über die Menüs "FirstSpirit – Konfiguration – Lizenz" des FirstSpirit Server-Monitorings können die gültigen FirstSpirit-Funktionen der Lizenzdatei `fs-license.conf` angezeigt werden (siehe Kapitel 8.6.1.2 Seite 451). Der Parameter `license.CLUSTERING` muss für die Verwendung der Funktionalität auf den Wert `1` gesetzt sein (siehe Abbildung 7-155).

Ist das nicht der Fall, kann eine neue gültige Lizenz angefordert und im blauen Fensterbereich eingefügt werden. Mit einem Klick auf den Button "Speichern" kann die neue Lizenzdatei gespeichert werden.



Manipulationen an der `fs-license.conf` führen zu einer ungültigen Lizenz. Sollten Änderungen notwendig werden, wenden Sie sich bitte an den Hersteller.





Beim Einfügen einer neuen Konfigurationsdatei `fs-license.conf` mit geänderter Clustering-Einstellung ist ein Neustart des Servers erforderlich. Der Cluster-Manager wird dabei automatisch gestartet.

Lizenz

```
license.ID=2226
#FIRSTspirit license
#Mon Jan 02 10:34:22 CET 2012
license.USER=e-spirit
license.EXPDATE=15.12.2012
license.MAXPROJECTS=0
license.MAXSESSIONS=0
license.MAXUSER=0
license.SOCKET_PORT=0
license.VERSION=5
license.MODULES=personalisation,search,integration,newsletter
license.WEBEDIT=1
license.WORKFLOW=1
license.REMOTEPROJECT=1
license.PACKAGEPOOL=1
license.DOCUMENTGROUP=1
license.ACCESS_API=1
license.APPTAB_SLOTS=20
license.ARCHIVE=0
license.CLUSTERING=1
license.ENTERPRISE_BACKUP=1
license.HIGHAVAILIBILITY=1
license.OFFICE_IMPORT=1
license.OFFICE_INTEGRATION=1
license.SCOPE=CORPORATE
license.TYPE=PRODUCTION
```



Speichern

Abbrechen

Abbildung 7-155: Anzeige der Parameter der Lizenzdatei (Server-Monitoring)

7.6.3 Konfiguration der Clusterknoten

Die Konfiguration der Clusterknoten erfolgt über die FirstSpirit Server- und Projektkonfiguration im Bereich "Sereigenschaften" im Menü "Clustering" (siehe



Kapitel 7.3.14 Seite 275).

7.6.4 Konfiguration des Generierungsauftrags

Die Erstellung und Konfiguration eines Generierungsauftrags erfolgt über die Anwendung zur Server- und Projektkonfiguration. Im Bereich "Projekteigenschaften" unter dem Menüpunkt "Auftragsplanung" werden alle Aufträge angezeigt, die für das Projekt angelegt wurden. Hier können auch neue Aufträge für die Generierung (oder Teil-Generierung) eines Projekts angelegt oder bereits bestehende Aufträge bearbeitet werden (Weiterführende Informationen zum Erstellen eines Generierungsauftrags siehe Kapitel 7.5.2 und Kapitel 7.5.10.)

Der Generierungsauftrag enthält die Aktion "generate" (im Register Aktionen). Diese Aktion dient dazu eine Voll- bzw. Teilgenerierung des gewählten Projekts durchzuführen. Eine Vollgenerierung generiert alle Inhalte des Projekts, eine Teilgenerierung nur die ausgewählten "Startpunkte" und deren Kinder.

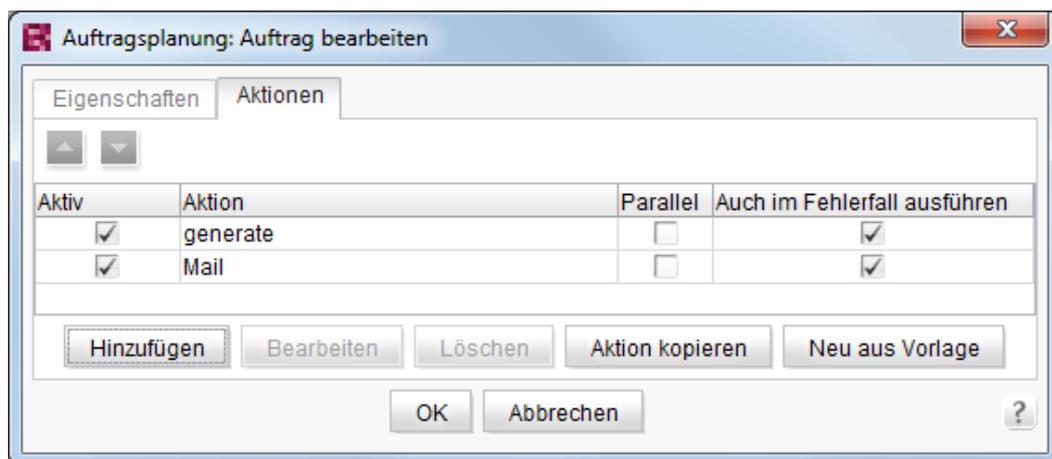


Abbildung 7-156: Projektbezogene Aktion "generate" eines Generierungsauftrags

Die Konfigurationseinstellung für die Aktion können mit einem Klick auf den Button "Bearbeiten" aufgerufen werden. Es öffnet sich der folgende Dialog:



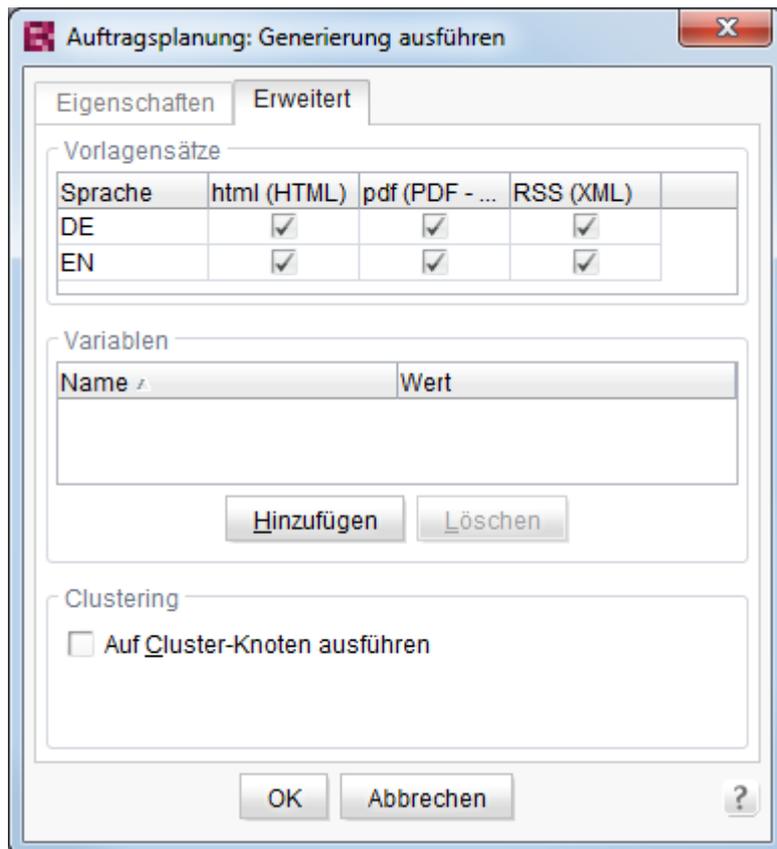


Abbildung 7-157: Erweiterte Einstellungen zur Aktion "generate"

Neben den bekannten Einstellungen zur Generierung (siehe Kapitel 7.5.10.2) enthält das Register "Erweitert" den Bereich "Clustering". Hier kann die Checkbox "Auf Cluster-Knoten ausführen" aktiviert werden. Mit Aktivierung dieser Checkbox wird die Generierung der Inhalte (Vollgenerierung bzw. Teilgenerierung) auf die vorhandenen Clusterknoten (bzw. Generierungs-Server) verteilt. Dabei wird immer der Clusterknoten verwendet, der zu diesem Zeitpunkt die geringste Auslastung aufweist.

Eine Übersicht der bestehenden Clusterknoten kann über das FirstSpirit-Server-Monitoring aufgerufen werden (siehe Kapitel 8.6.6 Seite 475).



7.7 Konfiguration der Rechtschreibprüfung

Das Modul "SpellService" dient der Rechtschreibprüfung im FirstSpirit-JavaClient und im WebClient.

Die Rechtschreibprüfung ist in den folgenden Eingabekomponenten verfügbar:

- CMS_INPUT_DOM
- CMS_INPUT_DOMTABLE

Das Modul "SpellService" besteht aus:

- einer Projekt-lokale Komponente: Diese Komponente kann nach der Installation des Moduls auf dem Server den gewünschten Projekten über deren Projekteigenschaften hinzugefügt werden (siehe Kapitel 7.4.17 Seite 184). Anschließend besteht die Möglichkeit, diese Komponente global (siehe 7.7.3 Seite 411) und projektspezifisch (siehe Kapitel 7.7.7 Seite 418) zu konfigurieren.
- einem Dienst: Der Dienst ist eine Serverkomponente, die über eine öffentliche Schnittstelle aus Eingabekomponenten (oder Skripten) angesprochen werden kann.

7.7.1 SpellService installieren / deinstallieren (Servereigenschaften)

7.7.1.1 Installieren

Das Modul "SpellService" wird als fsm-Datei zur Verfügung gestellt und muss zunächst über die Server- und Projektkonfiguration auf dem Server installiert werden.

Ist das Modul bereits auf dem Server installiert und soll lediglich durch eine neuere Version ersetzt werden, kann eine Aktualisierung erfolgen (siehe Kapitel 7.7.2). Eine Installation ist in diesem Fall nicht notwendig.

Die Installation erfolgt über die Servereigenschaften im Bereich "Module" (siehe Kapitel 7.3.11 Seite 254).

Mit einem Klick auf den Button "Installieren" öffnet sich ein Dialog zur Auswahl der fsm-Datei aus dem lokalen Dateisystem. Da das Modul "SpellService" einen Dienst beinhaltet, öffnet sich zunächst der Dialog "Dienst starten":



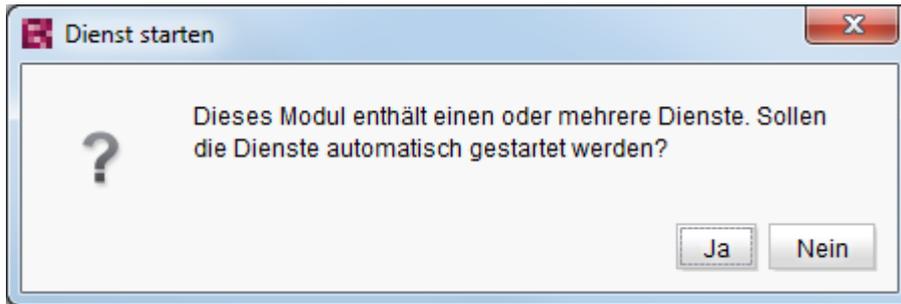


Abbildung 7-158: Dienste automatisch starten

Wird der Dialog mit "Ja" bestätigt, wird der Dienst "SpellService" automatisch gestartet. Der Dienst kann aber auch zu einem späteren Zeitpunkt entweder über die Server- und Projektkonfiguration oder über das Server-Monitoring gestartet werden (siehe Kapitel 7.7.5 Seite 416).

Die erfolgreiche Installation der fsm-Datei wird durch eine Meldung bestätigt. Das Modul "SpellService" wird anschließend im Bereich "Module" aufgelistet.

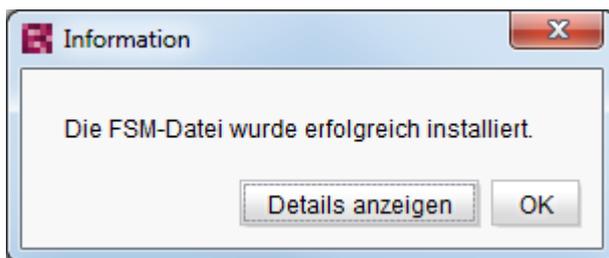


Abbildung 7-159: SpellService installiert

Im Anschluss an die Installation kann die Konfiguration der Projekt-lokalen Komponente erfolgen (siehe Kapitel 7.7.6 Seite 416).

7.7.1.2 Deinstallieren

Die Deinstallation des SpellService erfolgt über die Servereigenschaften im Bereich "Module" (siehe Kapitel 7.3.11 Seite 254).

Ist das Modul bereits auf dem Server installiert und soll lediglich durch eine neuere Version ersetzt werden, kann eine Aktualisierung erfolgen (siehe Kapitel 7.7.2). Eine Deinstallation ist in diesem Fall nicht notwendig.

Wird das Modul noch in Projekten verwendet, erscheint die folgende Fehlermeldung:



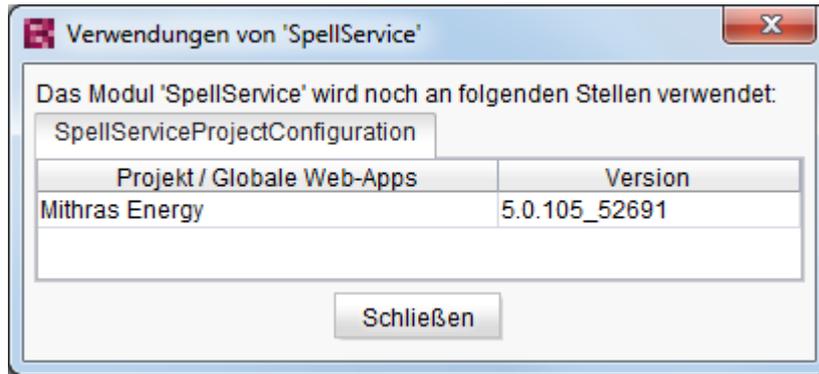


Abbildung 7-160: SpellService deinstallieren – Fehlermeldung

In diesem Fall müssen zunächst die Projekt-Komponenten innerhalb der Projekteigenschaften gelöscht werden (siehe Kapitel 7.7.6 Seite 416). Erst wenn keine Verwendungen mehr vorliegen, kann das Modul erfolgreich deinstalliert werden.

Mit einem Klick auf den Button "Deinstallieren" wird das Modul deinstalliert.

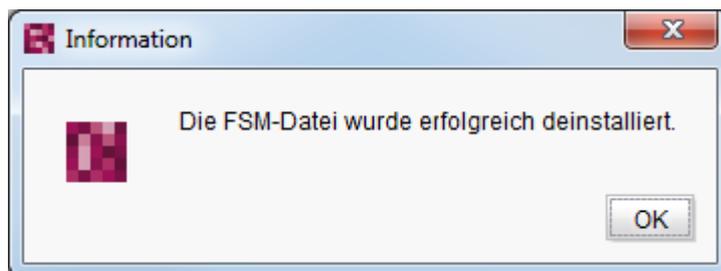


Abbildung 7-161: SpellService deinstallieren

7.7.2 SpellService aktualisieren (Servereigenschaften)

Eine Aktualisierung des Moduls kann immer dann erfolgen, wenn eine neuere Version der fsm-Datei vorliegt. (Ist die Datei älter oder identisch mit der installierten Version erscheint eine Fehlermeldung.)

Die Aktualisierung des Moduls erfolgt analog zur Installation und kann daher ebenfalls über den Button "Installieren" ausgeführt werden. Die alte Version wird dabei durch die neue Version des Moduls ersetzt. Eine zuverige Deinstallation ist nicht notwendig.

Nach erfolgreicher Aktualisierung kann die Version innerhalb der Projekteigenschaften ebenfalls aktualisiert werden.



Siehe auch Kapitel 7.3.11 Seite 254.

7.7.3 Globalen SpellService konfigurieren

Eine globale Konfiguration des Dienstes ist über die Servereigenschaften möglich. In diesem Bereich können Wörterbücher zum SpellService hinzugefügt werden. Wörterbücher besitzen einen eindeutigen Namen und verfügen über unterschiedliche Sprachinhalte.

Mit einem Klick auf den Button "Konfigurieren" öffnet sich der Dialog "Einstellungen". Standardmäßig wird das Wörterbuch "default" angezeigt. Das Wörterbuch wird serverweit für alle Projekte verwendet, in denen der SpellService konfiguriert wird, und enthält die Sprachen deutsch, englisch, französisch und spanisch.



Abbildung 7-162: Globale Konfigurationseinstellungen "SpellService"

Über den Dialog kann die globale Konfiguration des "SpellService" geändert werden. Dabei werden für jedes Wörterbuch folgende Informationen verwaltet:

Name: serverweit eindeutiger Name des Wörterbuchs.

Verwenden: Wird diese Option aktiviert, kann das Wörterbuch aus den Projekten heraus zentral verwendet werden. Wird die Verwendung eines Wörterbuchs deaktiviert, wird das Recht "Änderbar" ebenfalls automatisch deaktiviert.





Globale Wörterbücher werden in jedem neu angelegten Projekt verwendet, auch wenn die Spell-Service-Konfiguration im Projekt NICHT vorhanden ist. Soll die Verwendung des Wörterbuchs explizit unterbunden werden, muss die Projektkonfiguration "Spell-Service-Konfiguration" zunächst hinzugefügt werden und dann die ausgewählten globalen Wörterbücher oder der gesamte SpellService explizit deaktiviert werden.

Änderbar: Wird diese Option aktiviert, kann das Wörterbuch aus einem Projekt heraus zentral verändert werden. Das bedeutet, es können neue Einträge zum Wörterbuch hinzugefügt werden. Global definierte Schreib-/Leserechte können über die Projektkonfiguration geändert werden (in nur Leserechte). Der umgekehrte Fall ist natürlich nicht möglich. Die Checkbox "Änderbar" kann nur editiert werden, wenn "Verwenden" aktiviert ist.

Mit einem Klick auf den Button "Hinzufügen" können weitere globale Wörterbücher konfiguriert werden. Dabei muss zunächst ein eindeutiger Name für das neue Wörterbuch vergeben werden. Das globale Wörterbuch wird unter diesem Namen serverweit für alle Projekte verwendet, in denen der SpellService konfiguriert wird. Die global definierte Konfiguration (verwenden / änderbar) kann aber in den einzelnen Projekt-Konfigurationen angepasst werden.

Global hinzugefügte Wörterbücher werden im Server-Verzeichnis `conf\modules\SpellService.SpellService` abgelegt.

Mit einem Klick auf den Button "Löschen" kann ein zuvor hinzugefügtes Wörterbuch wieder entfernt werden. Vor dem Löschen erfolgt eine Sicherheitsabfrage. Das Wörterbuch "default" kann nicht gelöscht werden, es ist aber möglich, die Verwendung zu deaktivieren.

Mit einem Klick auf den Button "Bearbeiten" kann die Konfiguration für das Wörterbuch bearbeitet werden. Es öffnet sich der Dialog zur Konfiguration (siehe Kapitel 7.7.4 Seite 413).

Die Änderungen werden mit einem Klick auf den Button "OK" gespeichert.

Die Änderungen werden mit einem Klick auf den Button "Abbrechen" nicht gespeichert, das Fenster wird geschlossen.



7.7.4 Globale Wörterbücher konfigurieren

Mit einem Klick auf den Button "Bearbeiten" öffnet sich der Dialog zur Konfiguration des Wörterbuchs:

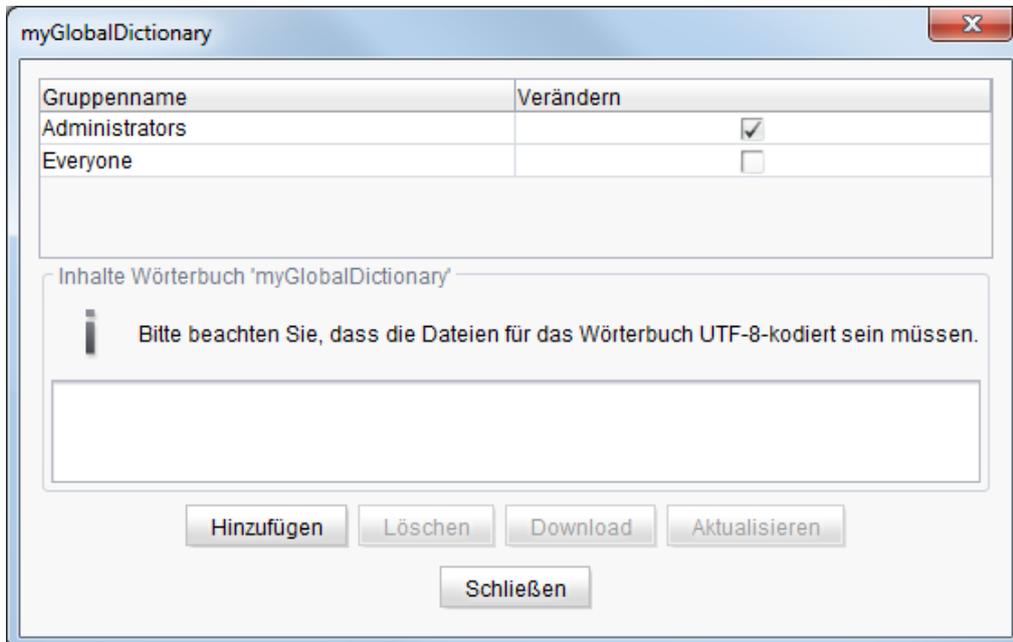


Abbildung 7-163: Wörterbuch konfigurieren

Der Dialog ist unterteilt in einen Bereich zur Rechtedefinition (siehe Kapitel 7.7.4.1 Seite 413) und einen Bereich zum optionalen Hinzufügen von sprachabhängigen Inhalten zum Wörterbuch (Dict-Dateien) (siehe Kapitel 7.7.4.2 Seite 414).

7.7.4.1 Berechtigungen für globale Wörterbücher konfigurieren

Gruppenname	Verändern
Administrators	<input checked="" type="checkbox"/>
Everyone	<input type="checkbox"/>

Abbildung 7-164: Berechtigungen konfigurieren

Im oberen Bereich des Konfigurationsdialogs (vgl. Abbildung 7-163) können die globalen Schreibrechte für die Gruppen "Administrators" und "Everyone" aktiviert oder deaktiviert werden. Innerhalb der Projektkonfiguration werden in diesem Bereich weitere Projektgruppen angezeigt (siehe Kapitel 7.7.8 Seite 420).

Die hier definierten globalen Rechte beeinflussen die Schreib-Leserechte innerhalb der Projektkonfiguration, können jedoch dort geändert werden (siehe Kapitel 7.7.8.1



Seite 420).



Exportieren von Projekten: Die globalen Wörterbücher werden zentral vom SpellService verwaltet. Das bedeutet, die globalen Wörterbücher werden beim Exportieren eines Projekts NICHT mit exportiert. Der Projektexport berücksichtigt nur die Projekt-lokalen Wörterbücher.

7.7.4.2 Sprachabhängige Inhalte zu globalen Wörterbüchern hinzufügen



Abbildung 7-165: Inhalte hinzufügen

Im unteren Bereich des Konfigurationsdialogs (vgl. Abbildung 7-163) können sprachabhängige Inhalte zum Wörterbuch hinzugefügt werden. Das Hinzufügen dieser Dateien ist optional. Entweder liegen bereits Wortlisten für die Sprachen (Inhaltssprachen) vor, die in diesem Bereich hinzugefügt werden können oder es werden keine Dateien angegeben. In diesem Fall werden automatisch beim ersten Hinzufügen eines unbekanntes Wortes im Java- oder WebClient neue Listen angelegt, die anschließend auch im Konfigurationsdialog angezeigt werden.

Fügt ein Redakteur beispielsweise für die Sprache (Inhaltssprache) englisch ein unbekanntes Wort zum Wörterbuch hinzu, wird automatisch die entsprechende Dict-Datei auf dem Server angelegt.

Mit einem Klick auf den Button "Hinzufügen" kann eine neue Wörterliste (Dict-Datei) zum Wörterbuch hinzugefügt werden. Es öffnet sich Dialog zur Auswahl der Dict-Datei aus dem lokalen Dateisystem. (Der Button ist nur aktiv, solange noch nicht für alle Sprachen eine Zuordnung erfolgt ist). Nach der Auswahl muss die Zuordnung der Datei zur gewünschten Inhaltssprache (in den Projekten) definiert werden:



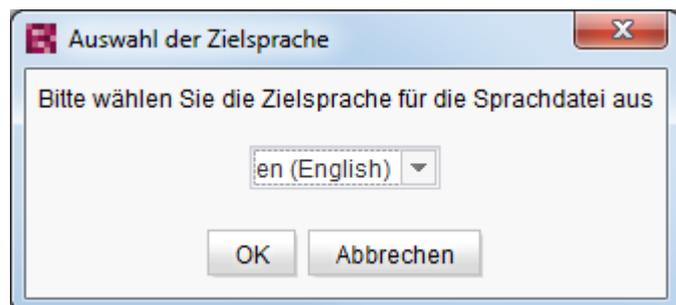


Abbildung 7-166: Zielsprache auswählen

Die Klappliste zeigt dabei alle Sprachen an, die im Bereich "Sprach-Vorlagen" innerhalb der Server-Eigenschaften definiert wurden (siehe Kapitel 7.3.6 Seite 246). Jede Dict-Datei wird also einer bestimmten Serversprache zugeordnet. Sprachen, für die bereits eine Zuordnung vorliegt, werden in der Liste grau dargestellt.

Nach der Zuordnung zu einer Sprache wird die Wortliste mit dem folgenden Namen im Dialog angezeigt: `Name_des_Wörterbuchs.Sprachkürzel.dict`



Anstelle manuell verfasster Wörterlisten lassen sich auch fremde, frei verfügbare OpenSource-Wörterbücher hinzufügen. Wichtig ist, dass diese die FirstSpirit-Anforderung (1 Wort pro Zeile) erfüllen.

Ein Beispiel hierfür sind Aspell-Wörterbücher, die über die offizielle Aspell-Seite (aspell.net) heruntergeladen werden können. Das erhaltene Archiv muss entpackt und die enthaltene Wörterbuch-Datei mit der Endung ".dict" versehen werden, bevor diese über den beschriebenen Weg dem FirstSpirit-Wörterbuch hinzugefügt werden kann.

Mit einem Klick auf den Button "Löschen" kann eine zuvor hinzugefügte Wörterliste (Dict-Datei) wieder entfernt werden. Vor dem Löschen erfolgt eine Sicherheitsabfrage.

Mit einem Klick auf den Button "Download" kann eine bestehende Dict-Datei exportiert werden. Es öffnet sich ein Dialog zur Auswahl eines Download-Verzeichnisses aus dem lokalen Dateisystem. Die Dict-Datei kann unter dem bestehenden oder einem neuen Namen in das ausgewählte Verzeichnis exportiert werden. Über die Funktion "Aktualisieren" kann die Dict-Datei lokal gespeichert, bearbeitet und anschließend über den Button "Aktualisieren" wieder zur Projektkonfiguration hinzugefügt werden.



Mit einem Klick auf den Button "Aktualisieren" kann eine bereits hinzugefügte Dict-Datei durch eine neue Datei ersetzt werden. Anders als beim "Hinzufügen" einer Dict-Datei zu einem Wörterbuch, ist hier keine Zuordnung der Datei zur gewünschten Inhaltssprache erforderlich. Es wird die bestehende Sprachzuordnung und der eindeutige Name der ursprünglichen Datei übernommen und nur die zugrundeliegende Datei ausgetauscht (beispielsweise die Datei, die mittels "Download" zum Editieren in ein lokales Verzeichnis geschrieben wurde).

7.7.5 Dienst "SpellService" starten und konfigurieren

Es gibt unterschiedliche Möglichkeiten den Dienst "SpellService" zu steuern und zu konfigurieren:

- Über die Server- und Projektkonfiguration innerhalb der Servereigenschaften im Bereich "Module" (siehe Kapitel 7.3.11 Seite 254)
- Über das Server-Monitoring im Bereich FirstSpirit – Konfiguration - Dienste (siehe Kapitel 8.6.1.7 Seite 455) und im Bereich FirstSpirit – Konfiguration - Steuerung (siehe Kapitel 8.6.2.4 Seite 463).

7.7.6 SpellService als Projekt-Komponente hinzufügen

Die Rechtschreibprüfung kann projektspezifisch aktiviert bzw. deaktiviert werden. Dabei erfolgt die Installation des Spell-Service-Moduls zunächst zentral über die Servereigenschaften (siehe Kapitel 7.7.1 Seite 408). Die globale Konfiguration wird ebenfalls über die Servereigenschaften ausgeführt (siehe Kapitel 7.7.3 Seite 411). Dabei können globale Wörterbücher angelegt und konfiguriert werden, die anschließend innerhalb der Spell-Service-Projektkonfiguration verfügbar sind (siehe Kapitel 7.7.4 Seite 413).

Die Projektspezifische Konfiguration des SpellServices erfolgt anschließend über die Projekteigenschaften im Bereich "Projekt-Komponenten" (siehe Kapitel 7.4.17 Seite 325).



Bevor eine Spell-Service-Projektkonfiguration erfolgen kann, muss der SpellService gestartet sein (siehe Kapitel 7.7.5 Seite 416).



Mit einem Klick auf den Button "Hinzufügen" kann die Spell-Service-Konfiguration einem Projekt hinzugefügt werden.

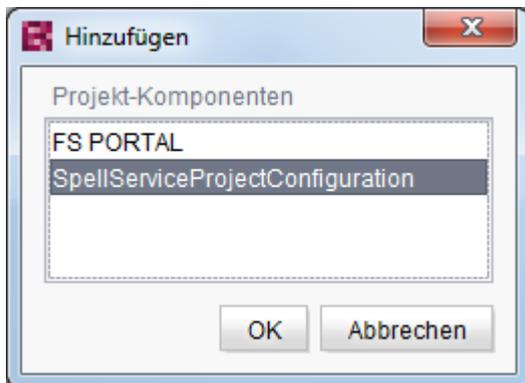


Abbildung 7-167: Servereigenschaften

Danach stehen die global konfigurierten Funktionalitäten der installierten Komponente im Projekt zur Verfügung.

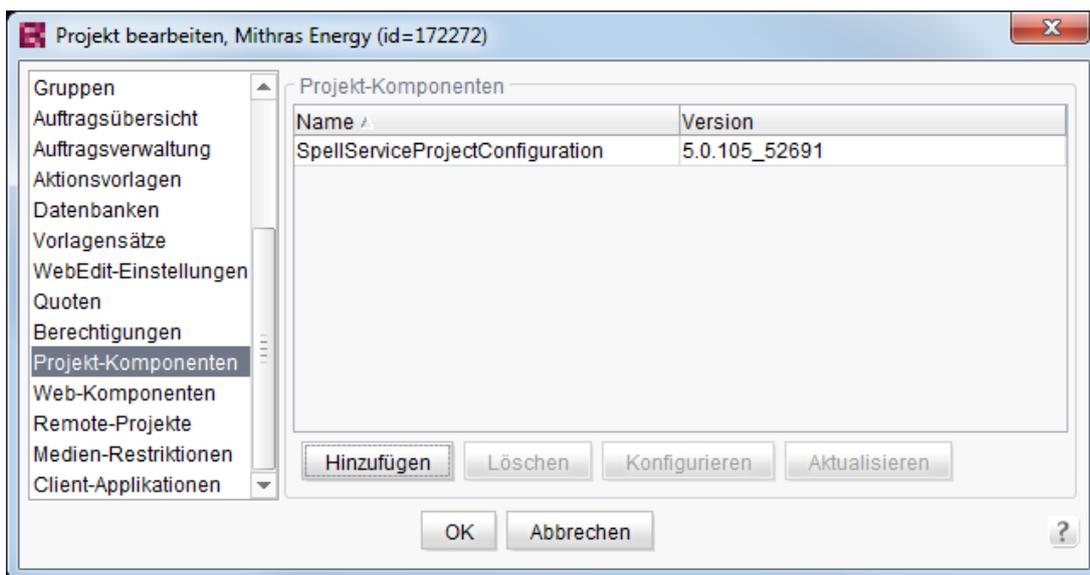


Abbildung 7-168: Projekteigenschaften – Spell-Service-Projektkonfiguration

Mit einem Klick auf den Button "Löschen" kann eine zuvor hinzugefügte Spell-Service-Projektkonfiguration wieder entfernt werden. Dabei werden alle projektspezifischen Spell-Service-Konfigurationen (projektspezifische Wörterbücher, Berechtigungen) ebenfalls entfernt. Vor dem Löschen erfolgt eine Sicherheitsabfrage.

Mit einem Klick auf den Button "Aktualisieren" kann eine bestehende Spell-Service-Konfiguration aktualisiert werden. Der Button ist nur aktiv, wenn auf dem Server eine



Aktualisierung des Moduls "SpellService" ausgeführt wurde (siehe Kapitel 7.7.2 Seite 410). In diesem Fall unterscheiden sich die Versionsnummern der globalen Spell-Service-Projektconfiguration und der projektspezifischen Spell-Service-Projektconfiguration. Die bisherigen Konfigurationseinstellungen (Wörterbücher, Berechtigungen) bleiben bei der Aktualisierung erhalten. (Abhängig von der neuen Modulversion können aber manuelle Anpassungen erforderlich sein.)

Mit einem Klick auf den Button "Konfigurieren" kann die projektspezifische Spell-Service-Konfiguration definiert werden (siehe Kapitel 7.7.7 Seite 418).

7.7.7 Projektspezifische Spell-Service-Konfiguration

Initial werden im projektspezifischen Konfigurationsdialog nur die global definierten Wörterbücher angezeigt (in der Übersicht grau dargestellt). Abhängig von der Konfiguration können diese globalen Wörterbücher im Projekt direkt verwendet und (eventuell) geändert werden. Innerhalb der Konfiguration können aber auch weitere projektspezifische Wörterbücher hinzugefügt und editiert werden.



Abbildung 7-169: Konfiguration der Spell-Service-Projektconfiguration

Über den Dialog kann die projektspezifische Konfiguration des "SpellService" geändert werden. Dabei werden für jedes Wörterbuch folgende Informationen verwaltet:

Name: projektweit eindeutiger Name des Wörterbuchs (für globale Wörterbücher ist der Name serverweit eindeutig).

Verwenden: Wird diese Option aktiviert, kann das Wörterbuch innerhalb des Projekts verwendet werden. Wird die Verwendung eines Wörterbuchs deaktiviert,



wird das Recht "Änderbar" ebenfalls automatisch deaktiviert.



Globale Wörterbücher werden in jedem neu angelegten Projekt verwendet, auch wenn die Spell-Service-Konfiguration im Projekt NICHT vorhanden ist. Soll die Verwendung des Wörterbuchs explizit unterbunden werden, muss die Projektkonfiguration "Spell-Service-Konfiguration" zunächst hinzugefügt werden und dann die ausgewählten globalen Wörterbücher explizit deaktiviert werden.

Änderbar: Ist diese Option aktiviert, kann das Wörterbuch aus dem Projekt heraus verändert werden. Das bedeutet, es können neue Einträge zum Wörterbuch hinzugefügt werden. Global definierte Schreib-/Leserechte können über die Projektkonfiguration geändert werden (nur Leserechte). Der umgekehrte Fall ist natürlich nicht möglich. Die Checkbox "Änderbar" kann nur editiert werden, wenn "Verwenden" aktiviert ist.

Wird die Checkbox "SpellService aktivieren" deaktiviert, wird die Rechtschreibprüfung für das Projekt deaktiviert. Es werden dann weder die Inhalte der globalen noch der projektspezifischen Wörterbücher ausgewertet. Standardmäßig ist die Rechtschreibhilfe für alle Projekte aktiviert.

Mit einem Klick auf den Button "Hinzufügen" können weitere projektspezifische Wörterbücher hinzugefügt werden (siehe Kapitel 7.7.8 Seite 420). Dabei muss zunächst ein projektweit, eindeutiger Name für das neue Wörterbuch vergeben werden. Die Inhalte projektspezifisch hinzugefügter Wörterbücher werden im Server-Verzeichnis `\data\projects\project_projectID\modules\SpellService.SpellServiceProjectConfiguration` angelegt.

Mit einem Klick auf den Button "Löschen" kann ein zuvor hinzugefügtes projektspezifisches Wörterbuch entfernt werden. Vor dem Löschen erfolgt eine Sicherheitsabfrage. Globale Wörterbücher können über die Projektkonfiguration nicht gelöscht werden, die Verwendung innerhalb des Projekts kann aber deaktiviert werden.

Mit einem Klick auf den Button "Bearbeiten" kann die Konfiguration für das Wörterbuch bearbeitet werden. Es öffnet sich der Dialog zur Konfiguration des Wörterbuchs (siehe Kapitel 7.7.8 Seite 420).

Die Änderungen werden mit einem Klick auf den Button "OK" gespeichert.



Die Änderungen werden mit einem Klick auf den Button "Abbrechen" nicht gespeichert, das Fenster wird geschlossen.



Importieren von Projekten: Beim Import eines Projektes mit einer Spell-Service-Konfiguration wird geprüft, ob der SpellService installiert und gestartet ist. Ist das nicht der Fall, erhält der Administrator eine Warnung, die lokale Konfiguration des SpellService wird aber angelegt. Wird der SpellService nachträglich aktiviert, steht die Konfiguration und die lokalen Wörterbücher sofort zur Verfügung.

7.7.8 Projektspezifische Wörterbücher hinzufügen

Die Konfiguration von projektspezifischen Wörterbüchern erfolgt analog zur Konfiguration globaler Wörterbücher (siehe Kapitel 7.7.4 Seite 413).

Mit einem Klick auf den Button "Bearbeiten" öffnet sich der Dialog zur Konfiguration des Wörterbuchs (siehe Abbildung 7-163).

Der Dialog ist unterteilt in einen Bereich zur Rechtedefinition (siehe Kapitel 7.7.8.1 Seite 420) und einen Bereich zum optionalen Hinzufügen von sprachabhängigen Inhalten zum Wörterbuch (Dict-Dateien) (siehe Kapitel 7.7.4.2 Seite 414).

7.7.8.1 Berechtigungen für projektspezifische Wörterbücher konfigurieren

Gruppenname	Verändern
Developer	<input type="checkbox"/>
Editor	<input type="checkbox"/>
Everyone	<input type="checkbox"/>
Administrators	<input checked="" type="checkbox"/>

Abbildung 7-170: Berechtigungen konfigurieren

Im oberen Bereich des Konfigurationsdialogs (vgl. Abbildung 7-163) können die projektspezifischen Schreibrechte für eine Projekt-Gruppe aktiviert oder deaktiviert werden.

Global gesetzte Rechte werden als Standardkonfiguration in die Projektkonfiguration übernommen, können dort aber geändert werden.

Im Unterschied zur globalen Konfiguration (siehe Kapitel 7.7.4.1 Seite 413) stehen hier nicht nur die Standardgruppen "Everyone" und "Administrators" zur Verfügung, sondern alle Gruppen, die Zugriff auf das Projekt besitzen (siehe Kapitel 7.4.8 Seite



303).

7.7.8.2 Sprachabhängige Inhalte zu globalen Wörterbüchern hinzufügen

Im unteren Bereich des Konfigurationsdialogs (vgl. Abbildung 7-163) können sprachabhängige Inhalte zum Wörterbuch hinzugefügt werden. Das Hinzufügen dieser Dateien ist optional. Entweder liegen bereits Wortlisten für die Sprachen (Inhaltssprachen) vor oder es können neue leere Listen angelegt werden.

Das Hinzufügen sprachabhängiger Inhalte zu projektspezifischen Wörterbüchern erfolgt analog zum Hinzufügen von Inhalten zu globalen Wörterbüchern (siehe Kapitel 7.7.4.2 Seite 414).

Der Dialog zur Zuordnung einer Dict-Datei zu einer Projektsprache (Inhaltssprache) bietet nur die im Projekt bekannten Sprachen an (siehe 7.4.5 Seiten 295).

7.8 Unterstützung für Apache FOP

Die Unterstützung für Apache FOP in FirstSpirit wird über ein Modul realisiert. Das Modul kann, wie gewohnt, über die Server- und Projektkonfiguration installiert werden (siehe Kapitel 7.3.11 Seite 254):

- **fs-fop.fsm:** Das Modul "fs-fop.fsm" befindet sich zum Zeitpunkt der Freigabe von FirstSpirit Version 5.0 auf dem Stand 1.00 von Apache FOP³⁴. Sollte es zu Fehlermeldungen wie

```
Font "Symbol,normal,700" not found. Substituting with "Symbol,normal,400".
```

kommen, können diese umgangen werden, indem der Parameter `font-family` in `fo:root` mit der Standardschrift des Dokuments ergänzt wird, z. B.:

```
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format" font-family="Helvetica">
```

Nach der Installation des Moduls steht der betreffende Präsentationskanal innerhalb der Servereigenschaften zur Verfügung (siehe auch Kapitel 7.3.2 Seite 235):

³⁴ Weiterführende Informationen siehe <http://xmlgraphics.apache.org/fop/compliance.html>



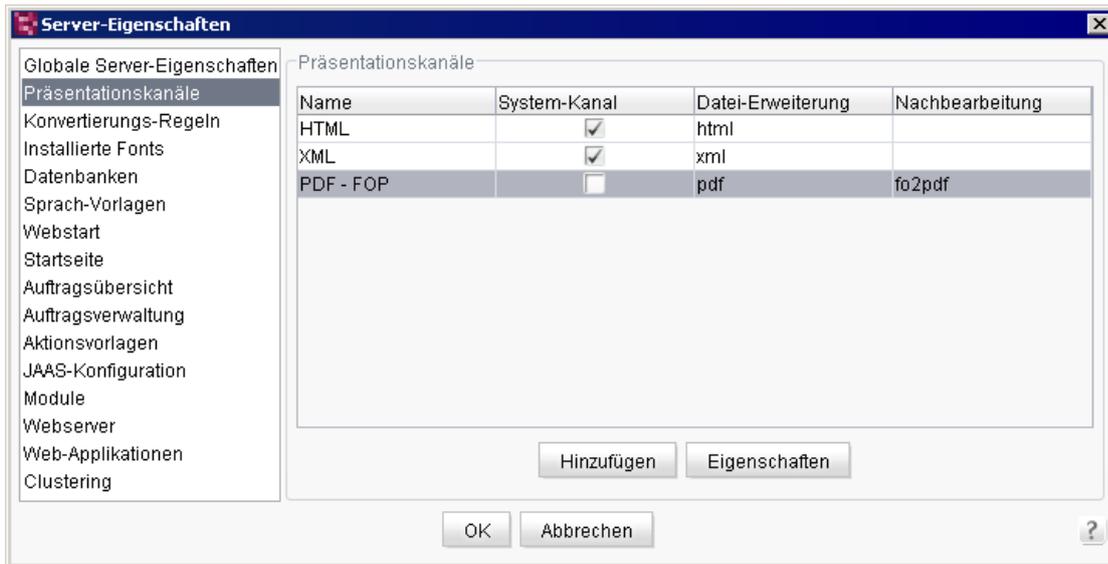


Abbildung 7-171: Präsentationskanal (Servereigenschaften) nach der FOP-Installation

und kann anschließend in den Projekten als neuer Vorlagensatz hinzugefügt werden (siehe Kapitel 7.4.13 Seite 313):

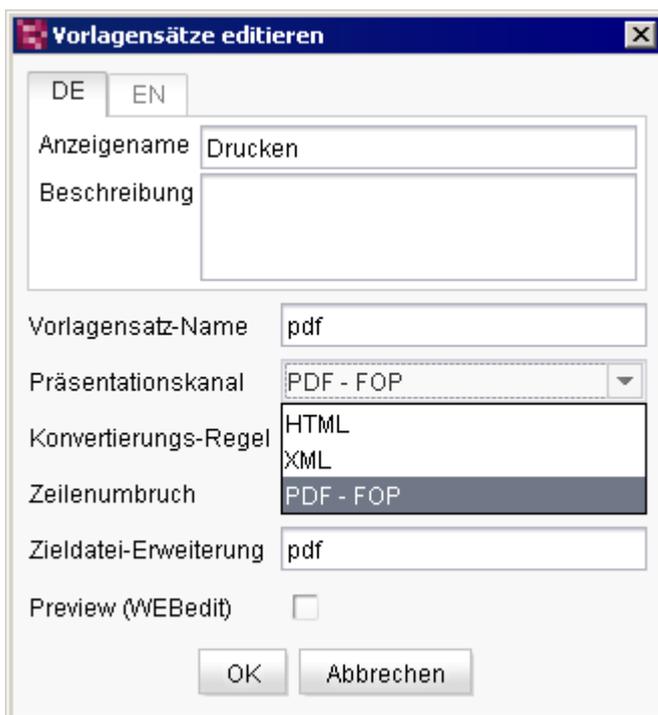


Abbildung 7-172: Vorlagensätze (Projekteigenschaften) nach der FOP-Installation



7.9 Projektarchivierung

In diesem Kapitel soll das FirstSpirit-Konzept zu Versionierung und Archivierung näher erläutert werden, um so einen besseren Einblick in die Vorgehensweise des Archivierungsauftrags (siehe auch Kapitel 7.5.10.1 Seite 380) zu erhalten.



Zur Verwendung der Archiv-Funktion im Vergleich zur Verwendung des Moduls "FirstSpirit EnterpriseBackup" siehe auch *FirstSpirit Release Notes* 4.2, Kap. "Langzeit-Archivierung und Backup in FirstSpirit".

7.9.1 Versionshistorie

Das FirstSpirit Versionierungs- und Historisierungs-Konzept fordert, dass möglichst alle Änderungen an Objekten im JavaClient lückenlos nachvollziehbar sind und so jederzeit ein Zugriff auf frühere Systemzustände möglich ist. Bei jeder Änderung an einem Objekt wird eine neue Version des Objektes angelegt. Damit verfügt ein Objekt über eine Versionsgeschichte (auch: **Versionshistorie**), die darüber Aufschluss gibt, welche Änderungen von welchem Benutzer zu welchem Zeitpunkt vorgenommen wurden. Da die einzelnen Objekte in einem Projekt wiederum mit anderen Objekten verbunden sind (z. B. bestehen Seiten aus einzelnen Absätzen und werden in der Struktur-Verwaltung zu einer Navigation verwoben), fließen Änderungen an diesen verbundenen Objekten ebenso in die Versionshistorie mit ein. Erst dann können Änderungen vollständig nachvollzogen werden. Über die Funktion "Versionshistorie" im Kontextmenü eines Objektes (siehe dazu auch *FirstSpirit Handbuch für Redakteure*) können die Einträge der für dieses Objekt relevanten Versionshistorie eingeblendet werden.

7.9.2 Revisionen

Zusätzlich zur Versionshistorie von Objekten werden in FirstSpirit auch Änderungen, die bei der Bearbeitung von Repositories (siehe Kapitel 7.5.10.1 Seite 380) in einem logischen Zusammenhang stehen, festgehalten. Dieser Zustand des Gesamtsystems zu einem Zeitpunkt wird als Revision bezeichnet. Revisionen werden fortlaufend durchnummeriert:

- alle in einer Revision neu angelegten Objekte erhalten die gleiche neue Revisionsnummer,
- veränderte Objekte werden im Repository nicht überschrieben, sondern als jeweils neues Objekt (mit einer höheren Revisionsnummer) eingefügt,



- alle nicht veränderten Objekte behalten ihre alten Revisionsnummern bei.

Beispiel:

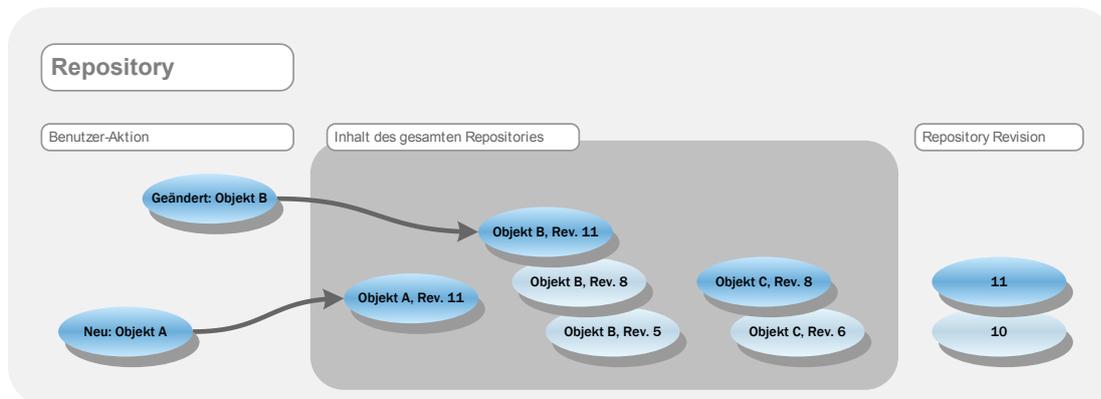


Abbildung 7-173: Beispiel Repository-Revisionen

Ausgangszustand: Die aktuelle Repository-Revision ist **10**.

Benutzer-Aktion: Der Benutzer legt das **Objekt A** an (z. B. einen neuen Absatz auf einer Seite) und verändert das **Objekt B** (z. B. einen bereits bestehenden Absatz). **Objekt C** (z. B. eine andere Seite) bleibt unverändert.

Ergebnis: Wird diese Aktion an das Repository übergeben (z. B. durch das Beenden des Bearbeitungsvorgangs auf der Seite mittels "Speichern" oder mithilfe der Tastenkombination <Strg> +E), wird eine neue Repository-Revision mit der Nummer 11 angelegt. Beide Objekte A und B erhalten ebenfalls die Revisionsnummer 11. Da Objekt A neu angelegt wurde, liegt dieses Objekt in nur genau einer Revision vor (11). Da Objekt B verändert wurde und der alte Stand nicht überschrieben werden darf, liegt Objekt B nun in mehreren Revisionen vor (in diesem Beispiel 5, 8 und 11). Alle unveränderten Objekte behalten ihre letzten Revisionen bei, so dass beispielsweise Objekt C die Revision 8 (die letzte Revision, in der dieses Objekt verändert wurde) beibehält.

7.9.3 Mindestanforderung an die Projektarchivierung

Objekte bestehen aus verschiedenen Dateien. Wird ein Objekt im JavaClient oder WebClient geändert, wird auch mindestens eine der Dateien verändert. Einzelne Dateien eines Objekts können in verschiedenen Revisionen verändert vorliegen, somit sind dann auch diese unterschiedlichen Revisionen Teil des Objektes.

Die Archivierung wird auf Basis dieser Dateien durchgeführt, d.h. je nach Konfiguration eines Archivierungsauftrags durch den Projektadministrator (siehe Kapitel 7.5.10.1 Seite 380) werden nicht mehr benötigte Dateien bei einer



Archivierung aus dem Repository in die Archiv-Datei verschoben. Die Folge ist, dass Objekte nach einer Archivierung potenziell nicht mehr in allen Revisionen vollständig vorliegen. Auf diese Weise entstehen sogenannte "teilarchivierte" Revisionen.

Wird ein **Archivierungszeitraum** angegeben (partielle Archivierung, siehe Abbildung 7-145 Seite 382), bleiben alle Objekte **innerhalb dieses Zeitraums** inklusive aller Einträge in der Versionshistorie erhalten. Alle Revisionen innerhalb dieses Zeitraums können uneingeschränkt wiederhergestellt werden, ebenso gelöschte Objekte.

Alle Dateien, die sich in den Revisionen **vor dem Archivierungszeitraum** geändert haben, sowie alle Dateien aller Revisionen im Falle einer vollständigen Archivierung, werden daraufhin betrachtet, ob sie noch aktuell im Projekt benötigt werden. Dabei wird als **Archivierungsgrenze** der Zeitpunkt angenommen, bis zu dem alle Dateien vollständig erhalten bleiben. Unterhalb dieser Archivierungsgrenze bleiben mindestens folgende Revisionen einer Datei im Projekt-Repository erhalten:

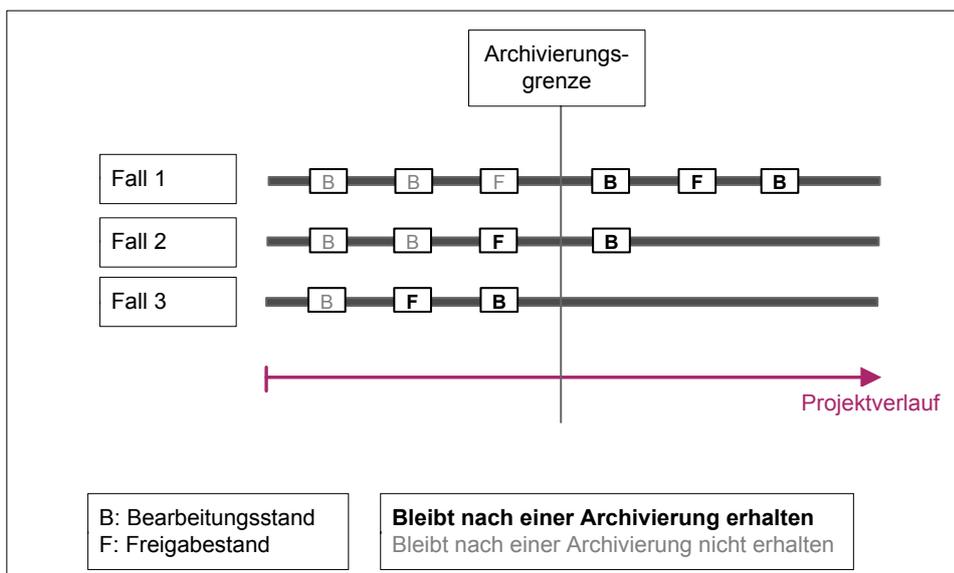


Abbildung 7-174: Mindestanforderung an die Projektarchivierung

- **Fall 1:** Liegen Bearbeitungs- und Freigabestand der Datei oberhalb der Archivierungsgrenze, bleibt in jedem Fall die Revision der Datei des aktuellen Bearbeitungsstands, des letzten Freigabestands und des davorliegenden Bearbeitungsstands erhalten.
- **Fall 2:** Ist oberhalb der Archivierungsgrenze kein Freigabestand der Datei vorhanden, bleibt zusätzlich zum letzten Bearbeitungsstand in jedem Fall auch der unterhalb der Archivierungsgrenze zuletzt freigegebene Stand der Datei erhalten.



- **Fall 3:** Ist oberhalb der Archivierungsgrenze weder Freigabe- noch Bearbeitungsstand der Datei vorhanden, bleiben der letzte Bearbeitungsstand und der letzte Freigabestand der Datei unterhalb der Archivierungsgrenze erhalten.

7.9.4 Versionshistorie nach Archivierung

Werden bei einer Archivierung Daten (z. B. gelöschte Objekte, Änderungsinformationen) aus dem Projekt ausgelagert, ist die jeweilige Revision, in der das Objekt vorhanden war, nicht mehr vollständig.

Revision	Datum	Änderung an	Redakteur	Kommentar	Attribute	Kindliste	Inhalt	Metadaten
20520	06.03.2013 09:41:37	Mithras Homepage	gutknecht (Barbara Gutknecht)	Zwischenspeichern	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
20519	06.03.2013 09:41:26	Mithras Homepage	gutknecht (Barbara Gutknecht)	Zwischenspeichern	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
20139	21.06.2012 11:31:06	Mithras Homepage	Admin (Admin)	Arbeitsablauf Freigabe An...	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
20005	31.05.2012 14:24:17	Mithras Homepage	Admin (Admin)	Speichern	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
16541	02.09.2010 13:32:18	Mithras Homepage	editor (editor)	Speichern	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
16516	23.06.2010 13:04:05	Mithras Homepage	Admin	Serverseitige Freigabe	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Auswahl

1. Revision:

2. Revision (Strg):

Optionen

- Änderungen an **K**indelementen einblenden
- Änderungen an **M**edien einblenden
- Änderungen an **V**orlagen einblenden
- Versteckte **R**evisionen einblenden
- Teilarchivierte Revisionen einblenden

Abbildung 7-175: Versionshistorie nach Archivierung

In der Versionshistorie eines Objektes wird dann nur noch die älteste **vollständig vorhandene** Revision (also die erste Revision oberhalb der Archivierungsgrenze) angezeigt sowie alle jüngeren (also alle weiteren Revisionen oberhalb der Archivierungsgrenze, 1. in Abbildung 7-175). Diese werden in der Versionshistorie mit schwarzer Schrift dargestellt. Sie können angezeigt (Schaltfläche "Anzeigen"), für Vergleiche mit anderen noch vollständig erhaltenen Revisionen herangezogen (Schaltfläche "Vergleichen") und wiederhergestellt (Schaltfläche "Wiederherstellen") werden.

Sind keine vollständigen Revisionen vorhanden (z. B. wenn im Auftrag die Option "Versionshistorie nicht erhalten (vollständige Archivierung)" ausgewählt wurde, siehe Abbildung 7-144), werden in jedem Fall der **letzte freigegebene Stand** (2. in Abbildung 7-175, in diesem Beispiel liegt er unterhalb der Archivierungsgrenze) und, falls vorhanden, der **jüngste Bearbeitungsstand** angezeigt (siehe dazu auch Kapitel 7.9.3 Seite 424, Abbildung 7-174). Sie können für Vergleiche mit



vollständig vorhandenen Revisionen herangezogen (Schaltfläche "Vergleichen") und wiederhergestellt werden (Schaltfläche "Wiederherstellen", siehe dazu auch Kapitel 7.9.4.1 Seite 427). Ggf. ist eine korrekte Anzeige (Schaltfläche "Anzeigen") nicht mehr möglich, da für die Anzeige erforderliche Revisionen von zugrundeliegenden Vorlagen nach einer Archivierung möglicherweise nicht mehr vorhanden sind. Es kommt zu der in Abbildung 7-177 gezeigten Meldung.

Revisionen, die **nicht mehr vollständig** vorhanden sind, können über die Option "Teilarchivierte Revisionen einblenden" angezeigt werden (3. in Abbildung 7-175). Sie können lediglich angezeigt werden (Schaltfläche "Anzeigen"), ggf. aber historisch nicht korrekt, vgl. Meldung in Abbildung 7-177.

7.9.4.1 Wiederherstellen nach Archivierung (Funktion "Wiederherstellen")

Liegt die wiederherzustellende Revision im archivierten Zeitraum, ist die Option "Spezifische Wiederherstellung – Fehlende abhängige Objekte ignorieren" voreingestellt und kann nicht abgewählt werden. Auf diese Weise werden fehlende Referenzen zum ausgewählten Objekt bei der Wiederherstellung ignoriert:

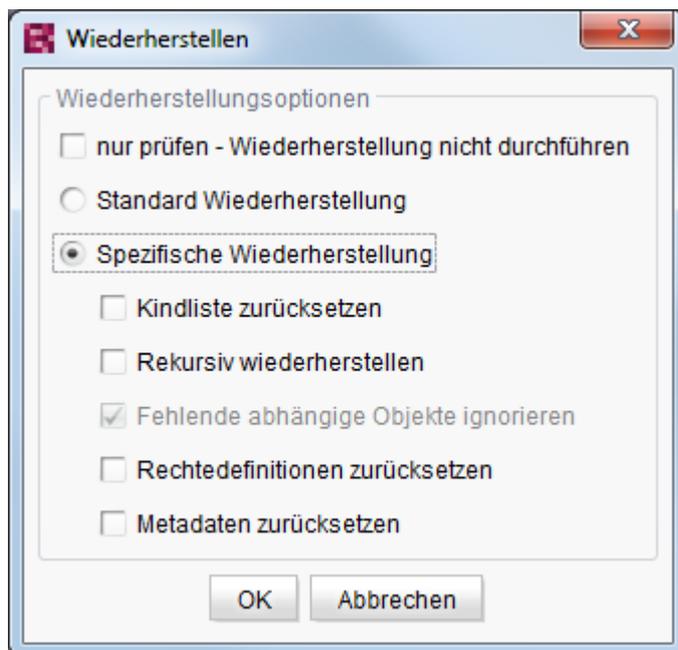


Abbildung 7-176: Wiederherstellung nach Archivierung



7.9.4.2 Vorschau von archivierten Revisionen (Funktion "Anzeigen")

Sind für die Vorschau eines Objekts Daten von verbundenen Objekten (z. B. Vater-Objekten) erforderlich, die jedoch archiviert wurden, wird immer die letzte vollständige Version für eine Vorschau genutzt. In diesem Fall wird folgende Meldung angezeigt:

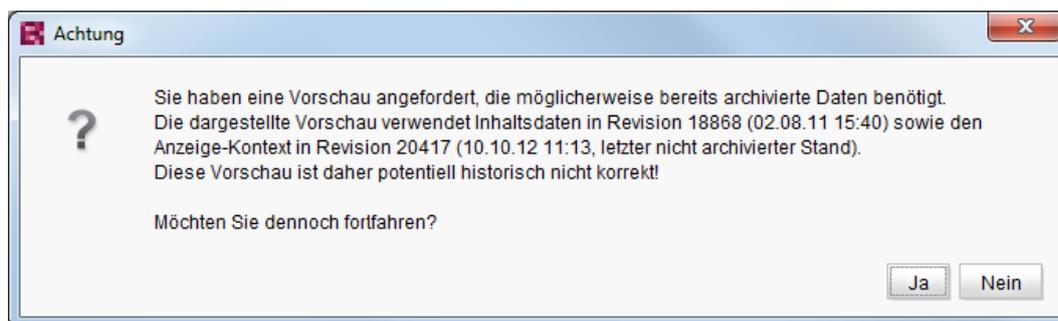


Abbildung 7-177: Meldung historisch nicht korrekte Vorschau

7.9.4.3 Vergleich von Revisionen nach Archivierung (Funktion "Vergleichen")

Stehen nach einer Archivierung nicht mehr alle Teile der Objekte zur Verfügung, die für einen Vergleich notwendig sind, wird folgende Meldung angezeigt:



Abbildung 7-178: Meldung Versionsvergleich nach Archivierung



8 FirstSpirit Server-Monitoring

Das browserbasierte FirstSpirit Server-Monitoring ist eine Webapplikation, die zur Überwachung des FirstSpirit-Servers dient. Mit ihrer Hilfe können aktuelle Betriebsparameter, wie Speicherauslastung, Benutzeranzahl u.v.a., angezeigt werden. Das FirstSpirit Server-Monitoring wird analog zum WebClient über einen Webbrowser bedient.

The screenshot displays the 'Übersicht Status' (Overview Status) page of the FirstSpirit Server-Monitoring application. At the top, there is a language selection dropdown ('Sprachumschaltung') and a 'Mailversand der angezeigten Informationen' (Email delivery of displayed information) option. The main content area is divided into several sections:

- Navigation:** A vertical sidebar on the left contains menu items such as 'Übersicht', 'Status', 'Aktivitäten', 'Sitzungen', 'Projekte', 'Logdateien', 'Benutzer', and 'FirstSpirit'.
- Status:** A central section titled 'Status' provides a summary of system and load information.

Grundinformationen (System)		Grundinformationen (Load)	
Hostname des Servers:	AIRAMI	Aktive Projekte:	0
Server-Version:	5.0_BETA.34.52009	Aktive Benutzer:	1
Server-Verzeichnis:	D:\FirstSpirit_Server\FirstSpirit_5	Aktive Sitzungen:	1
IP-Adresse:		Vorschauberechnungen:	0
Ports:	(HTTP/SOCKET)	Aktuell generierte Projekte:	0
Applikations-Server:	intern	Laufende Aufträge:	0
Lizenznehmer:	e-spirit	Auslastung der CPU:	1 %
Lizenzablauf:	15.12.2012	Anzahl laufender Threads:	56
Server Uptime:	1h 13m 27s 463ms	Freier Festplattenspeicher:	8,455 GB
Java-Version:	1.6.0_27	Aktueller Speicherverbrauch:	5 %
Java-Verzeichnis:	C:\Program Files (x86)\Java\jdk1.6.0_27\re	Frei:	1,246 GB
Anzahl Projekte:	53	Benutzt:	71,947 MB
Anzahl Benutzer:	960	Maximal:	1,317 GB
- Speicherauslastung der letzten 24h:** A bar chart showing memory usage over a 24-hour period, with a peak around 1.400M.
- Sessions der letzten 24h:** A bar chart showing the number of active sessions over a 24-hour period, with a peak around 50.

Abbildung 8-1: FirstSpirit Server-Monitoring – Übersicht Status

Auf der linken Bildschirmseite befindet sich die Navigation des FirstSpirit Server-Monitorings. Einige Menüebenen verfügen über weitere Untermenüs, die dann in einer weiteren Navigation im oberen Bereich der Seite angezeigt werden.

Aktuelle Server-Status-Informationen können auch per (Text-)Mail versandt werden.





Die gleichzeitige Anmeldung über einen Webbrowser (z. B. in mehreren Fenstern oder Tabs) auf mehreren FirstSpirit-Servern mit demselben Hostnamen (z. B. myServer:8200 und myServer:8400) wird nicht unterstützt.



Dokumentiert wird hier der vollständige Funktionsumfang für Serveradministratoren. Projektadministratoren haben ebenfalls Zugriff auf das FirstSpirit Server-Monitoring, ihnen stehen aber nicht alle Menüebenen zur Verfügung.

8.1 Übersicht

8.1.1 Übersicht – Status

Auf dieser Seite wird eine Übersicht über die wichtigsten Informationen dargestellt (siehe Abbildung 8-1). Die Seite ist in vier Bereiche unterteilt:

Grundinformationen (System): In diesem Bereich werden allgemeine Informationen zur Konfiguration des Servers dargestellt.

Grundinformationen (Load): In diesem Bereich werden allgemeine Informationen zur aktuellen Auslastung des Servers dargestellt.

Speicherauslastung der letzten 24h: Dieser Bereich gibt einen Überblick über die Speicherauslastung des Servers in den letzten 24 Stunden. Die schwarze Linie zeigt den definierten -Xmx-Wert aus der Konfigurationsdatei `fs-wrapper.conf` (siehe Kapitel 4.3.2.3, Parameter `wrapper.java.maxmemory`) an. Hell-brombeer ist der Bereich, den die Java VM im Betriebssystem für den Java-Heap reserviert hat und dunkel-brombeer der tatsächlich verbrauchte Heap. Durch einen Klick auf die Grafik gelangt man zum Bereich "FirstSpirit – Monitoring – Speicher" (siehe Kapitel 8.6.5.1 ab Seite 467).

Sessions der letzten 24h: Dieser Bereich gibt einen Überblick über die Anzahl der Sitzungen in den letzten 24 Stunden. Durch einen Klick auf die Grafik gelangt man zum Kapitel "FirstSpirit – Monitoring – Sitzungen" (siehe Kapitel 8.6.5.2 ab Seite 468).



8.1.2 Übersicht – Aktivitäten

Auf dieser Seite gibt es eine Übersicht über einige Aktivitäten auf dem Server:

Aktivitäten

Laufende Aufträge:

Mithras 12	Rebuild search index	Ausführung stoppen
	Name: Rebuild search index (?) Status: Läuft	Startzeitpunkt: 16.07.2012 11:51:23 Bisherige Laufzeit: 28s 81ms

Geplante Aufträge

Projekt-bezogen Server-weit

Dokumentation WebClient (PRODUKTIV) - Generate PDF Document:

Startzeitpunkt: 16.07.2012 18:00:00

- Generate
- Mail

Online-Dokumentation FirstSpirit 5.0 (PRODUKTIV) - generate (daily):

Startzeitpunkt: 16.07.2012 18:00:00

- Generate
- Mail

Dokumentation WebClient (PRODUKTIV) - generate (daily):

Startzeitpunkt: 16.07.2012 18:00:00

- Generate
- Mail

Zeitplanung der letzten 24 Stunden (maximal 15 Einträge)

Dokumentation WebClient (PRODUKTIV)

Name	Status	Beginn	Ende
Generate PDF Document	Generate [Erfolgreich] Mail [Erfolgreich]	16.07.2012 01:54:39	16.07.2012 01:55:57
Generate WebEdit Tooltips	Generate [Fehler] Mail [Erfolgreich]	16.07.2012 01:54:39	16.07.2012 01:55:13
generate full	Generate [Erfolgreich] Mail [Erfolgreich]	16.07.2012 01:54:39	16.07.2012 01:55:45

Online-Dokumentation FirstSpirit 5.0 (PRODUKTIV)

Name	Status	Beginn	Ende
generate full	Generate [Erfolgreich] Mail [Erfolgreich]	16.07.2012 01:54:39	16.07.2012 02:01:18
build dvd-index	Generate [Erfolgreich] Mail [Erfolgreich]	16.07.2012 01:54:39	16.07.2012 01:55:14

Aktive Projekte

Projekt	Sitzungen	Benutzer	Referenzberechnung	Suchindizierung
<u>Dokumentation WebClient (PRODUKTIV)</u>	1	1	Nicht gestartet	Nicht gestartet
<u>Online-Dokumentation FirstSpirit...</u>	1	1	Nicht gestartet	Nicht gestartet

Abbildung 8-2: FirstSpirit Server-Monitoring – Überblick – Aktivitäten

Laufende Aufträge: In dieser Tabelle werden alle Aufträge aufgelistet, die aktuell auf dem Server aktiv sind (siehe Kapitel 7.4.21 Seite 341).

Geplante Aufträge: In dieser Tabelle werden alle zukünftig anstehenden, aktiven (nicht manuelle) Aufträge aufgelistet (siehe Kapitel 7.5.1.2 Seite 352).



Zeitplanung der letzten 24 Stunden (maximal 15 Einträge): In dieser Tabelle werden alle ausgeführten Aufträge der letzten 24 Stunden aufgelistet.

Aktive Projekte: In dieser Tabelle werden alle Projekte aufgelistet, die aktuell auf dem Server aktiv sind. Dabei werden folgende Informationen angezeigt:

Während eines laufenden Auftrags werden weitere **auftragsspezifische Informationen** angezeigt. Bei einer Generierung wird beispielsweise der Startzeitpunkt, die bisherige Laufzeit des Auftrags und die geschätzte Restlaufzeit sowie die durchschnittliche Generierungsdauer pro Seite angezeigt. Außerdem zeigt die Übersicht, wie viele Seiten bisher bei der Generierung des Projektes erzeugt wurden.

Durch einen Klick auf ein aktives Projekt wird eine ausführliche Übersicht zu diesem Projekt angezeigt (siehe Kapitel 8.2.1 ab Seite 434).

Sitzungen: Anzahl der zurzeit aktiven Sitzungen im Projekt

Benutzer: Anzahl der zurzeit angemeldeten Benutzer im Projekt

Referenzberechnung: Status der Referenzberechnung im Projekt (siehe Kapitel 9.15).

Suchindizierung: Status der Indizierung für die Suche (siehe Kapitel 9.18).

8.1.3 Übersicht – Sitzungen

Auf dieser Seite befindet sich eine Tabelle, in der alle aktuell laufenden Sitzungen aufgelistet werden.

Session-ID: eindeutige ID der Sitzung

Typ: Art der Sitzung. Es wird zwischen folgenden Sitzungsarten unterschieden:

- WEB: Sitzung wird über die Startseite aufgebaut.
- WebEdit: Sitzung wird über den WebClient aufgebaut.
- Main: Sitzung wird über den JavaClient aufgebaut.
- Child: Untergeordnete, interne Sitzung, die beispielsweise beim Bearbeiten eines Projektes im JavaClient oder über die Server- und Projektkonfiguration aufgebaut wird. *Hinweis:* Beim Bearbeiten eines Projektes über die Server- und Projektkonfiguration wird eine CHILD-Session für jedes bearbeitete Projekt geöffnet, die erst nach dem Beenden der Java-Anwendung wieder geschlossen wird.



- **Webmonitor:** Sitzung wird über das Server-Monitoring aufgebaut.
- **Dummy:** Interne Sitzungen, die beispielsweise bei Ausführung einer Generierung aufgebaut werden. Diese Sessions werden nur angezeigt, wenn die Checkbox "**Alle Sessions anzeigen**" aktiviert wurde. Für die Auswertung der lizenzierten Sitzungen werden diese Sitzungen nicht mitgezählt (siehe Kapitel 4.3.5 Seite 94).
- **Remote:** Sitzung wird über ein Remote-Projekt aufgebaut.
- **Staging:** Sitzung wird bei der Erzeugung einer Vorschau des generierten Projektstands aufgebaut (über den bei der Generierung angegebenen URL). Diese Sessions werden für die Auswertung der lizenzierten Sitzungen nicht mitgezählt (siehe Kapitel 4.3.5 Seite 94).



Bei einer Sitzung vom Typ Child ist es möglich, dem Benutzer der Sitzung eine Nachricht zu schicken. Durch einen Klick auf die betreffende Session-ID gelangt man zu der Seite "FirstSpirit – Nachricht" (siehe Kapitel 8.6.3 ab Seite 466). In diesem Fall wird die Nachricht jedoch nicht an alle Benutzer auf dem Server verschickt, sondern ausschließlich an den Benutzer der ausgewählten Sitzung.

Benutzer: Login-Name des Benutzers. Durch einen Klick auf den Benutzer einer Sitzung wird man zu den Detailinformationen zu diesem Benutzer auf der Seite "Benutzer – Suche" verlinkt (Siehe Kapitel 8.5.1 ab Seite 449).

Projekt: Projekt, in dem der jeweilige Benutzer momentan arbeitet. Durch einen Klick auf das Projekt einer Sitzung wird man zu den Detailinformationen zu diesem Projekt auf der Seite "Projekte – Übersicht" geleitet (Siehe Kapitel 8.2.1.1 ab Seite 434).

Durch den Link "**Beenden**" besteht die Möglichkeit, eine Sitzung ohne Vorwarnung an den Benutzer zu beenden. Hierbei ist zu beachten, dass alle nicht gespeicherten Eingaben innerhalb der jeweiligen Sitzung verloren gehen.

Zugriff: zeigt den Zeitpunkt an, zu dem der jeweilige Benutzer das letzte Mal auf den Server zugegriffen hat

Login Zeit: zeigt den Zeitpunkt an, zu dem der jeweilige Benutzer sich auf dem Server angemeldet hat.

Diese und weitere Informationen können auch über das JMX-Monitoring angefordert werden (siehe Kapitel 9.22 Seite 505).



8.2 Projekte

8.2.1 Projekte – Übersicht

Auf dieser Seite befindet sich eine Tabelle mit allen auf dem Server installierten Projekten. Aufgelistet wird neben dem Projektnamen und der Projektbeschreibung noch die Anzahl der für das einzelne Projekt zugelassenen Benutzer und ob das Projekt auf dem Server aktiviert oder deaktiviert ist. Weiterhin bekommt man den momentanen Auftrags-Status des Projektes und die Projekt-ID angezeigt.

Diese und weitere Informationen können auch über das JMX-Monitoring angefordert werden (siehe Kapitel 9.14 Seite 496).

Diese Liste kann nach Projektnamen, Benutzer, Aktivierungsstatus oder der ID des Projekts sortiert werden. Über die Option "**Nur aktive Projekte**" können alle deaktivierten Projekte ausgeblendet werden. Durch einen Klick auf die Schaltfläche

Projekte aktualisieren

kann die Projektübersicht aktualisiert werden.



Für Projektadministratoren werden nur die Projekte aufgelistet, für die man als Administrator eingetragen ist.

Durch einen Klick auf ein aufgelistetes Projekt werden die Detailinformationen zu diesem Projekt angezeigt.

8.2.1.1 Detailinformationen zum Projekt

Zu den Detailinformationen gehören Angaben über:

Projektname, Beschreibung so wie sie in den Projekteigenschaften definiert sind.

Projekt-ID identifiziert das Projekt eindeutig für den Server.

Aktiviert gibt an, ob das Projekt aktiviert ist oder nicht.

Max. Sessions gibt die maximale Anzahl an Sitzungen an.

Letzter Zugriff, Letzte Änderung, Letzte Freigabe, Startzeit der Statistik gibt den Zeitpunkt an, an dem diese Aktionen durchgeführt wurden.



Anzahl Benutzer gibt die Anzahl der eingetragenen Benutzer für dieses Projekt an.

Benutzer gibt die Logins der eingetragenen Benutzer für dieses Projekt an.

Zuletzt benutzt gibt die letzten 5 Nutzer des Projektes an.

Außerdem gibt es noch eine Tabelle, in der alle für das Projekt definierten Gruppen und deren zugehörige Benutzer aufgelistet sind.

Import Log anzeigen Mit einem Klick auf die Schaltfläche kann die Log-Datei des Projektimports angezeigt werden.

Import ID Map anzeigen: Mit einem Klick auf die Schaltfläche kann die Import-ID-Map für das Projekt angezeigt werden. Beim Importieren eines Projekts werden die "alten" IDs durch neue Werte ersetzt. Die Ersetzungen von "alt" nach "neu" werden in der ID Map abgebildet.

8.2.2 Projekte – Statistik

Diese Seite ist noch einmal untergliedert in die Bereiche

- Zugriffe: Statistik über die Zugriffe (siehe Kapitel 8.2.2.1 ab Seite 435)
- Veröffentlichung: Statistik über die Veröffentlichungen (siehe Kapitel 8.2.2.2 ab Seite 436)
- Ressourcen: Statistik über die Ressourcen (siehe Kapitel 8.2.2.3 ab Seite 437)

8.2.2.1 Zugriffe

Auf dieser Seite befinden sich Informationen über die Zugriffe auf die installierten Projekte. In einer Tabelle werden alle auf dem Server installierten Projekte aufgelistet. Neben dem Projektnamen gibt es Angaben über den Zeitpunkt des letzten Zugriffs und der letzten Änderung des Projektes. Weiterhin wird angezeigt, welcher Benutzer zuletzt auf das Projekt zugegriffen hat, wie viele Benutzer für das Projekt zugelassen sind und wie viele Sitzungen maximal gleichzeitig aktiv waren.

Diese Liste kann nach Projektnamen, Benutzer, letzter Zugriff, Letzte Änderung oder den maximalen Sessions sortiert werden. Über die Option "**Nur aktive Projekte**" können alle deaktivierten Projekte ausgeblendet werden.





Für Projektadministratoren werden nur die Projekte aufgelistet, für die man als Administrator eingetragen ist.

Durch einen Klick auf ein aufgelistetes Projekt werden die Detailinformationen zu diesem Projekt angezeigt. (Siehe Kapitel 8.2.1.1 ab Seite 434)

8.2.2.2 Veröffentlichung

Auf dieser Seite befinden sich Informationen über die vorgenommenen Veröffentlichungen der Projekte. In der Tabelle werden alle auf dem Server installierten Projekte aufgelistet. Unterhalb der Projekte stehen die Aufträge, die zu den Projekten ausgeführt wurden. Neben dem Auftragsnamen gibt es Angaben über die Anzahl der bisherigen Ausführungen, sowie die Dauer der letzten Ausführung und die durchschnittliche Dauer einer Auftragsausführung.

Diese Liste kann nach allen Spalten der Tabelle sortiert werden. Über die Option **"Nur aktive Projekte"** können alle deaktivierten Projekte ausgeblendet werden.

Durch einen Klick auf einen aufgelisteten Auftrag, z.B. eine Veröffentlichung, werden die Detailinformationen zur Auftragsausführung in diesem Projekt angezeigt:

Veröffentlichung

Auftrag: generate (daily)

Projekt: Online-Dokumentation FirstSpirit 5.0 (PRODUKTIV)

ID	158611
Name	generate (daily)
Beschreibung	
Typ	täglich
Status	aktiv
Projektname	Online-Dokumentation FirstSpirit 5.0 (PRODUKTIV)

Anzahl Ausführungen	6
Letzte Dauer	4m 27s 280ms
Durchschnittliche Dauer	3m 0s 144ms
letzte Ausführung	16.07.2012 18:00:00 – 16.07.2012 18:04:27

Name	Projektname	Beginn	Status	Fehler	Benutzer	Dauer	
generate (daily)	Online-Dokumentation FirstSpirit 5.0 (PRODUKTIV)	16.07.2012 18:00:00	[Erfolgreich]	0F / 0E / 0W	SYSTEM	4m 27s 280ms	Historie anzeigen
» Generate		16.07.2012 18:00:00	[Erfolgreich]	0F / 0E / 0W		4m 27s 215ms	Historie anzeigen
» Mail		16.07.2012 18:04:27	[Erfolgreich]	0F / 0E / 0W		56ms	Historie anzeigen
generate (daily)	Online-Dokumentation FirstSpirit 5.0 (PRODUKTIV)	13.07.2012 18:00:00	[Erfolgreich]	0F / 0E / 0W	SYSTEM	2m 13s 942ms	Historie anzeigen
» Generate		13.07.2012 18:00:00	[Erfolgreich]	0F / 0E / 0W		2m 13s 882ms	Historie anzeigen
» Mail		13.07.2012 18:02:13	[Erfolgreich]	0F / 0E / 0W		55ms	Historie anzeigen

Abbildung 8-3: FirstSpirit Server-Monitoring - Veröffentlichungen zu einem Projekt



Diese Anzeige enthält weitere auftragsspezifische Informationen und zeigt beispielsweise an wieviele Veröffentlichungen bisher für dieses Projekt durchgeführt wurden und wie groß die durchschnittliche bzw. letzte Dauer der Generierungen war, so dass abgeschätzt werden kann, ob kürzlich vorgenommene Änderungen die Performance des Projekts beeinflusst haben.



Für Projektadministratoren werden nur die Projekte aufgelistet, für die man als Administrator eingetragen ist.

8.2.2.3 Ressourcen

Auf dieser Seite befinden sich Informationen über die Ressourcen, welche die Projekte benötigen. Hier wird der benötigte Festplattenspeicher zu jedem Projekt aufgeführt.

Über die Option **"Nur aktive Projekte"** können alle deaktivierten Projekte ausgeblendet werden.



Für Projektadministratoren werden nur die Projekte aufgelistet, für die man als Administrator eingetragen ist.

Durch einen Klick auf ein aufgelistetes Projekt werden die Detailinformationen zu diesem Projekt angezeigt. (Siehe Kapitel 8.2.1.1 ab Seite 434)



8.3 Log-Dateien

8.3.1 Log-Dateien – kompletter Server



Diese Menüebene steht nur Serveradministratoren zur Verfügung.

Diese Ansicht ist unterteilt in die Bereiche:

- Aktuelle Ansicht (siehe Kapitel 8.3.1.1 Seite 439)
- Historie (siehe Kapitel 8.3.1.2 Seite 440)
- Suche (siehe Kapitel 8.3.1.3 Seite 441)

✉ Über dieses Icon können Log-Dateien innerhalb der Anwendung zum Server-Monitoring heruntergeladen und/oder als Mailanhang verschickt werden. Beim Öffnen des Mailversands über das Icon werden die betreffenden Log-Dateien automatisch angehängt:

Mailversand

Empfänger



helpdesk@e-spirit.de ▼

Betreff

WEBMonitor: Protokoll-Datei

Kommentar

Anhang

fs-server INFO 20120717100542 20120717100542.log (2,177 KB)

Senden

Abbrechen

Abbildung 8-4: Mailversand mit angehängter Log-Datei



Sie können innerhalb des Dialogs aber auch mit einem Klick auf dem Link heruntergeladen und gespeichert werden.

8.3.1.1 Aktuelle Ansicht

Auf dieser Seite werden die aktuellsten Aktionen der Log-Datei ausgegeben.

The screenshot shows the 'aktuelle Ansicht' window with the following elements:

- Navigation: aktuelle Ansicht, Historie, Suche
- Level: Info (dropdown)
- Zeilen: 100 (dropdown)
- Buttons: Anzeigen, Suchen
- Log List Header: Jetzt, Pause, Details, Ältere
- Log Entries:
 - INFO [-] 16.07.2012 15:25:25 {pID=5674266} (de.espirit.firstspirit.store.access.sitestore.PageRefImpl): generating page reference 'produkteeinzelseite' (id=5674606, html, EN, 1986)
 - INFO [+] 15:25:25 {pID=5674266} (fs.generate.SiteProduction): creating output stream for /en [...]
 - INFO [+] 15:25:25 {pID=5674266} (fs.generate.SiteProduction): copy picture audio_video_conne [...]
 - INFO [+] 15:25:24 {pID=5674266} (fs.generate.SiteProduction): copy picture set_top_box (id=5 [...]
 - INFO [+] 15:25:24 {pID=5674266} (fs.generate.SiteProduction): copy picture set_top_box (id=5 [...]
 - INFO [+] 15:25:19 {pID=5674266} (fs.generate.SiteProduction): copy picture connector_cable ([...]
 - INFO [+] 15:25:19 {pID=5674266} (fs.generate.SiteProduction): copy picture audio_video_conne [...]
 - INFO [+] 15:25:19 {pID=5674266} (fs.generate.SiteProduction): copy picture audio_video_conne [...]
 - INFO [+] 15:25:19 {pID=5674266} (fs.generate.SiteProduction): copy picture audio_video_conne [...]
 - ERROR [+] 15:25:16 {pID=5674266} (fs.generate.SiteProduction): cannot create access editor fo [...]
 - ERROR [+] 15:25:16 {pID=5674266} (fs.generate.SiteProduction): cannot create access editor fo [...]
 - INFO [+] 15:25:16 {pID=5674266} (fs.generate.SiteProduction): copy file content product_flea [...]

Abbildung 8-5: Log-Dateien – Aktuelle Ansicht

In der Titelzeile befindet sich die Angabe über den Zeitraum, in dem die Serveraktionen stattgefunden haben.

"**Jetzt**" bedeutet, dass die Liste ständig aktualisiert wird und die aktuellsten Aktionen in der Liste ganz oben stehen.

Durch einen Klick auf die Schaltfläche **Pause** kann die Aktualisierung der Liste angehalten werden, um sich einige Einträge in Ruhe ansehen zu können. Durch einen Klick auf das Plus [+] hinter jeder Aktion, wird der Eintrag in seiner vollständigen Länge angezeigt.

Durch einen Klick auf die Schaltfläche **Details** können alle Einträge gleichzeitig in ihrer vollständigen Länge angezeigt werden.

Durch einen Klick auf die Schaltfläche **Ältere** ist es möglich, sich einen älteren Teil der Log-Datei ausgeben zu lassen. (siehe hierzu Kapitel 8.3.1.2 ab Seite 440)



Oberhalb der Log-Liste gibt es einige Auswahlmöglichkeiten, um die aktuelle Anzeige einzuschränken:

Level: In diesem Feld kann ausgewählt werden, bis zu welchem Informationslevel die Serveraktionen angezeigt werden sollen. Zur Auswahl stehen die Level Debug, Info, Warnung und Fehler.

Zeilen: In diesem Feld kann ausgewählt werden, wie viele Zeilen aus dem Server-Log gleichzeitig dargestellt werden sollen.

Durch einen Klick auf die Schaltfläche **Anzeigen** wird die Ansicht aktualisiert.

Suchen Über die Suchfunktion können bestimmten Textfragmenten in den Log-Daten gesucht werden. Groß- und Kleinschreibung muss hierbei beachtet werden. Die Suchergebnisse werden unterhalb der aktuellen Logliste angezeigt.



Abbildung 8-6: Log-Datei – Aktuelle Ansicht – Suchergebnis

8.3.1.2 Historie

Auf dieser Seite ist es möglich, sich einen älteren Teil der Log-Datei ausgeben zu lassen. Die Anzeige und Einschränkungsmöglichkeiten (Level, Zeilen) entsprechen weitgehend der "Aktuellen Ansicht" aus Kapitel 8.3.1.1.

Ergänzt werden die Eingabemöglichkeiten hier durch eine Datums- und eine Zeitauswahl:

Datum: Durch einen Klick auf die Schaltfläche  öffnet sich ein Fenster für Auswahl eines Datums. Mithilfe der Pfeiltasten neben dem Monatsnamen kann jeweils ein Monat vor bzw. zurück geblättert werden. Durch einen Klick auf den gewünschten Tag schließt sich das Fenster und das ausgewählte Datum wird in das Feld übernommen.

Zeit: Durch einen Klick auf die Schaltfläche  öffnet sich ein Fenster für Auswahl einer Uhrzeit (mit einem erneuten Klick wird das Fenster wieder geschlossen). Durch einen Klick auf die gewünschte Uhrzeit schließt sich das Fenster und die



ausgewählte Zeit wird in das Feld übernommen.

Durch einen Klick auf die Schaltfläche **Anzeigen** wird die Ansicht aktualisiert.

In der Titelzeile befindet sich die Angabe über den Zeitraum, in dem die Serveraktionen stattgefunden haben. Durch einen Klick auf das Plus [+] hinter jeder Aktion, wird der Eintrag in seiner vollständigen Länge angezeigt.

Durch einen Klick auf die Schaltfläche **Details** können alle Einträge gleichzeitig in ihrer vollständigen Länge angezeigt werden.

Suchen Über die Suchfunktion können bestimmte Textfragmente in den Log-Daten gesucht werden. Groß- und Kleinschreibung muss hierbei beachtet werden. Die Suchergebnisse werden unterhalb der aktuellen Logliste angezeigt.

8.3.1.3 Suche

Auf dieser Seite ist es möglich, sich einen bestimmten Zeitraum der Log-Datei ausgeben zu lassen. Die Ausgabe kann durch Suchkriterien eingeschränkt werden.

The screenshot displays the 'Suche' (Search) section of the FirstSpirit administration interface. At the top, there are navigation tabs for 'aktuelle Ansicht', 'Historie', and 'Suche'. Below the tabs, the search criteria are defined: 'Zeitraum' (Time Range) from '16.07.2012 11:33:50' to '17.07.2012 10:05:42'. The search options are set to 'Suchwort' (Search Term) with the value 'generate', and 'Exceptions' are unselected. The search level is set to 'Info' and 'Treffer pro Seite' (Hits per page) is set to '10'. A 'Suchen' button is visible.

The search results are displayed in a table with columns for 'Logauriss' (Log Summary), 'Zeilen vorher' (Previous Lines), and 'Zeilen nachher' (Next Lines). The results show several log entries for the search term 'generate' on July 17, 2012, at 10:05:42. The entries include information about the session manager, scheduler, and site production tasks.

Below the search results, there is a section for 'Logauriss' (Log Summary) with a 'Historie anzeigen' (Show History) button. This section displays a list of log entries with their timestamps and descriptions, such as 'deleted file (timeout exceeded)', 'Session terminated', and 'finished schedule'.



Abbildung 8-7: Log-Dateien– Suche

Zeitraum: Über die beiden Datums- und Zeitauswahlfelder kann der Zeitraum festgelegt werden, für den die Aktionen der Log-Datei ausgegeben werden sollen. Durch einen Klick auf die Schaltfläche  öffnet sich ein Fenster für Auswahl eines Datums und durch einen Klick auf die Schaltfläche  öffnet sich ein Fenster für Auswahl einer Uhrzeit. Standardmäßig ist immer ein Suchintervall von einem Tag eingestellt.

Suchoptionen: Ist die Option **Suchwort** aktiviert, dann kann nach bestimmten Textfragmenten in den Log-Daten gesucht werden. Ist die Option **Exceptions** aktiviert, dann werden alle Aktionen im angegebenen Zeitraum ausgegeben.

Level: In diesem Feld kann ausgewählt werden, bis zu welchem Informationslevel die Serveraktionen angezeigt werden sollen. Zur Auswahl stehen die Level Debug, Info, Warnung und Fehler.

Treffer pro Seite: In diesem Feld kann ausgewählt werden, wie viele Zeilen aus dem Server-Log gleichzeitig dargestellt werden sollen.

Durch einen Klick auf die Schaltfläche  wird die Ansicht aktualisiert.

In der Titelzeile befindet sich die Angabe über den Zeitraum, in dem die Serveraktionen stattgefunden haben. Durch einen Klick auf das Plus [+] hinter jeder Aktion, wird der Eintrag in seiner vollständigen Länge angezeigt.

Durch einen Klick auf die Schaltfläche  können alle Einträge gleichzeitig in ihrer vollständigen Länge angezeigt werden.

Beim Klick auf einen Log-Eintrag der Liste wird außerdem der Bereich **Logausriss** gefüllt, der die Logeinträge vor und nach dem ausgewählten Eintrag (definierbare Anzahl – "Zeilen vorher" / "Zeilen nachher") anzeigt (vgl. Abbildung 8-7).

 Durch einen Klick auf die Schaltfläche wird der ausgewählte Eintrag in der zugehörige Log-Datei angezeigt. Die Anzeige im Server-Monitoring wechselt dazu auf den Bereich "Logdateien / Historie" (siehe Kapitel 8.3.1.2 Seite 440).



8.3.2 Log-Dateien – nach Projekt



Für Projektadministratoren werden nur die Projekte aufgelistet, für die man als Administrator eingetragen ist.

Auf dieser Seite befindet sich eine Tabelle mit allen auf dem Server installierten Projekten. Angezeigt werden hier der Projektname und die Projektbeschreibung.

Mit einem Klick auf den Verweis [Logdatei anzeigen](#) hinter jedem Projekteintrag wird die aktuelle Logansicht für das ausgewählte Projekt angezeigt.

Mit einem Klick auf den Verweis [Projektetails](#) können weitere Informationen zum ausgewählten Projekt aufgerufen werden (siehe Kapitel 8.2.1.1 ab Seite 434).

Diese Ansicht ist unterteilt in die Bereiche:

- Aktuelle Ansicht
- Historie
- Suche

 Über dieses Icon können Log-Dateien innerhalb der Anwendung zum Server-Monitoring heruntergeladen und/oder als Mailanhang verschickt werden (Beschreibung siehe Kapitel 8.3.1).

8.3.2.1 Aktuelle Ansicht

Auf dieser Seite werden die aktuellsten Aktionen der Log-Datei für das ausgewählte Projekt ausgegeben. Über der Log-Liste wird der Name des ausgewählten Projektes angegeben. Über den Verweis [Wechseln](#) hinter dem Projektnamen kann ein anderes Projekt zur Ansicht ausgewählt werden.

Eine ausführliche Dokumentation zum Aufbau der aktuellen Log-Ansicht befindet sich in Kapitel 8.3.1.1 ab Seite 439.

8.3.2.2 Historie

Auf dieser Seite ist es möglich, sich einen älteren Teil der Log-Datei für das ausgewählte Projekt ausgeben zu lassen. Über den Verweis [Wechseln](#) hinter dem Projektnamen kann ein anderes Projekt zur Ansicht ausgewählt werden.



Eine ausführliche Dokumentation zum Aufbau der historischen Log-Ansicht befindet sich in Kapitel 8.3.1.2 ab Seite 440.

8.3.2.3 Suche

Auf dieser Seite ist es möglich, sich einen bestimmten Zeitraum der Log-Datei für das ausgewählte Projekt ausgeben zu lassen. Über den Verweis [Wechseln](#) hinter dem Projektnamen kann ein anderes Projekt zur Ansicht ausgewählt werden.

Eine ausführliche Dokumentation zur Suche nach bestimmten Aktionen der Log-Datei befindet sich in Kapitel 8.3.1.3 ab Seite 441.

8.3.3 Log-Datei – nach Veröffentlichung



Für Projektadministratoren werden nur die Projekte aufgelistet, für die man als Administrator eingetragen ist.

Durch einen Klick auf den Projektnamen werden die Detailinformationen zu diesem Projekt angezeigt. (Siehe Kapitel 8.2.1.1 ab Seite 434)

Auf dieser Seite befindet sich eine Tabelle mit allen auf dem Server installierten Projekten. Angezeigt werden hier der Projektname und die Projektbeschreibung.

Mit einem Klick auf den Verweis [Logdatei anzeigen](#) hinter jedem Projekteintrag wird die aktuelle Logansicht für das ausgewählte Projekt angezeigt.

Mit einem Klick auf den Verweis [Projektdetails](#) können weitere Informationen zum ausgewählten Projekt aufgerufen werden (siehe Kapitel 8.2.1.1 ab Seite 434).

Diese Ansicht ist unterteilt in die Bereiche:

- Aktuelle Ansicht
- Historie

 Über dieses Icon können Log-Dateien innerhalb der Anwendung zum Server-Monitoring heruntergeladen und/oder als Mailanhang verschickt werden (Beschreibung siehe Kapitel 8.3.1).



8.3.3.1 Aktuelle Ansicht

Auf dieser Seite wird die Log-Datei einer aktuell laufenden Veröffentlichung für das ausgewählte Projekt ausgegeben. Über der Log-Liste wird der Name des ausgewählten Projektes angegeben. Über den Verweis Wechseln hinter dem Projektnamen kann ein anderes Projekt zur Ansicht ausgewählt werden.

Eine ausführliche Dokumentation zum Aufbau der aktuellen Log-Ansicht befindet sich in Kapitel 8.3.1.1 ab Seite 439.

8.3.3.2 Historie

Auf dieser Seite ist es möglich, sich die Log-Datei einer älteren Veröffentlichung ausgeben zu lassen. Angezeigt wird eine Liste mit allen auf dem Server installierten Projekten.

Oberhalb der Projektliste kann der Zeitraum festgelegt werden, in dem die gesuchten Veröffentlichungen stattgefunden haben sollen. Außerdem kann angegeben werden, wie viele Veröffentlichungen maximal für jedes Projekt aufgelistet werden sollen. Durch einen Klick auf die Schaltfläche **Suchen** wird die Projektliste aktualisiert.

Abbildung 8-8: Projektauswahl für Veröffentlichungen

Hinter dem Projektnamen wird angezeigt, wie viele Log-Dateien zu dem Projekt vorhanden und in welchem Zeitraum sie entstanden sind. Durch einen Klick auf das Plus [+] vor jedem Projektnamen werden die vorhandenen Veröffentlichungen absteigend sortiert dargestellt. Durch einen Klick auf den Verweis Historie anzeigen, hinter jedem Eintrag, wird die zugehörige Log-Datei angezeigt. Durch einen Klick auf



den Eintrag [Download](#) kann die zugehörige Log-Datei heruntergeladen werden.

Über den Verweis [Wechseln](#) hinter dem Projektnamen kann ein anderes Projekt bzw. eine andere Veröffentlichung zur Ansicht ausgewählt werden.

Eine ausführliche Dokumentation zum Aufbau einer historischen Log-Ansicht befindet sich in Kapitel 8.3.1.2 ab Seite 440.

8.4 Zeitplanung

8.4.1 Zeitplanung – Übersicht

Auf dieser Seite wird eine Übersicht über die gesamte Zeitplanung dargestellt.

Übersicht

Aufträge:

Name	Projektname	Typ	Status	letzte Ausführung	Letzte Dauer	nächstes Update
Repair references	Dokumentation WebClient (PRODUKTIV)	fest	aktiv	22.03.2012 10:12	1s 954ms	unbekannt
Generate WebEdit Tooltips	Dokumentation WebClient (PRODUKTIV)	täglich	aktiv	16.07.2012 18:00	38s 875ms	17.07.2012 18:00
Generate PDF Document	Dokumentation WebClient (PRODUKTIV)	täglich	aktiv	16.07.2012 18:00	1m 15s 865ms	17.07.2012 18:00
generate (daily)	Dokumentation WebClient (PRODUKTIV)	täglich	aktiv	16.07.2012 18:00	1m 10s 85ms	17.07.2012 18:00

Abbildung 8-9: Auftragsübersicht des Servers

In einer Tabelle werden alle auf dem Server eingestellten Aufträge aufgelistet (siehe Kapitel 7.5.1 Seite 351). Neben dem Projektnamen wird angezeigt, in welchem Intervall die Aufträge durchgeführt werden und in welchem Status sie sich aktuell befinden. Außerdem werden Informationen über den Zeitpunkt der letzten geplanten Ausführung und deren Dauer sowie der Zeitpunkt der nächsten geplanten Ausführung dargestellt.

Diese Liste kann nach allen vorhandenen Spalten sortiert werden.



Für Projektadministratoren werden nur Aufträge von Projekten aufgelistet, für die man als Administrator eingetragen ist.

Durch einen Klick auf den Projektnamen eines aufgelisteten Projektes werden die Detailinformationen zu diesem Projekt angezeigt (siehe Kapitel 8.2.1.1 ab Seite 434). Durch einen Klick auf den Auftragsnamen werden Detailinformationen



zum Auftrag angezeigt.

Unterhalb der ersten Tabelle befindet sich noch eine Tabelle mit den ausgeführten Aufträgen, z. B. den Veröffentlichungen, innerhalb der letzten 48 Stunden.

Ausgeführte Aufträge in den letzten 48 Stunden:

Name	Projektname	Status	Beginn	Ende	Dauer
generate (daily)	Online-Dokumentation FirstSpirit...	aktiv	16.07.2012 18:00:00	16.07.2012 18:04:27	4m 27s 280ms
build dvd-index	Online-Dokumentation FirstSpirit...	aktiv	16.07.2012 18:00:00	16.07.2012 18:00:29	29s 100ms
generate (daily)	Dokumentation WebClient (PRODUKTIV)	aktiv	16.07.2012 18:00:00	16.07.2012 18:01:10	1m 10s 85ms
Generate PDF Document	Dokumentation WebClient (PRODUKTIV)	aktiv	16.07.2012 18:00:00	16.07.2012 18:01:15	1m 15s 865ms
Generate WebEdit Tooltips	Dokumentation WebClient (PRODUKTIV)	aktiv	16.07.2012 18:00:00	16.07.2012 18:00:38	38s 875ms
generate_full	Online-Dokumentation FirstSpirit...	aktiv	16.07.2012 01:54:39	16.07.2012 02:01:18	6m 39s 106ms
generate_full	Dokumentation WebClient (PRODUKTIV)	aktiv	16.07.2012 01:54:39	16.07.2012 01:55:45	1m 6s 410ms

Abbildung 8-10: Zeitplanung – Ausführung in den letzten 48 Stunden

Neben dem Namen des Auftrags, gibt es Angaben über den Status der ausgeführten Aktionen sowie den Namen des Projekts für das der Auftrag ausgeführt wurde. Weiterhin wird der Start- und Endzeitpunkt der Ausführung des Auftrages angezeigt, sowie die Dauer der letzten Durchführung.

Log-Dateien zu den einzelnen Veröffentlichungs-Aufträgen stehen über den Menüeintrag "Logdateien / nach Veröffentlichung" zur Verfügung (siehe Kapitel 8.3.3 Seite 444).

8.4.2 Zeitplanung – geplante Aufträge

Auf dieser Seite wird eine Übersicht über die geplanten Aufträge dargestellt. Es werden maximal 25 Aufträge gleichzeitig dargestellt, zwischen den einzelnen Seiten kann über die Pfeiltaste oberhalb der Tabelle geblättert werden.

In einer Tabelle werden alle Aufträge aufgelistet, für die noch eine Ausführung geplant ist. Neben dem Projektnamen gibt es Angaben über die ID des Auftrags und die Art der geplanten Aktion. Weiterhin wird angezeigt, in welchem Intervall die Aufträge durchgeführt werden und zu welchem Zeitpunkt die nächste Ausführung geplant ist.

Diese Liste kann nach allen vorhandenen Spalten sortiert werden.

Durch einen Klick auf den Projektnamen eines aufgelisteten Projektes werden die Detailinformationen zu diesem Projekt angezeigt. (Siehe Kapitel 8.2.1.1 ab Seite 434)



8.4.3 Zeitplanung – ausgeführte Aufträge

Auf dieser Seite wird eine Übersicht über die bereits ausgeführten Aufträge dargestellt. Es werden maximal 25 Aufträge gleichzeitig dargestellt, zwischen den einzelnen Seiten kann über die Pfeiltaste oberhalb der Tabelle geblättert werden.

In einer Tabelle werden alle Aufträge aufgelistet, die bereits durchgeführt wurden.

Neben dem Namen des Auftrags, gibt es Angaben über den Namen des Projekts für das der Auftrag ausgeführt wurde. Weiterhin wird der Startzeitpunkt der Ausführung des Auftrages angezeigt, Status, Fehler, Benutzer sowie die Dauer der letzten Durchführung. Diese Liste kann nach Projektnamen, Auftragsnamen oder Ausführungsbeginn sortiert werden.

Durch einen Klick auf den Projektnamen eines aufgelisteten Projektes werden die Detailinformationen zu diesem Projekt angezeigt. (Siehe Kapitel 8.2.1.1 ab Seite 434)

Durch einen Klick auf den Verweis "Historie anzeigen" können die Logausgaben zum Zeitpunkt der Auftragsausführung angezeigt werden. Die Ausgabe der Logausgaben kann über die Checkboxen konfiguriert werden:

So ist es möglich, unterschiedliche **Log-Level** einzustellen. Soll beispielsweise gezielt nach Fehlermeldungen während einer Generierung gesucht werden, kann man das Loglevel auf "Error" stellen (siehe dazu Kapitel 4.3.6 Seite 97).

Die Anzahl der maximal angezeigten **Zeilen** der Logausgabe ist ebenfalls konfigurierbar. Die Anzeige wird bei einem Klick auf die Schaltfläche **Anzeigen** aktualisiert.

Pause Mit einem Klick auf die Schaltfläche kann die Ausgabe der Zeilen unterbrochen werden.

Details Mit einem Klick auf die Schaltfläche kann die zeilenbegrenzte auf eine erweiterte Ausgabe umgestellt werden.

Neben einer gefilterten Ausgabe, kann auch gezielt nach bestimmten Begriffen innerhalb des Logs gesucht werden. Das Ergebnis der Suche wird nach einem Klick auf die Schaltfläche **Suchen** im unteren Bereich der Seite unter **Suchergebnisse** dargestellt.



8.5 Benutzer

8.5.1 Benutzer – Suche

Auf dieser Seite wird eine sortierte Liste aller auf dem Server eingetragenen Benutzer angezeigt. Es werden maximal 25 Benutzer gleichzeitig dargestellt, zwischen den einzelnen Seiten kann über die Pfeiltaste oberhalb der Tabelle geblättert werden.

Neben dem Benutzernamen gibt es Angaben über das Kürzel, den Zeitpunkt des letzten Login und ob es sich um einen LDAP Benutzer handelt. Weiterhin werden die e-Mail Adresse und Telefonnummer angegeben.

Über die Suchfunktion kann die Auswahl der angezeigten Benutzer eingeschränkt werden. In das Suchfeld können Worte oder Wortteile angegeben werden, nach denen in der Spalte Benutzername gesucht wird. Durch einen Klick auf die Schaltfläche  wird die Suche durchgeführt.

Durch Anklicken eines Benutzers werden die Detailinformationen zu diesem Benutzer angezeigt. Dazu gehören neben den Angaben, die beim Neuanlegen eines Benutzer festgelegt werden (siehe Kapitel 7.2.4.1 ab Seite 225), auch eine Liste aller dem Benutzer zugänglichen Projekte auf diesem Server. Für jedes Projekt wird angegeben, welche Gruppenzugehörigkeit der Benutzer in diesem Projekt besitzt. Durch einen weiteren Klick auf ein Projekt des ausgewählten Benutzers wird eine ausführliche Übersicht zu diesem Projekt angezeigt. (Siehe Kapitel 8.2.1.1 ab Seite 434)

Durch einen Klick auf die e-Mail Adresse eines Benutzers, öffnet sich ein e-Mail Fenster, um dem Benutzer direkt eine Nachricht schicken zu können.



8.6 FirstSpirit

8.6.1 FirstSpirit – Konfiguration

Die Konfiguration des FirstSpirit-Servers, der Datenbankanbindung, des Anmeldevorgangs und einige weitere Einstellungen finden über spezielle Konfigurationsdateien im Installationsverzeichnis des FirstSpirit-Servers statt. Von einer direkten Bearbeitung dieser Konfigurationsdateien wird abgeraten (siehe Kapitel 4.2 Seite 29).

Die FirstSpirit Server- und Projektkonfiguration bietet eine komfortable Möglichkeit zum Bearbeiten der Konfigurationseinstellungen für die Konfiguration der Datenbankanbindung (`fs-database.conf`) und die Konfiguration des Anmeldevorgangs (`fs-jaas.conf`) (siehe Kapitel 6 Seite 185). Weitere Konfigurationseinstellungen können über die JMX-Konsole vorgenommen werden (siehe Kapitel 9 Seite 444).

Das FirstSpirit Server-Monitoring bietet folgende Konfigurationsmöglichkeiten:

- | | | |
|-------------------------|--|---------------------------|
| • Server | (Datei: <code>fs-server.conf</code>) | Kapitel 8.6.1.1 Seite 450 |
| • Lizenz | (Datei: <code>fs-license.conf</code>) | Kapitel 8.6.1.2 Seite 451 |
| • Protokollierung | (Datei: <code>fs-logging.conf</code>) | Kapitel 8.6.1.3 Seite 452 |
| • System | (keine Konfiguration) | Kapitel 8.6.1.4 Seite 453 |
| • Startoptionen | (Datei: <code>fs-wrapper.conf</code>) | Kapitel 8.6.1.5 Seite 454 |
| • Webapplikationen | (Datei: <code>fs-webapps.xml</code>) | Kapitel 8.6.1.6 Seite 455 |
| • Dienste | (Konfig. Systemdienste) | Kapitel 8.6.1.7 Seite 455 |
| • Anmeldekongfiguration | (Datei <code>fs-jaas.conf</code>) | Kapitel 8.6.1.8 Seite 456 |



Diese Menüebene steht nur Serveradministratoren zur Verfügung

8.6.1.1 Server

Die Konfiguration des FirstSpirit-Servers erfolgt über die Konfigurationsdatei `fs-server.conf`, die sich im Installationsverzeichnis des FirstSpirit-Servers befindet (siehe Kapitel 4.3, Seite 30).

Die Datei kann mit einem Klick auf den Eintrag "Server" zum Bearbeiten geöffnet werden. Die einzelnen Parameter zur Konfiguration des FirstSpirit-Servers werden in



Kapitel 4.3.1 ff. erläutert. Zur Realisierung der Änderungen ist teilweise ein Neustart des Servers erforderlich (z. B. bei einer Änderung des Ports), andere Änderungen können während des laufenden Serverbetriebs erfolgen. Dazu müssen die Änderungen gespeichert und die Konfigurationsdatei auf dem Server neugeladen werden.

Durch einen Klick auf die Schaltfläche **Speichern** werden die aktuellen Änderungen der Server-Konfiguration gespeichert und in den laufenden Serverbetrieb eingespielt.

Durch einen Klick auf die Schaltfläche **Abbrechen** kann der Benutzer zurück auf die Konfigurationsübersicht wechseln. Aktuelle Änderungen an der Konfigurationsdatei, die nicht bereits gespeichert wurden, werden zurückgesetzt.

Server neustarten: Ist diese Option aktiviert, dann werden durch einen Klick auf die Schaltfläche **Speichern** die aktuellen Änderungen der Server-Konfiguration gespeichert und anschließend der Server neu gestartet. (Der Start des FirstSpirit-Servers wird über die Konfigurationsdatei `fs-wrapper.conf` gewährleistet, siehe Kapitel 8.6.1.5).

8.6.1.2 Lizenz

Im Unterverzeichnis `conf` des FirstSpirit-Servers befindet sich die Datei `fs-license.conf`. Die Datei enthält die aktuelle FirstSpirit Lizenz und kann bei Bedarf mit einem Klick auf den Eintrag "Lizenz" über das FirstSpirit Server-Monitoring angezeigt werden. Die zugehörigen Parameter werden in Kapitel 4.3.5, ab Seite 94 erläutert.

Soll eine neue Lizenzdatei auf dem Server eingespielt werden, muss der Inhalt vollständig und unverändert auf dieser Seite eingefügt werden. Durch einen Klick auf die Schaltfläche **Speichern** wird die neue Lizenzdatei gespeichert.

Durch einen Klick auf die Schaltfläche **Abbrechen** kann der Benutzer zurück auf die Konfigurationsübersicht wechseln. Aktuelle Änderungen an der Lizenzdatei, die nicht bereits gespeichert wurden, werden zurückgesetzt.

Änderungen an den Konfigurationsdateien werden im laufenden Betrieb in festgelegten Zeitintervallen automatisch auf dem Server aktualisiert. Ein Neustart des FirstSpirit-Servers ist daher nicht erforderlich.





Manipulationen an der `fs-license.conf` führen zu einer ungültigen Lizenz. Sollten Änderungen notwendig werden (z. B. Wechsel der IP-Adresse), wenden Sie sich bitte an den Hersteller.

8.6.1.3 Protokollierung

Im Unterverzeichnis `conf` des FirstSpirit-Servers befindet sich die Datei `fs-logging.conf`, die wichtige Konfigurationseinstellungen für die "Log"-Ausgaben enthält und bei Bedarf angepasst werden muss. Die zugehörigen Parameter werden im Kapitel 4.3.6, ab Seite 97, erläutert.

Standardmäßig werden auftretende Fehler und Infomeldungen dem Loggingsystem "log4j"³⁵ übergeben. Über das Framework kann eine Gewichtung der Log-Ausgaben vorgenommen werden. Zur Verfügung stehen dabei die Kategorien (Log-Level) `DEBUG`, `INFO`, `ERROR`. Potentiell können aber noch weitere Kategorien konfiguriert werden (z. B. `FATAL`, `WARN`). Ausnahmen bilden die beiden Stufen `ALL` und `OFF`, die das Logging entweder komplett deaktivieren (`OFF`) oder alle Meldungen ungefiltert ausgeben (`ALL`). Die Filterung und Art der Ausgabe kann zur Laufzeit konfiguriert werden.

Neben dieser Standardprotokollierung können weitere Log-Dateien konfiguriert werden. Hinter dem Namen jeder Log-Datei steht in Klammern, in welchem Status sich diese Datei befinde (aktiv | inaktiv). Eine inaktive Log-Datei kann durch einen Klick auf die Schaltfläche **Verwenden** aktiviert werden. Die bisher aktivierte Log-Datei wird dann deaktiviert, da zu jedem Zeitpunkt nur genau eine Protokollierung aktiv sein kann.

Durch einen Klick auf die Schaltfläche **Editieren** könne Änderungen an der Log-Datei vorgenommen werden.

Durch einen Klick auf die Schaltfläche **Speichern** werden die Änderungen gespeichert. Änderungen an den Konfigurationsdateien werden im laufenden Betrieb in festgelegten Zeitintervallen automatisch auf dem Server aktualisiert. Ein Neustart des FirstSpirit-Servers ist daher nicht erforderlich.

³⁵ Weiterführende Informationen <http://logging.apache.org/log4j/>



Durch einen Klick auf die Schaltfläche **Abbrechen** kann der Benutzer zurück auf die Konfigurationsübersicht wechseln. Aktuelle Änderungen an der Konfigurationsdatei, die nicht bereits gespeichert wurden, werden zurückgesetzt.

8.6.1.4 System

Auf dieser Seite befinden sich alle relevanten Systeminformationen des Servers.



System

System-Informationen **Datei-Download**

Betriebssystem	
Java Virtual Machine	Java HotSpot(TM) Server VM 20.2-b06
Java Runtime	Java(TM) SE Runtime Environment 1.6.0_27-b07
Betriebssystem	Windows 7 () x86 6.1
Prozessoren	4
Aktuelle Zeit	Dienstag, 17.07.2012 17:03:51
Arbeitsordner	D:\Firstspirit_Server\Firstspirit_5

Systemeigenschaften	
backup_files	50
cmsroot	D:\Firstspirit_Server\Firstspirit_5
com.sun.management.jmxremote	
com.sun.management.jmxremote.authenticate	false

Abbildung 8-11: Server Monitoring – Systeminformationen und Datei-Download

Betriebssystem: Hier werden Informationen zum zugrunde liegenden Betriebssystem dargestellt, z. B. verwendetes JDK, Server-Betriebssystem, Anzahl der Prozessoren.

Systemeigenschaften: Hier werden Systemeigenschaften (Informationen zu Verzeichnispfade, Portnummern, Classpath, usw.) in einer übersichtlichen Liste angezeigt.

Die dargestellten Systemeigenschaften können in dieser Übersicht nicht konfiguriert werden, sondern dienen nur der Information.

System-Informationen: Mit einem Klick auf den Button "System-Informationen" werden weitere Systeminformationen angezeigt.

Datei-Download: Mit einem Klick auf den Button "Datei-Download" öffnet sich der Dateiauswahldialog des Server-Log-Verzeichnisses (.../server/log). Hier können die Server-Log-Dateien (fs-server.log) und weitere Log-Dateien heruntergeladen werden, die über das Server Monitoring nicht verfügbar sind (beispielsweise fs-gc.log).



8.6.1.5 Startoptionen

Im Unterverzeichnis `conf` des FirstSpirit-Servers befindet sich die Datei `fs-wrapper.conf`, die wichtige Konfigurationseinstellungen für den Serverstart enthält und bei Bedarf angepasst werden muss. Die zugehörigen Parameter werden in Kapitel 4.3.2, ab Seite 66, erläutert.

Die Konfigurationsdatei ist zuständig für Start und Stopp des Java-Prozesses. Je nach Konfiguration kann bei Bedarf das Starten der VM definiert und zusätzlich die entsprechenden Log-Ausgaben in die Standardausgabe geschrieben werden.

Mit einem Klick auf **Editieren** kann die bestehende Konfiguration bearbeitet werden. Es öffnet sich ein Bearbeitungsfenster mit der Möglichkeit die Inhalte der Konfigurationsdatei `fs-wrapper.conf` zu bearbeiten.

Liste Mit einem Klick auf den Button wird der Bearbeitungsmodus wieder verlassen und zurück in die übersichtliche Listenansicht gewechselt.

Mit einem Klick auf **Speichern**, werden die Änderungen in der Konfigurationsdatei `fs-wrapper.conf` gespeichert. Werden ungültige Konfigurationen vorgenommen, wird beim Speichern der Konfiguration im Server-Monitoring auf den Fehler hingewiesen und der Speichervorgang wird nicht ausgeführt:

```
unexpected configuration property key 'wrapper.startUp.timeout' in line 76
```

Durch einen Klick auf die Schaltfläche **Abbrechen** kann der Benutzer zurück auf die Konfigurationsübersicht wechseln. Aktuelle Änderungen an der Konfigurationsdatei, die nicht bereits gespeichert wurden, werden zurückgesetzt.

Server neustarten: Ist diese Option aktiviert, dann werden durch einen Klick auf die Schaltfläche **Speichern** die aktuellen Änderungen der Server-Konfiguration gespeichert und anschließend der Server neu gestartet.



Bei allen Änderungen der Konfigurationsdatei `fs-wrapper.conf` ist ein Neustart des Servers erforderlich.



8.6.1.6 Webapplikationen

Im Unterverzeichnis `conf` des FirstSpirit-Servers befindet sich die Datei `fs-webapp.xml`, die wichtige Konfigurationseinstellungen für den internen Webserver enthält und bei Bedarf angepasst werden muss. Die zugehörigen Parameter werden in Kapitel 4.3.7, ab Seite 100, erläutert.

Mit einem Klick auf die Schaltfläche **Speichern**, werden die Änderungen in der Datei `fs-webapp.xml` gespeichert.

Durch einen Klick auf die Schaltfläche **Abbrechen** kann der Benutzer zurück auf die Konfigurationsübersicht wechseln. Aktuelle Änderungen an der Konfigurationsdatei, die nicht bereits gespeichert wurden, werden zurückgesetzt.



Damit die Änderungen auf dem Server aktualisiert werden, ist ein Neustart des Jettys erforderlich. Nach dem Speichern erscheint eine Meldung und der Verweis in den Menüeintrag Steuerung / Webapplikationen (siehe Kapitel 8.6.2.2 Seite 460). Dort besteht die Möglichkeit zum Neustart des Jettys (innerhalb der Server-Monitoring Umgebung).

8.6.1.7 Dienste

Über diesen Eintrag können die FirstSpirit Systemdienste konfiguriert werden. Ein Dienst ist eine Serverkomponente, die über eine öffentliche Schnittstelle aus Eingabekomponenten oder Skripten angesprochen werden kann. (Beispiele sind die Rechtschreibprüfung oder der Dienst der Rechte-Eingabekomponente `CMS_INPUT_PERMISSION`, siehe Kapitel 7.3.11 Seite 254.)

In der Tabelle werden alle auf dem Server verfügbaren Dienste angezeigt:

Übersicht über die konfigurierbaren Dienste:

Name: Name des Systemdiensts. Mit einem Klick auf den Eintrag öffnet sich ein weiteres Fenster mit Konfigurationsmöglichkeiten für den jeweiligen Dienst (s.u. "Konfiguration eines Dienstes").

Kommentar: Optionaler Kommentar zum Dienst.

Typ: Name des Moduls zu dem der Dienst gehört. Mit System bezeichnete Dienste gehören zum Standard-Systemmodul von FirstSpirit.



Autostart: In diesem Bereich kann der automatische Start für einen Dienst aktiviert bzw. deaktiviert werden. Die Funktion kann analog zu "Autostart aktivieren" bzw. "Autostart deaktivieren" innerhalb der Server- und Projektkonfiguration verwendet werden (siehe Kapitel 7.3.11 Seite 254).

Konfiguration eines Dienstes:

Dateiname: Dateiname des Dienstes. Mit einem Klick auf den Dateinamen öffnet sich ein weiteres Fenster zum Bearbeiten der Datei, beispielsweise der groups.xml-Datei des PermissionService.

Mit einem Klick auf die Schaltfläche **Speichern**, werden die Änderungen in der Datei gespeichert.

Durch einen Klick auf die Schaltfläche **Abbrechen** kann der Benutzer zurück auf die Dateiübersicht wechseln. Aktuelle Änderungen, die nicht bereits gespeichert wurden, werden zurückgesetzt.

Größe: Dateigröße

Letzte Änderung: Datum der letzten gespeicherten Version der Datei.

Verzeichnis anlegen: Über diesen Button kann ein Verzeichnis zum jeweiligen Dienst angegeben werden. Das Verzeichnis wird im Dateisystem unterhalb des Modulverzeichnisses mit dem eingegebenen Namen angelegt.

Datei erstellen: Über diesen Button kann eine Datei zum jeweiligen Dienst erstellt werden. Die Datei wird im Dateisystem unterhalb des Modulverzeichnisses mit dem eingegebenen Namen angelegt.

8.6.1.8 Anmeldekonfiguration

Im Unterverzeichnis `conf` des FirstSpirit-Servers befindet sich die Datei `fs-jaas.conf`, die wichtige Konfigurationseinstellungen für den Anmeldevorgang enthält und bei Bedarf angepasst werden muss. Die zugehörigen Parameter werden in Kapitel 4.3.4, ab Seite 77, erläutert.

Mit einem Klick auf die Schaltfläche **Speichern**, werden die Änderungen in der Datei `fs-jaas.conf` gespeichert. Die Datei wird automatisch auf dem FirstSpirit-Server aktualisiert. Ein Neustart des Servers ist nicht erforderlich.



Durch einen Klick auf die Schaltfläche **Abbrechen** kann der Benutzer zurück auf die Konfigurationsübersicht wechseln. Aktuelle Änderungen an der Konfigurationsdatei, die nicht bereits gespeichert wurden, werden zurückgesetzt.

8.6.2 FirstSpirit – Steuerung

Über den Menüeintrag Steuerung kann nach einer Änderung an der Server-Konfiguration oder nach einer Aktualisierung der FirstSpirit-Server gesteuert werden.



Diese Menüebene steht nur Serveradministratoren zur Verfügung.

8.6.2.1 Wartungsmodus

Die Funktion dient zum Aktivieren bzw. Deaktivieren des Wartungsmodus des FirstSpirit-Servers. Der Wartungsmodus kann eingesetzt werden:

- Bei geplanten Aktualisierungen des FirstSpirit-Servers (um den Server im laufenden Betrieb geordnet runterzufahren)
- Um den Zugriff auf Projekte beschränken (z. B. bei Umbauarbeiten oder größeren Updates).

Der Wartungsmodus wird über die Servereigenschaften in der Server- und Projektkonfiguration konfiguriert. Dazu wird einem Auftrag (über die Auftragsverwaltung) die Aktion "Wartungsmodus" zugewiesen (Konfiguration siehe Kapitel 7.5.9.5 Seite 376).

Über das FirstSpirit Server-Monitoring können

- vorhandene Wartungsmodus-Aufträge gestartet
- der aktuelle Status von Wartungsmodus-Aufträgen eingesehen und
- laufende Wartungsmodus-Aufträge gestoppt

werden:



Wartungsmodus

Wartungsmodus-Auftrag:

Geplante Wartungszeiträume

Auftrag	Projekte	Nächster Schritt	Wartungszeitraum
Serveraktualisierung	alle	Erster Hinweis anzeigen 04.09.2012 23:02:00	04.09.2012 23:12:00 Dauer: 30 Minuten

Aktive Wartungszeiträume

Auftrag	Status	Projekte	Nächster Schritt	Wartungszeitraum	Aktionen
Projekterweiterung	Im Vorlauf Sitzungsende-Hinweis	Mithras Energy (ID: 11)	Sitzungen ablehnen 04.09.2012 14:36:33	04.09.2012 14:40:33 Dauer: 60 Minuten	Wartungsmodus beenden

Abbildung 8-12: Server-Monitoring – Steuerung – Wartungsmodus

ausführen: Wurde ein Wartungsmodus-Auftrag erstellt (siehe Kapitel 7.5.9.5 Seite 376), kann er aus der Drop-Down-Liste "Wartungsmodus-Auftrag" ausgewählt und über die Schaltfläche "ausführen" gestartet werden. Es wird die Nachricht "Wartungsmodus gestartet" ausgegeben. Über diese Schaltfläche kann ein Auftrag auch mehrfach gleichzeitig gestartet werden.

Geplante Wartungszeiträume: Hier werden geplante Wartungsmodus-Aufträge mit den Einstellungen, die in den Server-Eigenschaften vorgenommen wurden, angezeigt.

Aktive Wartungszeiträume: Hier werden laufende Wartungsmodus-Aufträge mit den Einstellungen, die in den Server-Eigenschaften vorgenommen wurden, angezeigt.

Status: In dieser Spalte wird angezeigt, in welchem Status sich der Auftrag aktuell befindet (für eine aktuelle Anzeige die Schaltfläche "Aktualisieren" betätigen). Je nach Einstellungen im Auftrag werden dabei die Stati in folgender Reihenfolge angezeigt:

- *Im Vorlauf | Inaktiv:* Der Auftrag ist gestartet, aber es gibt noch keine Auswirkungen für die Benutzer.
- *Im Vorlauf | Erster Hinweis anzeigen:* In diesem Status wird als Vorwarnung die Meldung "In x Minuten werden Wartungsarbeiten am FirstSpirit-Server | an diesem Projekt durchgeführt. Geplante Dauer: ca. y Minuten." in den Clients und ggf. auf der Startseite angezeigt. Siehe Option "Warnung anzeigen nach" im Auftrag (Kapitel 7.5.9.5 Seite 376).
- *Im Vorlauf | Sitzungsende-Hinweis:* In diesem Status wird die Meldung "Der



FirstSpirit-Server | Das Projekt ist wegen Wartungsarbeiten ab 21.08.2012 15:15:00 nicht verfügbar. Geplante Dauer: ca. y Minuten. Bitten beenden Sie Ihre Sitzung." in den Clients und ggf. auf der Startseite angezeigt. Siehe Option "Sitzungsendewarnung anzeigen nach" im Auftrag (Kapitel 7.5.9.5 Seite 376).

- *Im Vorlauf | Sitzungen ablehnen:* In diesem Status sind keine Anmeldungen mehr möglich. Siehe Option "Neue Sitzungen ablehnen nach" im Auftrag (Kapitel 7.5.9.5 Seite 376).
- *Aktiv | Wartungsmodus:* In diesem Status ist der Wartungsmodus aktiv. Nun können die Wartungsarbeiten vorgenommen und bei Bedarf der FirstSpirit-Server heruntergefahren werden. Siehe Option "Wartungsmodus starten nach" im Auftrag (Kapitel 7.5.9.5 Seite 376).
- *Ende:* In diesem Status ist wieder ein Anmelden am Server möglich. Siehe Option "Geplante Dauer" im Auftrag (Kapitel 7.5.9.5 Seite 376).

Projekte: In dieser Spalte wird angezeigt, welche Projekte der Wartungsmodus-Auftrag betrifft (Option "Gültig für Projekte" im Auftrag, siehe Kapitel 7.5.9.5 Seite 376).

Nächster Schritt: In dieser Spalte wird angezeigt, welche Stufe die nächste ist und wann sie beginnt. Die Stufen sind unter "Status" (siehe oben) beschrieben.

Wartungszeitraum: In dieser Spalte wird angezeigt, wann der Wartungszeitraum beginnt (ergibt sich aus dem Startzeitpunkt des Auftrags und den Zeitangaben für die Stufen 1. bis 4.) und wie lange er dauert (Option "Geplante Dauer" im Auftrag, siehe Kapitel 7.5.9.5 Seite 376). Je nach Konfiguration der Optionen "Aufträge nicht annehmen" und "Sitzungen beenden, neue Sitzungen ablehnen" werden Aufträge, die in den Wartungszeitraum fallen, abgebrochen und sind während des Wartungszeitraums keine Anmeldungen am Server bzw. an den ausgewählten Projekten möglich.

Wartungsmodus beenden: Mit einem Klick auf diesen Verweis wird der betreffende Auftrag beendet. Im Server-Monitoring wird die Nachricht "Wartung beendet" angezeigt. In den Clients und auf der Startseite wird die Nachricht "Die Wartungsarbeiten am FirstSpirit-Server wurden beendet" ausgegeben.

Aktualisieren: Über diese Schaltfläche kann die Ansicht aktualisiert werden.



8.6.2.2 Web-Applikationen

In der Ansicht werden die auf dem Server konfigurierten Web-Applikationen angezeigt (vgl. Kapitel 7.3.13 Seite 269). Standardmäßig sind die FirstSpirit Web-Applikationen (also *fs5root*, *fs5preview*, *fs5staging*, *fs5webedit*, *fs5webmon*) vorhanden.

Neu starten: Mit einem Klick auf den Verweis kann die betreffende Web-Applikation neu gestartet werden.

Jetty neu starten: Mit einem Klick auf diesen Button kann der interne Jetty neu gestartet werden. Ein Neustart ist beispielsweise erforderlich, damit Änderungen an der die Datei *fs-webapp.xml* auf dem Server aktualisiert werden (siehe Kapitel 8.6.1.6 Seite 455).

8.6.2.3 Aktualisierung



Diese Menüebene steht nur Serveradministratoren zur Verfügung.

Außerdem ist die Aktualisierungsfunktionalität ist nur verfügbar, wenn der FirstSpirit-Server über den Java-Wrapper gestartet wurde (Konfiguration siehe Kapitel 4.3.2).

Die Funktion dient zur Aktualisierung des FirstSpirit-Servers auf eine neue Software-Versionen.

► Wartungs-Modus ► Web-Applikationen ► **Aktualisierung** ► Dienste ► Server Neustart
► AppCenter Lizenzen ► Netzwerk

Aktualisierung

Aktuelle Server-Version: 5.0_BETA.34.52075

fs-server.jar:

Abbildung 8-13: Server-Monitoring – Steuerung – Aktualisierung (initiale Ansicht)



Durchsuchen: Mit einem Klick auf den Button öffnet sich ein Fenster zur Auswahl einer neuen jar-Datei (fs-server.jar) oder einer neuen Modul-Datei (.fsm). Die Auswahl der Dateien erfolgt über das lokale Dateisystem.

Modul-Datei hinzufügen: Mit jedem Klick auf den Button wird ein neuer Eingabebereich für eine Modul-Datei angelegt. Die Auswahl der Modul-Datei erfolgt analog zur Auswahl der Datei fs-server.jar über den Button "Durchsuchen".

Hinweis: Dieser Bereich dient nicht der Installation von Modulen. Das bedeutet, es können nur Module aktualisiert werden, die bereits auf dem Server installiert wurden. Hier hinzugefügte Modul-Dateien, die nicht bereits in einer früheren Version auf dem Server installiert wurden, werden nicht installiert oder aktualisiert.

Löschen: Bereits ausgewählte Dateien können mit einem Klick auf diesen Button wieder entfernt werden.

► Wartungs-Modus ► Web-Applikationen ► **Aktualisierung** ► Dienste ► Server Neustart
► AppCenter Lizenzen ► Netzwerk

Aktualisierung

Aktuelle Server-Version: 5.0_BETA.34.52075

fs-server.jar:

fs-spell.fsm:

Modul-Datei:

Abbildung 8-14: Server-Monitoring – Steuerung – Aktualisierung (nach Dateiauswahl)

Aktualisierung starten: Nach Auswahl der Dateien kann der Server über den Button "Aktualisierung starten" aktualisiert werden. Dazu werden nach einem Klick auf den Button zunächst die Dateien hochgeladen. Im Anschluss erfolgt eine interne Integritäts- und Signaturprüfung. Ist diese erfolgreich wird der Server heruntergefahren und anschließend wieder automatisch gestartet.



Alle Clients, die am Server angemeldet sind, sollten vor einer Server-Aktualisierung beendet werden, um Datenverluste zu vermeiden. Dazu kann der Wartungsmodus verwendet werden (siehe Kapitel 8.6.2.1 Seite 457).



Die Dateien werden vor dem Herunterfahren unter:

~\server\update\server\lib\ bzw.

~\server\update\data\modules\update\

zwischengespeichert und beim Neustart von dort geladen. Bei einer erfolgreichen Aktualisierung werden die Dateien automatisch aus diesen Verzeichnissen entfernt. Schlägt die Aktualisierung fehl, können hier liegengebliebenen jar- und fsm-Dateien manuell aus diesen Verzeichnissen gelöscht werden (Hinweis: Die Dateien fs-update.conf und fs-update.jar dürfen nicht entfernt werden).

Aktualisierung der installierten Webapplikationen: Alle installierten FirstSpirit-Webapplikationen, die unterhalb von ~fs5\web liegen, stehen unter Server-Versionsverwaltung und werden beim Serverstart neu ausgerollt. Gibt es im jeweiligen Applikations-Verzeichnis (unterhalb von ~fs5\web) eine Datei `version.txt`, wird vor der Aktualisierung zunächst anhand der Versionsnummer in der `version.txt` überprüft, ob eine neuere Version der installierten Applikation vorliegt. Ist das der Fall, wird die neue Version ausgerollt. Die Webserver-Steuerung übernimmt anschließend die Aktualisierung (und den Neustart) der Webapplikation auf einem oder mehreren Webservern (ausgenommen für externe Webserver, vgl. Kapitel 7.3.12 Seite 261).

Aktualisierung in einem Clusterverbund: Handelt es sich beim aktualisierten FirstSpirit-Server um eine Cluster-Master-Server in einem Clusterverbund, werden die beteiligten Cluster-Slave-Server ebenfalls automatisch aktualisiert und neu gestartet. Das bedeutet, wird der Master-Server beendet, werden auch alle Slave-Server automatisch beendet und starten nach einer kurzen Wartezeit von 60 s erneut.

Hinweis: Die Aktualisierung der Slave-Server wird nur für die Standardkonfiguration ausgeführt, wenn der Cluster-Master-Server und die Cluster-Slave-Server ein geteiltes Dateisystem verwenden (Clustering siehe Kapitel 7.3.14 Seite 275). In diesem Fall greifen die Slave-Server auf das `fs-server.jar` des Master-Servers zu und werden so bei einer Aktualisierung des Master-Servers ebenfalls aktualisiert. Um Konflikte während des Update-Prozesses auszuschließen, arbeiten die Slave-Server während des Neustarts des Masterservers in einen Zwischenzustand mit einem eigenen `update.jar`. Das gemeinsam verwendete `fs-server.jar` ist so beim Neustart des Master-Servers nicht in Benutzung und kann aktualisiert werden.





Bei einer Aktualisierung auf eine andere Major-Version muss auch die Lizenzdatei `fs-license.conf` ausgetauscht werden. Darüber hinaus sollten die Release-Notes zu der Version, auf die aktualisiert werden soll, vorab gelesen werden.

Mithilfe der Funktion Server-Monitoring-Steuerung-Netzwerk können weitere Server zur Aktualisierung ausgewählt werden (siehe Kapitel 8.6.2.7 Seite 465).

8.6.2.4 Dienste

Die Funktion dient zur Steuerung der Dienste des FirstSpirit-Servers. Ein Dienst (oder Service) ist eine Serverkomponente, die über eine öffentliche Schnittstelle aus Eingabekomponenten oder Skripten angesprochen werden kann. Beispiele sind die Rechtschreibprüfung oder der Service der Rechte-Eingabekomponente. Das System-Modul mit den Standard-Services (z. B. der PermissionService) ist bereits im Standard-Lieferumfang von FirstSpirit enthalten und steht nach der Installation zur Verfügung. Es können hier aber auch weitere Services angezeigt werden, die nachträglich auf dem Server installiert wurden (siehe Kapitel 7.3.11 Seite 254). Services sind immer systemweit gültig.

Name: Name des Dienstes.

Kommentar: Beschreibung zum jeweiligen Dienst.

Typ: Name des Moduls, zu dem der Dienst gehört.

Start/Stop: Funktion zum Starten oder Stoppen des Dienstes. Wurde der Dienst bereits gestartet, besteht nur die Möglichkeit zum "Stoppen". Ist der Dienst aktuell nicht gestartet, besteht nur die Möglichkeit zum "Starten". Diese Funktion ist analog zum Starten und Stoppen eines Dienstes über die FirstSpirit Server- und Projektkonfiguration (siehe Kapitel 7.3.11 Seite 254).

Neustart: Funktion zum Neustart des Dienstes. Anders als beim einfachen "Starten" oder "Stoppen" wird dabei ein bereits gestarteter Dienst zuerst gestoppt und anschließend automatisch neu gestartet.





Nach dem Update von Modulen, die Abhängigkeiten zu Modulen mit Diensten ("Service") haben, müssen diese Dienste manuell neugestartet werden.

8.6.2.5 Server Neustart

Server neustarten: Durch einen Klick auf die Schaltfläche wird der FirstSpirit-Server neu gestartet.



Es erfolgt keine Sicherheitsabfrage vor dem Server-Neustart. Alle Clients, die am Server angemeldet sind, sollten vor dem Neustart beendet werden, um Datenverluste zu vermeiden. Dazu kann der Wartungsmodus verwendet werden (siehe Kapitel 8.6.2.1 Seite 457).

8.6.2.6 AppCenter Lizenzen

Die Verwendung des FirstSpirit AppCenters unterliegt einem neuen Lizenzmodell. Im Unterschied zur bisherigen Lizenzierung einer FirstSpirit-(Modul)-Erweiterung, wird nicht die Funktionalität lizenziert, sondern die Anzahl der integrierten Applikationen (im JavaClient und im WebClient). Der Lizenzparameter `license.APPTAB_SLOTS` bestimmt, wie viele verschiedene Applikationsintegrationen verwendet werden können (siehe Kapitel 4.3.5 Seite 94).

Diese Ansicht zeigt die Art und Anzahl der Anwendungen, die über den Parameter `license.APPTAB_SLOTS` aktuell registriert sind.

Über die Schaltfläche **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.



8.6.2.7 Netzwerk

Auf dieser Seite werden alle FirstSpirit-Server aufgeführt, die sich innerhalb des aktuellen Netzwerks befinden. Zu jedem Server werden die FirstSpirit-Version und weitere Informationen eingeblendet, die sich auf Werte aus der Konfigurationsdatei `fs-server.conf` (Hostname, SocketPort, HttpPort des FirstSpirit Servers) und auf Lizenzeinstellungen des Servers (Lizenztyp) beziehen.

Server	HTTP-Port	Version	Typ	Socket-Port
[+] 4.2 Stable Release		4.2.454.47473	production	
[+] 4.2R4 Stable Release		4.2.454.47473	production	
[+] 5.0 SVN-Version		5.0_BETA.33.51813	production	
[+] FirstSpirit 4.2R4 (Produktion)		4.2.472.51994	production	
[+] e-Spirit Intranet (Produktiv)		5.0_BETA.34.52001	production	
Apache FOP		5.0_DEV.19_49938		
Apache FOP v0_20_5		5.0_DEV.19_50021		
ContentDependentMediaRelease		0.0.1_129		
Exalead-Push		2.0		
ExcelAddressImporter		1.0.20		
FIRSTpersonalisation		5.0_DEV.19_50179		
FirstSpirit FormEdit		5.0_DEV.19_49938		
FirstSpirit Geolocation Development		5.0_DEV.19_47664		
FirstSpirit Office		5.0_DEV.19_49938		
Intranet - FSDev		1.3_5.0_DEV.25_2564		
JSTL		1.1		
Perf4J		0.9.13		
PostgreSQL_JDBC_Driver_8_4		8.4.701		
SpellService		5.0_DEV.19_49938		
lastChanges		1.0-SNAPSHOT		
[+] fs42build (Intranet/Internet)		4.2.453.46978	production	

FirstSpirit-Server aktualisieren

Mit einem Klick auf das [+] kann ein Eintrag expandiert werden. Es werden dann Informationen zu den Modulen bereitgestellt, die auf dem jeweiligen FirstSpirit-Server installiert sind. Selbst erstellte Module werden **nicht** angezeigt.

Netzwerk aktualisieren Die angezeigten Serverinformationen können über diesen Button aktualisiert werden.

Diese und weitere Informationen sind auch über das JMX-Monitoring einsehbar und werden im Kapitel 9.6 Seite 485 genauer beschrieben.

FirstSpirit-Server aktualisieren In jedem expandierten Servereintrag wird der Button "FirstSpirit-Server aktualisieren" angezeigt. Über diese Funktion können



FirstSpirit-Server auf eine neue Software-Versionen aktualisiert werden (über das FirstSpirit-Servermonitoring). Beim Klick auf den Button öffnet sich die Server-Monitoring-Webapplikation des ausgewählten FirstSpirit-Servers. Nach erfolgreicher Authentifizierung des Serveradministrators wird der Bereich FirstSpirit Steuerung – Aktualisierung, zur Auswahl der fs-server.jar-Datei sowie der gewünschte Modul-Dateien, angezeigt (siehe Kapitel 8.6.2.3 Seite 460).

8.6.3 FirstSpirit – Nachricht

Auf dieser Seite kann ein Nachrichtentext editiert werden, der allen aktiven Clients (Startseite, Server- und Projektkonfiguration, Server-Monitoring, JavaClient und WebClient) am Server in Form eines PopUp-Fensters übermittelt wird.



Diese Menüebene steht nur Serveradministratoren zur Verfügung.

8.6.4 FirstSpirit – Datenbanken

FirstSpirit speichert die stark strukturierten Inhalte der Datenquellen-Verwaltung in einer Datenbank ab. Alle Informationen zur Anbindung von Datenbanken an den FirstSpirit Server befinden sich in Kapitel 4.8, ab Seite 142.

8.6.4.1 Übersicht

Übersicht über alle aktuell konfigurierten Datenbankanbindungen auf dem FirstSpirit Server.

Die Konfiguration einer Datenbankanbindung wird über die Konfigurationsdatei `fs-database.conf` vorgenommen. Eine Erläuterung zu den Parametern befindet sich in Kapitel 4.3.3, auf Seite 76. Änderungen an der Konfigurationsdatei `fs-database.conf` können über die FirstSpirit Server- und Projektkonfiguration vorgenommen werden (siehe Kapitel 7.3.5 Seite 241).



Die Parameter `PASSWORD` und `USER` werden nur Serveradministratoren angezeigt.



8.6.4.2 Status

Weitere Informationen zu einer angebotenen Datenbank zeigt der Menüeintrag "Status" an. Unterhalb der angezeigten Datenbankverbindung, werden die Projekte angezeigt, die auf diese Datenbank zugreifen.

Über den Verweis auf das Projekt unterhalb des Datenbank-Layers kann die Projekt-Detail-Seite aufgerufen werden (siehe Kapitel 8.2.1.1 Seite 434).

8.6.5 FirstSpirit – Monitoring

8.6.5.1 Speicher

Auf dieser Seite wird eine Grafik über den Speicherverbrauch der letzten 24 Stunden angezeigt. Die schwarze Linie zeigt den definierten -Xmx-Wert aus der Konfigurationsdatei `fs-wrapper.conf` (siehe Kapitel 4.3.2.3, Parameter `wrapper.java.maxmemory`) an. Hell-brombeer ist der Bereich, den die Java VM im Betriebssystem für den Java-Heap reserviert hat und dunkel-brombeer der tatsächlich verbrauchte Heap.

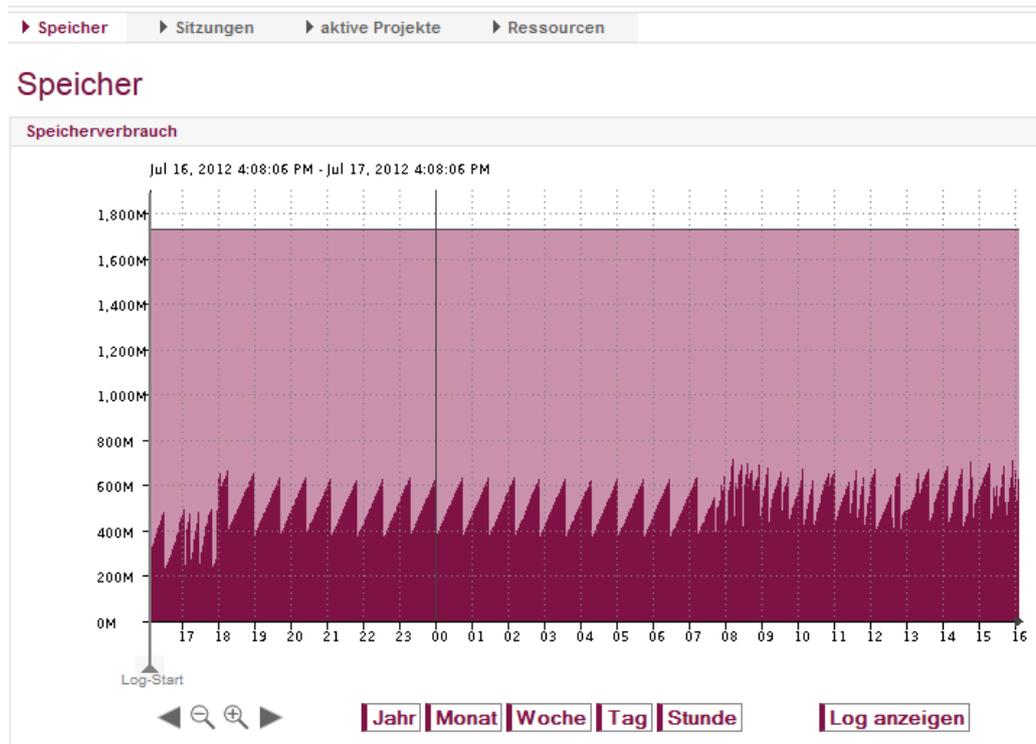


Abbildung 8-15: Monitoring – Speicherverbrauch



Der Zeitraum für die Anzeige des Speicherverbrauchs kann über die Schaltflächen **Jahr**, **Monat**, **Woche**, **Tag** und **Stunde** beliebig verändert werden. Ebenso kann über die Schaltfläche  der Zeitraum um eine Kategorie in Richtung "Jahr" verschoben werden und über die Schaltfläche  um eine Kategorie in Richtung "Stunde". Über die Pfeiltasten kann der angezeigte Zeitraum je nach eingestellter Kategorie um ein Jahr (einen Monat, eine Woche...) in die Vergangenheit  bzw. in die Zukunft  verlegt werden.

Durch einen Klick auf die Schaltfläche **Log anzeigen** wird die Log-Datei des Servers für den ausgewählten Zeitraum angezeigt. (Siehe Kapitel 8.3.1.2 ab Seite 443)

8.6.5.2 Sitzungen

Auf dieser Seite wird in einer Grafik die Anzahl der Sitzungen dargestellt, die in den letzten 24 Stunden gleichzeitig auf dem Server aktiv waren.

Der Zeitraum für die Anzeige der aktiven Sitzungen kann über die Schaltflächen **Jahr**, **Monat**, **Woche**, **Tag** und **Stunde** beliebig verändert werden. Ebenso kann über die Schaltfläche  der Zeitraum um eine Kategorie in Richtung "Jahr" verschoben werden und über die Schaltfläche  um eine Kategorie in Richtung "Stunde". Über die Pfeiltasten kann der angezeigte Zeitraum je nach eingestellter Kategorie um ein Jahr (einen Monat, eine Woche...) in die Vergangenheit  bzw. in die Zukunft  verlegt werden.

Durch einen Klick auf die Schaltfläche **Log anzeigen** wird die Log-Datei des Servers für den ausgewählten Zeitraum angezeigt. (Siehe Kapitel 8.3.1.2 ab Seite 443).



8.6.5.3 Aktive Projekte

Auf dieser Seite wird in einer Grafik die Anzahl der Projekte dargestellt, auf die in den letzten 24 Stunden gleichzeitig zugegriffen wurde.

Der Zeitraum für die Anzeige der aktiven Sitzungen kann über die Schaltflächen **Jahr**, **Monat**, **Woche**, **Tag** und **Stunde** beliebig verändert werden. Ebenso kann über die Schaltfläche  der Zeitraum um eine Kategorie in Richtung "Jahr" verschoben werden und über die Schaltfläche  um eine Kategorie in Richtung "Stunde". Über die Pfeiltasten kann der angezeigte Zeitraum je nach eingestellter Kategorie um ein Jahr (einen Monat, eine Woche...) in die Vergangenheit  bzw. in die Zukunft  verlegt werden.

Durch einen Klick auf die Schaltfläche **Log anzeigen** wird die Log-Datei des Servers für den ausgewählten Zeitraum angezeigt. (Siehe Kapitel 8.3.1.2 ab Seite 443).

8.6.5.4 Ressourcen

Auf dieser Seite befinden sich Informationen

- zur Speicherkapazität (Benutzer Speicher, Allozierter Speicher)
- zum Plattenplatz (Benutzer Festplattenspeicher, Freier Festplattenspeicher)
- sowie zur Größe des Backup- und des Export-Verzeichnisses.



Diese Informationen werden nicht automatisch beim Start des FirstSpirit-Servers ermittelt, sondern sind erst nach der Statistikaktualisierung (bis zum nächsten Serverstart) verfügbar. Der Wert für die Statistikaktualisierung wird über die globalen Servereigenschaften konfiguriert (vgl. "Tägliche Statistik (Uhrzeit)" in Kapitel 7.3.1 Seite 231) (Standardeinstellung 0 Uhr).

In der Tabelle werden alle auf dem Server installierten Projekte aufgelistet. Neben dem Projektnamen gibt es Angaben über die ID des Projektes und den benötigten Plattenplatz. Weiterhin wird angezeigt, wie viel Plattenplatz für das Projekt maximal zur Verfügung steht (Quota) (Ist keine Begrenzung eingetragen wird der Wert -1 angezeigt). Diese Liste kann nach allen vorhandenen Spalten sortiert werden.



8.6.5.5 Threads



Diese Menüebene steht nur Serveradministratoren zur Verfügung.

Auf dieser Seite befinden sich Informationen zur Überwachung des aktuellen Systemzustands.

Thread Dump: Mit einem Klick auf den Button wird ein aktueller Thread Dump erstellt und angezeigt. Mit einem Klick auf das [+] kann die angezeigte Ebene expandiert werden. Über die Buttons im oberen Fensterbereich kann der aktuelle Thread Dump analysiert und in einer aufbereiteten Ansicht angezeigt werden.

[+]	"AppletIOServlet.sessionTimer" id=47 in TIMED_WAITING block count = 1 block time = -1	10
[+]	"BerkeleyDbBackend-0" id=33 in TIMED_WAITING cpu time = 109200700 block count = 23 block time = -1	8
[+]	"PoolThread[DEFAULT]-1" id=24 in WAITING cpu time = 312002000 block count = 20 block time = -1	4
[+]	"Attach Listener" id=5 in RUNNABLE block count = 0 block time = -1	3
[+]	"Checkpointner" id=32 in WAITING block count = 134 block time = -1	2
[+]	"LogWriter.PLAIN/Logging" id=19 in WAITING block count = 0 block time = -1	2

Abbildung 8-16: Threads – Sortierte Ansicht nach "Trace"

Trace: Ein Klick auf die Schaltfläche gruppiert die aktuellen Threads nach ähnlichen Stacktraces. Stacktraces werden zu einer Gruppe zusammengefasst, wenn sie bis auf die Objektadressen identisch sind. Die Zahl im rechten Bereich zeigt die Anzahl der gefundenen ähnlichen Traces an. Durch die aufbereitete Ansicht kann relativ einfach festgestellt werden, welche Aktionen aktuell auf dem Server stattfinden.

Topline: Ein Klick auf die Schaltfläche gruppiert die aktuellen Threads nach Ähnlichkeiten innerhalb der ersten fünf Zeilen. Ähnliche Threads werden in der aufbereiteten Anzeige zu einem Eintrag zusammengefasst. Die Zahl im rechten Bereich zeigt die Anzahl der gefundenen ähnlichen Threads an. Durch die aufbereitete Ansicht kann relativ einfach festgestellt werden, an welcher Stelle der Ausführung sich die Threads aktuell befinden. Diese Ansicht erleichtert das Aufspüren von Engpässen (Bottlenecks), beispielsweise wenn viele Threads immer



innerhalb einer bestimmten Codestelle warten.

Lock: Ein Klick auf diese Schaltfläche untersucht die aktuellen Threads nach Locks. Befindet sich mehrere Threads im Zustand BLOCKED (siehe unten "Status"), warten diese auf die Freigabe eines Objekts von einem anderen Thread. Befinden sich ständig viele Objekte in einem Wartestatus, bei einer gleichzeitigen geringen Verarbeitungsgeschwindigkeit, kann das ein Indiz für Bottlenecks sein.

Text: Ein Klick auf den Button zeigt den vollständigen unbearbeiteten Thread Dump mit allen Informationen an.

Neben der einzelnen Erstellung eines Thread Dumps können weitere Dumps in bestimmten Zeitintervallen erstellt werden. Dabei können die insgesamt gewünschte Anzahl an Thread Dumps im ersten Feld und die Zeitintervalle im zweiten Eingabefeld eingegeben werden. Ein Klick auf den Button "Erstellen" startet dann die Ausführung.

3 Thread Dumps im Abstand von 3 Minuten **Erstellen**

Tabelle: Die tabellarische Übersicht der Seite zeigt den Zustand der aktuellen Threads. Die hier abgebildeten Thread-Informationen stellen eine Momentaufnahme bzw. ein Ausschnitt des aktuellen Zustands dar:

ID: jeder Thread besitzt eine Thread-ID über die er eindeutig identifizierbar ist.

Name: Name des Threads.

Status: Die Threads können sich in unterschiedlichen Zuständen befinden:

- NEW – neu erzeugter Thread, der noch nicht gestartet wurde.
- RUNNABLE – der Thread wird gerade abgearbeitet oder ist ablaufbereit (wartet auf CPU).
- BLOCKED – der Thread wartet auf die Freigabe eines Locks.
- WAITING – der Thread wartet auf einen anderen Thread, um weiterlaufen zu können.
- TIMED_WAITING – der Thread wartet auf den Ablauf einer festgelegten Zeit (z. B. nach Aufruf von sleep() oder wait() mit Timeout)
- TERMINATED – der Thread ist beendet

IN (In Native): Gibt Auskunft darüber, ob der Thread nativen Code über das Java Native Interface (JNI) ausführt (true) oder nicht (false).



SP (Suspended): Gibt Auskunft darüber, ob ein Thread aktuell gestartet ist (false) oder nicht (true).

BC (Blocked Count): Die Zahl gibt Auskunft darüber, wie oft sich ein Thread während der Ausführung bereits im Status BLOCKED befand.

WC (Waited Count): Die Zahl gibt Auskunft darüber, wie oft sich ein Thread während der Ausführung im bereits Status WAITING befand.

Ein Klick auf einen Thread innerhalb der Tabelle öffnet ein Detailfenster zu diesem Thread. Dort werden alle tabellarischen Informationen übersichtlich aufgelistet und zusätzlich das Lockobjekt und der "Owner" angezeigt:

ID	3
Name	Finalizer
Status	WAITING
Lock	java.lang.ref.ReferenceQueue\$Lock@a7d346
Owner	-
In Native	false
Suspended	false
Blocked Count	240
Waited Count	32

Abbildung 8-17: Threads – Detailinformation

Lock: Befindet sich ein Thread im Status BLOCKED wartet er auf die Freigabe des hier angegeben Lock-Objekts.

Owner: Befindet sich ein Thread im Status BLOCKED wird der Name des Threads ausgegeben, der aktuell das Lock auf das Objekt hält.

Die weiteren Ausgaben werden innerhalb des aktuellen Kapitels beschrieben (s.o.).



8.6.5.6 VM Speicher

In diesem Bereich werden Informationen zur aktuellen Speichernutzung der Java VM angezeigt.



Diese Menüebene steht nur Serveradministratoren zur Verfügung.

Name: Bezeichnung des abgebildeten Speicherbereichs (Memory Pool).

- **Code Cache:** Speicherbereich, der für die interne Auswertung beispielsweise für das Kompilieren verwendet wird.
- **Eden Space:** Speicherbereich, in dem die meisten Objekte initial erzeugt werden. Sobald der Garbage Collector (GC) den Eden Space bereinigt, werden die überlebenden Objekte in den Survivor Space überführt.
- **Survivor Space:** Speicherbereich, in den die kurzlebigen Objekte aus dem Eden Space nach dem GC (Garbage Collection) verschoben werden.
- **Tenured Gen:** Speicherbereich für langlebige Objekte, die nach einiger Zeit aus dem Survivor Space hierher verschoben werden.
- **Perm Gen:** Speicherbereich für permanent benötigte Objekte der JVM.
- **Total:** Kein Speicherbereich. Liefert einen Gesamtüberblick über alle verfügbaren Speicherbereiche.

Typ: Art des Speichers (HEAP || NON_HEAP).

Max: Maximal möglicher Speicher (in bytes), der für das Speichermanagement verwendet worden ist.

Genutzt: Aktuell genutzter Speicher (in bytes).

Initial: Initial, beim Start der JVM, zugewiesener Speicher (in bytes).

Verpflichtet: Garantiert zur Verfügung gestellter Speicher (in bytes) für die JVM.

Neben der tabellarischen Übersicht im oberen Fensterbereich, steht für jeden Speicherbereich eine graphische Darstellung zur Verfügung.



8.6.5.7 Kennzahlen

Die Java Management Extensions (JMX) stellen eine einheitliche Schnittstelle für das Management von Java Anwendungen zur Verfügung. Mithilfe der JMX-Konsole ist es möglich, den FirstSpirit-Server zur Laufzeit zu überwachen und zu verwalten (siehe Kapitel 9 Seite 476).

Einige Informationen stehen als grafische Darstellung auch über den Eintrag Kennzahlen innerhalb des Server-Monitorings zur Verfügung.



Diese Menüebene steht nur Serveradministratoren zur Verfügung.

TasksWaiting: Anzahl der Aktionen, die bereits aus der entsprechenden ExecutionQueue zur Ausführung in den internen ThreadPool weitergeleitet wurden, aber noch warten (vgl. Abbildung 9-6) (siehe Kapitel 9.8.3 Seite 490).

TasksRunning: Anzahl der momentan laufenden Aktionen (beispielsweise ein Indizierungsjob oder ein Event) (siehe Kapitel 9.8.3 Seite 490).

TasksQueued: Wird intern eine Aktion angestoßen, beispielsweise eine Indizierung, dann wird der entsprechende Task zuerst in eine Warteschlange gestellt (Ausnahme: hochpriorisierte Tasks) (vgl. Abbildung 9-6). Der hier dargestellte Wert gibt die Anzahl der momentan zwischengespeicherten Aktionen wieder (siehe Kapitel 9.8.3 Seite 490).

ExecutionRate: Anzahl der ausgeführten Tasks, innerhalb der letzten 60 Sekunden (siehe Kapitel 9.8.3 Seite 490).

Average Preview Creation Count: Anzahl der Seiten bzw. Seitenreferenzen, für die innerhalb der letzten 60 Sekunden eine neue Vorschau berechnet wurde (siehe Kapitel 9.13 Seite 495).

Average Session Creation Rate: Anzahl der neu geöffneten Sitzungen innerhalb der letzten 60 Sekunden (siehe Kapitel 9.22.1 Seite 505).



8.6.6 FirstSpirit – Clustering

Über den Menüeintrag "Clustering" kann eine Übersicht über die bestehenden Clusterknoten für die Generierung aufgerufen werden. Initial ist die Übersicht leer (falls nur ein FirstSpirit Master-Server vorhanden ist). Clusterknoten werden erst angezeigt, wenn diese im Clusterverbund vorhanden sind (siehe Kapitel 7.3.14 Seite 275).

Es kann eine beliebige Anzahl an "Generation Slaves" für die verteilte Bearbeitung der Generierungsaufträge dargestellt werden.

Für jeden Generierungs-Server werden folgende Informationen abgebildet:

Version: Versionsnummer des FirstSpirit-Servers

Server Typ: Hier wird der Server-Typ der Clusterknoten angegeben. Die Übersicht zeigt beispielsweise den Typ "SLAVE (Generation Slave)" an, der für die Veröffentlichungsvorgängen zuständig ist. Natürlich können auch mehrere Generierungs-Server zum Einsatz kommen. Bei Bedarf können so mehrere Veröffentlichungen auf verschiedene Server verteilt werden.

Letzter Kontakt: Zeitpunkt des letzten Server-Kontakts (ping).

Aktueller Zustand: Aktueller Zustand des Servers:

- IDLE Leerlauf
- BUSY Der Server ist momentan mit der Abarbeitung von Prozessen beschäftigt.

Auslastung: Zeigt die prozentuale Auslastung des Servers an, beispielsweise bei der Bearbeitung eines Generierungsauftrags.



9 FirstSpirit JMX-Konsole

Die Java Management Extensions (JMX) stellen eine einheitliche Schnittstelle für das Management von Java Anwendungen zur Verfügung. Mithilfe der JMX-Konsole ist es möglich, den FirstSpirit-Server zur Laufzeit zu überwachen und zu verwalten. Während das FirstSpirit Server-Monitoring primär auf die manuelle Überwachung eines FirstSpirit-Servers abzielt, dient die JMX-Schnittstelle zur automatischen Überwachung und integriert sich optimal in ein ggf. bereits vorhandenes, unternehmensweites Monitoring. Grundsätzlich kann die JMX-Konsole auch interaktiv verwendet werden.

Im Vergleich zum Server-Monitoring (siehe Kapitel 8 Seite 429) sind die Informationen, die über die JMX-Konsole zur Verfügung gestellt werden, deutlich feingranularer. JMX bietet einen standardisierten Weg, Ressourcen einer Anwendung oder der darunter liegenden JVM anzusehen und zu verwalten. Alle Werte und Operationen werden dabei über so genannte Managed Beans (MBeans) zur Verfügung gestellt. Einige MBeans und die von ihnen gelieferten Informationen werden in den folgenden Kapiteln (ab Kapitel 9.3) vorgestellt.



Die Struktur und Funktion der nachfolgend erläuterten MBeans ist kontinuierlichen Änderungen unterworfen. Diesem ständigen Aktualisierungsprozess kann die Dokumentation nur in begrenztem Umfang gerecht werden. So ist es möglich, dass aufgrund geringer Zeitverzögerungen Abbildung oder Beschreibungen in dieser Dokumentation von der aktuellen Ansicht der JMX-Konsole abweichen.



9.1 Start der JMX-Konsole

Um die JMX-Konsole effektiv zu nutzen, muss eine Remote-Verbindung zur zu überwachenden Anwendung bestehen. Ab JDK 1.5 ermöglicht eine plattformeigene Funktion das Starten eines JMX-Agenten. Um diese Funktion für den FirstSpirit-Server zu aktivieren, muss die Konfigurationsdatei `fs-server.conf` angepasst werden (Konfiguration siehe Kapitel 4.3.1.18 Seite 63).

In Verbindung mit der Anwendung wird die JConsole über den Kommandozeilenaufruf "jconsole" gestartet:



Abbildung 9-1: Starten der JConsole (Remote)

Der Remote-Zugriff wird über die Option "Remote Process" gestartet. Hier kann neben dem Hostname auch der JMX-Port der Zielanwendung eingetragen werden. Wenn die entsprechenden Parameter konfiguriert wurden (empfohlen für Produktivsysteme), muss außerdem eine Authentifizierung über "User Name" und "Passwort" erfolgen. Mit einem Klick auf "Connect" wird die JConsole gestartet. (Die Darstellung der JMX-Konsole ist abhängig von der verwendeten JConsole.)

Eine umfangreiche Beschreibung der GUI befindet sich unter:

<http://docs.oracle.com/javase/6/docs/technotes/guides/management/jconsole.html>



9.2 MBeans

Alle Werte und Operationen für das JMX-Monitoring des FirstSpirit-Servers werden über Managed Beans (MBeans) zur Verfügung gestellt. Diese befinden sich unter der Registerkarte "MBeans" der JConsole (siehe Abbildung 9-2).

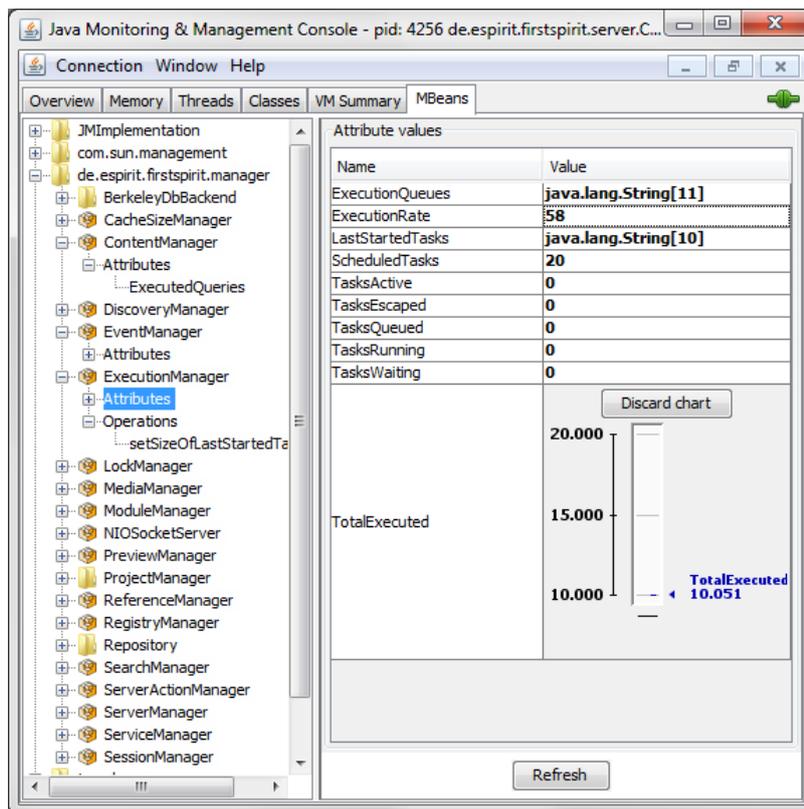


Abbildung 9-2: Registerkarte "MBean" der JMX-Konsole

Die MBeans stellen neben Informationen und Statistikwerten ("Attributes") auch Bearbeitungsmöglichkeiten ("Operations") bereit:

Attributes: Informationen und Statistikwerte, die sowohl tabellarisch (Standardanzeige) als auch graphisch (Umschaltung mit einem Doppelklick auf den entsprechenden Wert) dargestellt werden können. Mit einem Klick auf den Button "Discard chart" wird die graphische Darstellung wieder geschlossen.



Attribute values	
Name	Value
ActiveCalculated	0
ActiveDuration	432ms
ActiveQueued	1
ActiveRecalculations	3
ActiveRepaired	0
CalculationRate	1694

Abbildung 9-3: Information und Statistikwerte über die JMX-Konsole (Attribute Values)

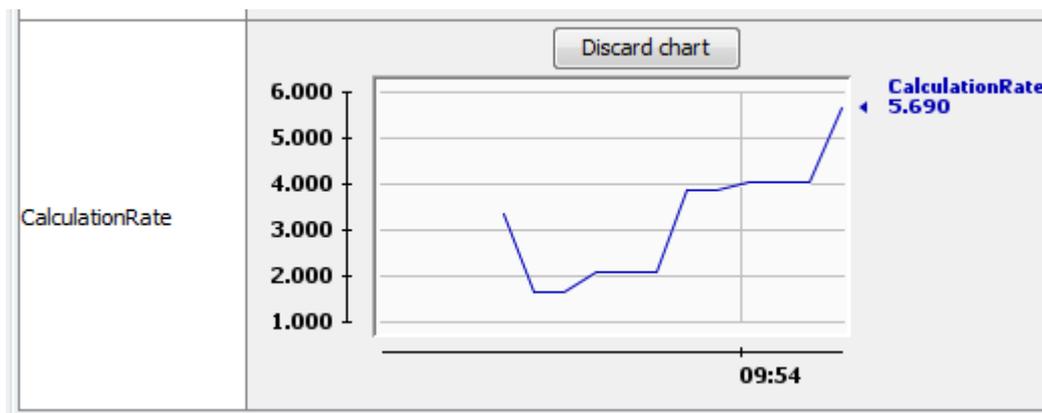


Abbildung 9-4: Statistikwerte über die JMX-Konsole (Attribute Values - Graph)

Operations: Operationen, die auf bestimmten Managern aufgerufen werden können, beispielsweise um weitere Statistikwerte anzuzeigen oder um bestimmte Aktionen anzustoßen (siehe Abbildung 9-5). Die Operationen werden mit einem Klick auf den entsprechenden Button gestartet.



Das Ausführen von Operationen über die JMX-Konsole im laufenden Betrieb kann zu Problemen führen (z.B. beim Neustart eines Managers) und sollte immer nur von erfahrenen FirstSpirit-Administratoren durchgeführt werden.

Die Operationen werden unmittelbar (ohne vorherige Sicherheitsabfrage) ausgeführt.



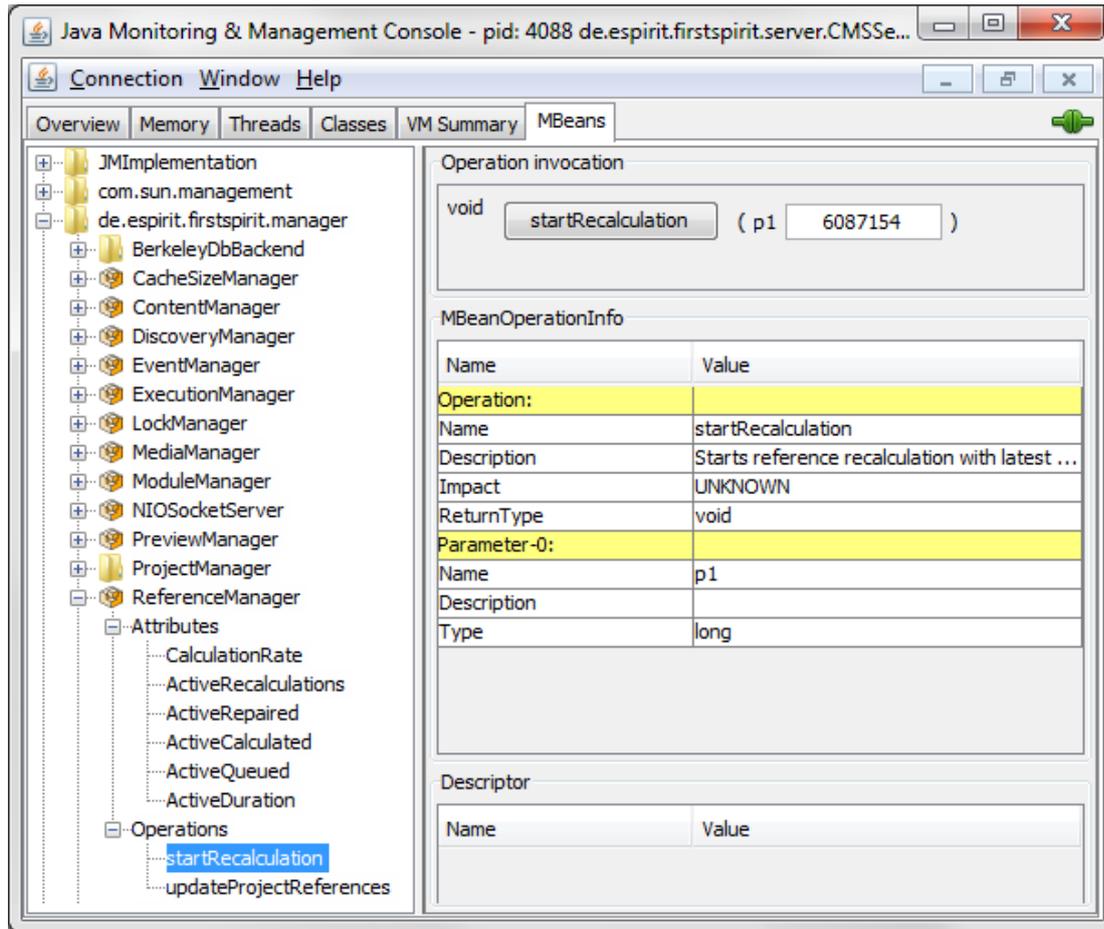


Abbildung 9-5: Referenzneuberechnung über die JMX-Konsole ausführen (Operations)

Info: Allgemeine Informationen zum Manager (Name des Managers und Klasse, die den Manager implementiert).

Die entsprechenden Informationen oder Operationen werden nur in Managern angezeigt, in denen diese auch relevant sind (Beispiele siehe Kapitel 9.3.1 Seite 481 und Kapitel 9.3.2 Seite 482).

Einige MBeans und die von ihnen gelieferten Informationen werden in den folgenden Kapiteln (ab Kapitel 9.3 ff.) vorgestellt.



9.3 BerkeleyDB-Backend

Unter diesem Eintrag werden Informationen zu den angebundenen Backendsystemen dargestellt. Momentan wird als Backendsystem nur BerkeleyDB unterstützt, daher werden in der JMX-Konsole ausschließlich Informationen über das Berkeley-Storage-Repository der verwendeten Projekte angezeigt. Diese Anzeige ist, ebenso wie der RepositoryManager (siehe Kapitel 9.17) oder CacheSizeManager (siehe Kapitel 9.4) dynamisch. Das heißt, der Eintrag wird erst dann in der JMX-Konsole geladen, wenn das entsprechende Projekt von mindestens einem Benutzer geöffnet wurde. Die Informationen werden dann dynamisch zur Laufzeit aktualisiert. Dabei besitzt jedes Projekt ein eigenes MBean, das unterhalb des Ordners "BerkeleyDBBackend" angezeigt wird.

9.3.1 Attribute

`backupMode`: Die BerkeleyDB verwendet interne Cleaning-Mechanismen, um nicht mehr verwendete Daten in bestimmten Intervallen aus der Datenbank zu entfernen. Über den Backup-Mode können bei einem Backup die Cleaner der BerkeleyDB ausgeschaltet werden (Wert: `true`) (Standardwert: `false`).

`batchCleanerRunning`: Neben dem BerkeleyDB-Cleaning kann ein interner Cleaner verwendet werden, der den Cleaning-Mechanismus der BerkeleyDB explizit aufruft.

`cachePercent`: Gibt den Speicherverbrauch für das Backend als prozentualer Wert in Bezug auf den Hauptspeicher an.

`cacheSize`: Gibt den Speicherverbrauch für das Backend in Byte an.

`cleanedLogs`: Anzahl der bisher weggeräumte Log-Dateien (Datenbank-Dateien der BerkeleyDB).

`cleanerBacklog`: Anzahl (Queue) der noch nicht weggeräumten Log-Dateien.

`cleanerThreads`: Anzahl der BerkeleyDB internen Threads, die für das Cleaning zuständig sind. Die Anzahl kann wechseln und ist abhängig vom Wert der `cleanerBacklogs`.

`environmentHome`: Pfad zum Verzeichnis in dem die BerkeleyDB-Daten gespeichert sind (Repository).



`isReadOnly`: Information, ob das Backend schreibgeschützt ist (Wert `true`) oder nicht (Wert `false`).

`isTransactional`: Information, ob das Backend transaktional geöffnet ist (Wert `true`) oder nicht (Wert `false`).

`lockTimeout`: Timeout der Berkeley interne Locks in μ s.

`openCursors`: Anzahl der Datenbank-Cursors, die beim Iterieren über die Datenbank geöffnet werden.

`runCleaner`: Information, ob das BerkeleyDB-Cleaning an- (Wert `true`) oder ausgeschaltet ist (Wert `false`).

9.3.2 Operation

Die hier angezeigten **Operationen** können mit einem Klick auf die entsprechende Schaltfläche, beispielsweise auf `cleanLog()` ausgeführt werden:

`java.lang.Integer cleanLog()`: Beim Ausführen dieser Methode werden nicht mehr benötigte Log-Dateien (Datenbank-Dateien der BerkeleyDB) weggeräumt (Rückgabewert: Anzahl der durch diese Funktion weggeräumten Logs).

`void evictMemory()`: Beim Ausführen dieser Methode werden die Caches der BerkeleyDB freigegeben.

`void sync()`: Beim Ausführen dieser Methode werden alle geänderten Daten, die noch nicht gespeichert wurden auf die Festplatte geschrieben.

Neben den Operationen stehen eine Reihe von Statistiken zur Verfügung, die ebenfalls mit einem Klick auf das entsprechende Feld, beispielsweise auf `getLockStats(...)` ausgeführt werden können:

`void resetStats()`: Beim Ausführen dieser Methode werden alle Statistik-Informationen der BerkeleyDB zurückgesetzt.

`java.lang.String getEnvironmentStats(boolean)`: Beim Ausführen dieser Methode werden allgemeine Statistkinformationen über die BDB angezeigt.

`java.lang.String getLockStats(boolean)`: Beim Ausführen dieser Methode werden die Lock-Statistiken der BerkeleyDB angezeigt.



`java.lang.String getKeyDatabaseStats(boolean)`: FirstSpirit-Daten werden in unterschiedlichen Datenbanken gespeichert (Key-, Value- und Blob-DBs). Beim Ausführen dieser Methode werden die Statistiken der Key-BerkeleyDB angezeigt.

`java.lang.String getValueDatabaseStats(boolean)`: FirstSpirit-Daten werden in unterschiedlichen Datenbanken gespeichert (Key-, Value- und Blob-DBs). Beim Ausführen dieser Methode werden die Statistiken der Value-BerkeleyDB angezeigt.

`java.lang.String getBlobDatabaseStats(boolean)`: FirstSpirit-Daten werden in unterschiedlichen Datenbanken gespeichert (Key-, Value- und Blob-DBs). Beim Ausführen dieser Methode werden die Statistiken der Blob-BerkeleyDB angezeigt.



Der Aufruf kann besonders bei vielen Daten einen Moment Zeit in Anspruch nehmen.

9.4 CacheSizeManager

Der CacheSizeManager teilt den zur Verfügung stehenden Hauptspeicher auf die verschiedenen Zwischenspeicher (Caches) auf. Unter diesem Eintrag werden Informationen zur Gesamt-Cache-Größe und Informationen zu den projektspezifischen-Caches ("SharedCaches") dargestellt. Der CacheSizeManager ist, ebenso wie der RepositoryManager (siehe Kapitel 9.17) ein dynamischer Manager. Das heißt, der Eintrag wird erst dann in der JMX-Konsole geladen, wenn das entsprechende Projekt von mindestens einem Benutzer geöffnet wurde. Die Informationen werden dann dynamisch zur Laufzeit aktualisiert ("Refresh").

9.4.1 Attribute (gesamt)

`CacheCount`: Gesamtanzahl der momentan angemeldeten Caches (aller Projekte).

`CacheSize`: Absolute Größe des Caches. Der hier abgebildete Wert ergibt sich aus den Werten, die für die Parameter `CACHE_SIZE` (absolute Angabe) oder `CACHE_PERCENT` (prozentuale Angabe) in der Konfigurationsdatei `fs-server.conf` (siehe Kapitel 4.3.1.12 Seite 55) gesetzt wurden.

`LastCalculationDate`: Zeitpunkt, zu dem der Speicher zwischen den Caches zuletzt neu verteilt wurde.



`UsedCacheSize`: Gibt die Cachegröße an, die momentan von allen Caches tatsächlich genutzt wird.

9.4.2 Operation

`void distributeCacheMemory()`: Beim Ausführen dieser Methode wird der zur Verfügung stehende Speicher neu verteilt. Nach der Ausführung wird der unter "LastCalculationDate" gespeicherte Wert angepasst (vgl. 9.4.1).

9.4.3 Attribute (projektbezogen)

Neben den Gesamtinformationen für die `CacheSize` besitzen die Projekte eigene `CacheSizeMBeans`, die unterhalb des `CacheSizeMangers` unter dem MBean `SharedCaches` zusammengefasst werden.

`MissCount`: Der Zähler erfasst die Anzahl an Anfragen aller Projekte auf dem Server, die nicht direkt aus dem Cache bedient werden konnten. Diese Anfragen müssen an das Backend weitergeleitet und über einen Zugriff auf die Festplatte bedient werden. Ein hoher oder stark ansteigender Wert ist ein Indikator für eine schlechte Cache-Nutzung und damit für lange Antwortzeiten.

`PreferredSize`: Hier wird die Speichergröße abgebildet, die der Projekt-Cache bevorzugen würde. Diese Speichergröße entspricht nicht immer dem real zugeteilten Wert ("Size"), der unter anderem abhängig von der "PreferredSize" verteilt wird.

`Size`: Standardmäßig wird jedem Projekt-Cache vom `CacheSizeManager` die Größe von einem MB zugeteilt. Ist die "Preferred Size" höher, kann eventuell mehr Speicher zur Verfügung gestellt werden. Wieviel Speicher ein Projekt-Cache erhält, ist abhängig von der zur Verfügung stehenden Gesamtspeichergröße (Globale `CacheSize`), der "PreferredSize" aller Projekte und der Gewichtung für die einzelnen Projekte.

`Weight`: Die Gewichtung ist ein Parameter, der über die Server- und Projektkonfiguration über die Projekteigenschaften definiert wird (siehe Kapitel 7.4.2). Über die Gewichtung eines Projekts erhält ein Projekt-Cache mehr (bei hoher Gewichtung) oder weniger (bei niedriger Gewichtung) Speicher vom `CacheSizeManager` zugeteilt.



9.5 Content Manager

Der ContentManager enthält Informationen zu FirstSpirit-Datenbankverbindungen und -zugriffen.

Attribute:

`ExecutedQueries`: Anzahl der ausgeführten Datenbank-SQL-Statements innerhalb der letzten Minute.

9.6 DiscoveryManager

Der DiscoveryManger ermittelt über das Senden und Beantworten von Broadcast-Nachrichten weitere Server innerhalb des aktuellen Netzwerks. Unter diesem Eintrag werden Informationen zu diesen Servern gesammelt und bereitgestellt. Diese Informationen können auch über das ServerMonitoring im Bereich "FirstSpirit – Steuerung – Netzwerk" eingesehen werden (siehe Kapitel 8.6.2.7 Seite 465).

9.6.1 Attribute

`NewestServer`: Zeitstempel des neuesten, bekannten Servers.

`OldestServer`: Zeitstempel des ältesten, bekannten Servers.

`ServerCount`: Gesamtanzahl der Server in diesem Netzwerk.

9.6.2 Operations

`void updateServers()`: Beim Aufruf dieser Methode werden die Serverinformationen aktualisiert. Diese Aktion kann auch über das FirstSpirit-ServerMonitoring ausgeführt werden (siehe Kapitel 8.6.2.7).

9.6.3 Attribute (serverbezogen)

`Hostname`: Hostname des FirstSpirit-Servers. Dieser Wert wird über den Parameter `HOST` in der Konfigurationsdatei `fs-server.conf` definiert (vgl. Kapitel 4.3.1.1).

`HttpPort`: Http-Port des FirstSpirit-Servers. Dieser Wert wird über den Parameter `HTTP_PORT` in der Konfigurationsdatei `fs-server.conf` definiert (vgl.



Kapitel 4.3.1.1).

`InetAddress`: Adressinformationen des FirstSpirit-Servers.

`LicenseId`: FirstSpirit Lizenz-Id (Informationen zur FirstSpirit-Lizenz siehe Kapitel 8.6.1.2).

`LicenseType`: FirstSpirit Lizenz-Typ (Informationen zur FirstSpirit-Lizenz siehe Kapitel 8.6.1.2).

`Modules`: Modulinformationen des FirstSpirit-Servers.

`SocketPort`: Socket-Port des FirstSpirit-Servers. Dieser Wert wird über den Parameter `SOCKET_PORT` in der Konfigurationsdatei `fs-server.conf` definiert (vgl. Kapitel 4.3.1.1).

`Timestamp`: Zeitstempel der angezeigten Serverinformation.

`Url`: URL zur Startseite des FirstSpirit-Servers. Dieser Wert wird über den Parameter `URL` in der Konfigurationsdatei `fs-server.conf` definiert (vgl. Kapitel 4.3.1.1).

`User`: User auf den die FirstSpirit-Lizenz ausgestellt wurde.

`Version`: FirstSpirit-Server-Version.

9.7 Event Manager

Der EventManager stellt Informationen über gefeuerte und gesendete Events bereit. Viele Aktionen, die innerhalb der FirstSpirit Umgebung angestoßen werden, führen zu bestimmten Events. Beispielsweise führt das Laden eines Projekts dazu, dass die entsprechenden Referenzen über den ReferenceManager geladen werden. Der EventManager sorgt damit für eine Koppelung unterschiedlicher Manager (z. B. ReferenceManager und SearchManager).

Weitere Aktionen die Events feuern sind:

- Elemente anlegen, bearbeiten, speichern und löschen
- Projekt laden, ändern, deaktivieren und löschen
- Datenbankinhalte anlegen, bearbeiten und löschen

Über den EventManager lassen sich damit Rückschlüsse auf die Verarbeitungsgeschwindigkeit und Last auf dem FirstSpirit-Server, insbesondere in



Multi-User-Umgebungen, ziehen.

Attribute:

`EventListeners`: Anzahl der server- und clientseitigen EventListener, die zur Zeit auf dem Server registriert sind.

`EventQueues`: Die EventQueues sind das serverseitige Gegenstück zum clientseitigen EventListener. Da der Server die Events nicht direkt an den Client stellen kann, werden diese Events "zwischengelagert", bis sie abgeholt werden (siehe auch `EventsQueued`). Indirekt gibt dieser Wert Auskunft darüber, wie viele Clients mit dem Server verbunden sind.

`EventsFired`: Gefeuerte Events der letzten 60 Sekunden.

`EventsQueued`: Die Anzahl der Events die gefeuert und noch nicht abgearbeitet wurden (siehe auch `EventQueues`) (innerhalb der letzten 60 Sekunden).

`EventsSent`: Anzahl der tatsächlich (an einen EventListener) gesendeten Events der letzten 60 Sekunden.

`LastFiredEvents`: Zeigt die letzten, gefeuerten Events (String-Array).

9.8 ExecutionManager

In FirstSpirit wird eine Vielzahl von Aktionen parallel im Hintergrund ausgeführt. Dazu gehört beispielsweise das Aktualisieren des Refrenzgraphs, das Abarbeiten von Client-Calls oder auch das Indizieren von Dokumenten. Die Ausführung dieser Aktionen erfolgt über den ExecutionManager.

9.8.1 Klassifizierung der Thread Queues

Der ExecutionManager verwaltet eine Vielzahl von unterschiedlich klassifizierten Thread-Queues. In einer Thread-Queue werden die unterschiedlichen Tasks (beispielsweise Client-Calls) bis zu ihrer Ausführung gesammelt. Thread-Queues können abhängig von ihrer Klassifikation konfiguriert werden:

`ThreadQueue.SERIAL`: In die mit `SERIAL` klassifizierten Queues werden Tasks gestellt, die nicht parallel ausgeführt werden können. Da immer nur ein Task gleichzeitig ausgeführt wird, ist eine Begrenzung der Queue



über die Konfiguration hier nicht notwendig.

`ThreadQueue.LOW`: In die mit `LOW` klassifizierten Queues werden speicher- oder rechenintensive Tasks gestellt, von denen jeweils nur eine geringe Anzahl gleichzeitig ausgeführt werden soll (Standardwert 2). Ein Beispiel hierfür ist das Indizieren von Dokumenten. Die Anzahl der Tasks, die gleichzeitig ausgeführt werden dürfen, kann über den Parameter `maxRunning` definiert werden (siehe Kapitel 4.3.1.5 Seite 40).

`ThreadQueue.DEFAULT`: `DEFAULT` ist die Standard-Klassifikation einer Queue. In eine mit `DEFAULT` klassifizierte Queue werden alle Tasks gestellt, die nicht einer der anderen klassifizierten Queues zugeordnet sind. Die Anzahl der Tasks, die gleichzeitig ausgeführt werden dürfen, kann über den Parameter `maxRunning` definiert werden (siehe Kapitel 4.3.1.5 Seite 40).

`ThreadQueue.HIGH`: Die `HIGH` klassifizierte Queue dient allen Tasks, die direkt ausgeführt werden sollen. Dabei handelt es sich beispielsweise um hochpriorisierte Client-Calls (z.B. ping). Auch hier ist eine Konfiguration nicht notwendig.

`ThreadQueue.BOUNDED`: In die mit `BOUNDED` klassifizierte Queue werden ausschließlich Server Calls der Clients gestellt. Diese Queue kann zweifach begrenzt werden. Es kann sowohl die Anzahl der aktiven Tasks, als auch die Aufnahmekapazität der Queue eingeschränkt werden (zur Konfiguration siehe Kapitel 4.3.1.5 Seite 40). Ist die maximale Aufnahmekapazität der Queue erreicht, werden weitere Client-Calls vom Server kurzfristig zurückgewiesen, bevor ein weiterer Versuch gestartet wird (Rejection-Strategie).



9.8.2 Verarbeitung innerhalb des ExecutionManagers

Das folgende Schaubild verdeutlicht die Ausführung der Tasks im ExecutionManager. Die hier beschriebenen Werte stehen innerhalb der JMX-Konsole (siehe nächste Seite) und teilweise innerhalb des FirstSpirit Server-Monitorings (siehe Kapitel 8 Seite 429) zur Verfügung:

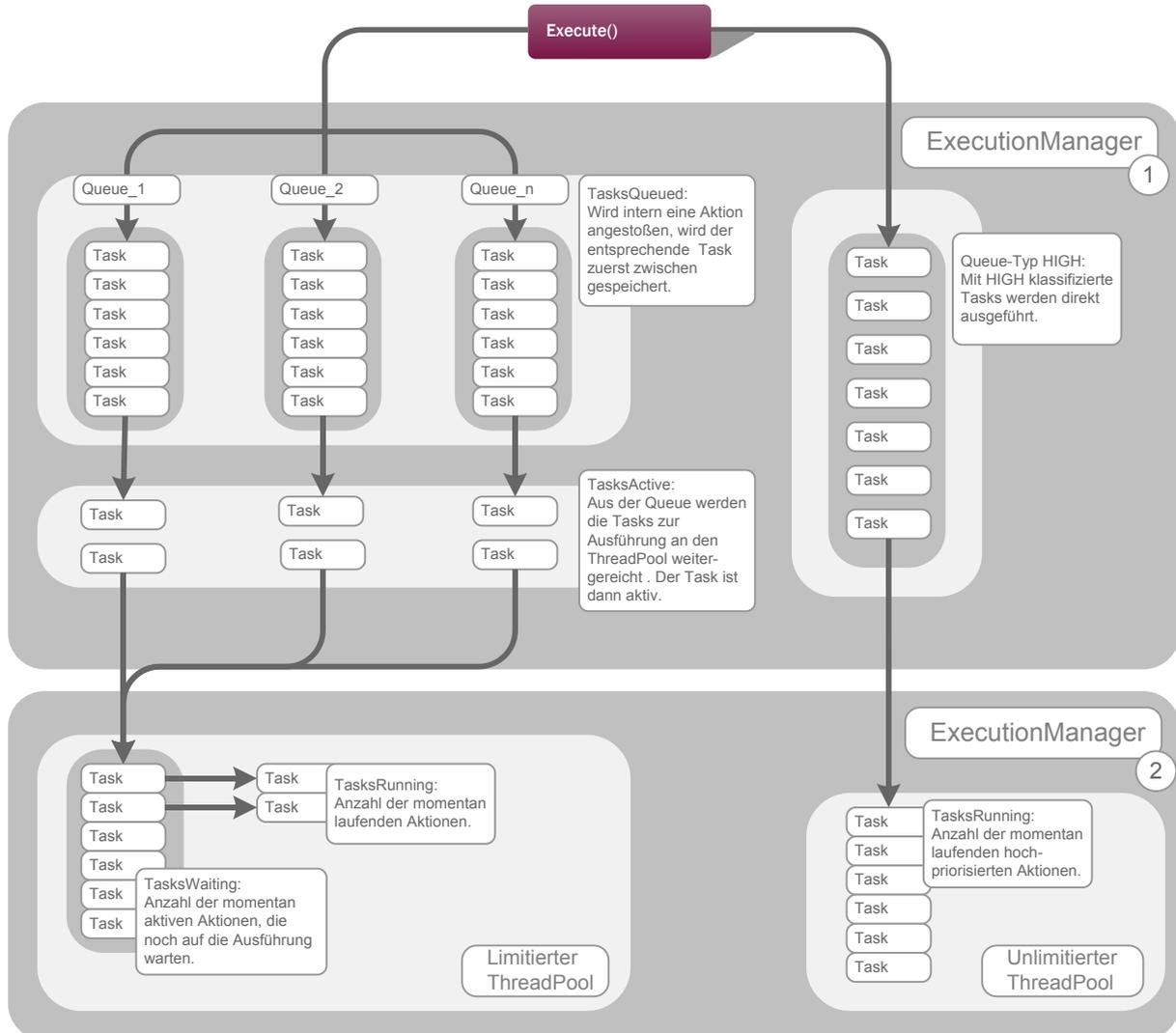


Abbildung 9-6: Ausführung der Tasks über den ExecutionManager



9.8.3 Attribute

ExecutionRate: Anzahl der ausgeführten Tasks, innerhalb der letzten 60 Sekunden.

LastStartedTasks: Zeigt die letzten, gestarteten Tasks (String-Array).

ScheduledTasks: Zusätzlich zu den durch Benutzeraktionen ausgelösten Events, gibt es regelmäßige, interne Aktionen. Beispielsweise wird einmal pro Minute geprüft, ob eine Session weiterhin gültig oder bereits abgelaufen ist. Der Wert gibt die Anzahl aller regelmäßig anstehenden Aktionen wieder.

TasksActive: Eine zwischengespeicherte, interne Aktion (vgl. "TasksQueued") wird aus der Queue weitergereicht an die zweite Verarbeitungsebene des ExecutionManagers (vgl. Abbildung 9-6). Der entsprechende Task ist dann aktiv, wird aber noch nicht ausgeführt, sondern wandert in den (limitierten) ThreadPool. Die Anzahl der möglichen aktiven Tasks, die aus der Queue in den ThreadPool gelangen, kann über den Parameter `maxRunning` begrenzt werden (siehe Kapitel 4.3.1.5 Seite 40).

TasksQueued: Wird intern eine Aktion angestoßen, beispielsweise eine Indizierung, dann wird der entsprechende Task zuerst zwischengespeichert. Der hier dargestellte Wert gibt die Anzahl der momentan zwischengespeicherten Aktionen wieder (vgl. Abbildung 9-6).

TasksEscaped: Gibt die Anzahl der Tasks an, die aufgrund einer vollen Queue außerhalb ihrer eigentlichen Queue ausgeführt werden.

Hintergrund: Die Queues des ExecutionManagers arbeiten mit einer Rejection-Strategie: ist eine Queue "bounded", also in ihrer Größe beschränkt, werden darüber hinaus abgesetzte Tasks an die Rejection-Strategie übergeben. Der aufrufende Thread stellt seinen Task in die Queue und führt dafür den ältesten Eintrag selbst aus. Solche Tasks heißen "escaped", da sie nicht über ihre eigentliche Queue ausgeführt werden. Die "überlastete" Queue bedient sich sozusagen der aufrufenden Threads und die "aktiven" Tasks der Queue übersteigen "maxRunning". Die Queue-Kapazität kann innerhalb der Konfigurationsdatei `fs-server.conf` definiert werden (siehe Kapitel 4.3.1.5 Seite 40).



TasksRunning: Anzahl der momentan laufenden Aktionen (beispielsweise ein Indizierungsjob oder ein Event) innerhalb des limitierten und unlimitierten ThreadPools (vgl. Abbildung 9-6). Die Anzahl der parallel ausgeführten Tasks (über den limitierten ThreadPool) kann über den Parameter `maxSize` begrenzt werden (siehe Kapitel 4.3.1.4 Seite 39).

TasksWaiting: Anzahl der Aktionen, die bereits aus der entsprechenden Queue weitergeleitet wurden, aber noch auf die Ausführung innerhalb des ThreadPools warten (vgl. Abbildung 9-6).

TotalExecuted: Die Gesamtanzahl der ausgeführten Aktionen.

ExecutionQueues: Der Eintrag `ExecutionQueues` stellt eine Übersicht zu allen Warteschlangen (Queues) der unterschiedlichen Manager zur Verfügung. Mit einem Doppelklick auf den Eintrag oder über die Navigation im linken Fensterbereich können die aktuellen Warteschlangen angezeigt werden (siehe Abbildung 9-7).

```
Queue[2:SearchManager.dataIndexing,DEFAULT] - a:0 q:0 m:24 c:104
Queue[1:SearchManager.mediaIndexing,LOW] - a:0 q:0 m:2 c:130
Queue[3:SearchManager.searching,DEFAULT] - a:0 q:0 m:24 c:104
Queue[4:MediaManager,LOW] - a:0 q:0 m:2 c:130
```

Abbildung 9-7: ExecutionManager – Queues

Jeder Eintrag beinhaltet die folgenden Informationen:

Name der Queue: Der Name setzt sich zusammen aus (siehe Abbildung 9-7):

- der ID der Queue
- dem Namen des Managers, der die Queue verwendet
- der Projekt-ID. (Dieser Wert ist optional. Manche Manager, beispielsweise der `ReferenceManager`, erzeugen pro Projekt eine Queue)
- der Queue-Klasse. Mögliche Klassen sind `LOW`, `HIGH SERIAL`, `BOUNDED`, `DEFAULT` (siehe Kapitel 9.8.1 Seite 487). Einige Queue-Klassen können über die Konfigurationsdatei `fs-server.conf` parametrisiert werden (siehe Kapitel 4.3.1.5, ab Seite 40).

Anzahl der aktiven Threads: Anzahl der aktiven Threads [`a: Anzahl`], die aus der Queue an den ThreadPool weitergereicht werden (vgl. "TasksActive").



Anzahl der zwischengespeicherten Tasks: Anzahl der zwischengespeicherten Threads [`q: Anzahl`], die innerhalb der Queue auf ihre Ausführung warten (vgl. "TasksQueued").

Anzahl der parallel ausführbaren Tasks: Anzahl der Tasks, die gleichzeitig ausgeführt werden dürfen, [`m: Anzahl`] (vgl. "maxRunning").

Anzahl der erlaubten zwischengespeicherten Tasks: Anzahl der Tasks [`c: Anzahl`], die innerhalb einer "Bounded"-Queue zwischengespeichert werden dürfen (vgl. "queueCapacity" in Kapitel 4.3.1.5 Seite 40)

Anzahl der aktuell ausgeführten Tasks: Anzahl der momentan laufenden Tasks [`r: Anzahl`] (beispielsweise ein Indizierungsjob oder ein Event) innerhalb des limitierten und unlimitierten ThreadPools (vgl. "TasksRunning").

Anzahl der wartenden Threads: Anzahl der zwischengespeicherten Threads [`w: Anzahl`], die aus der Queue an den ThreadPool weitergereicht, aber noch nicht ausgeführt wurden (vgl. "TasksWaiting").

9.9 LockManager

Innerhalb von FirstSpirit müssen Objekte bei bestimmten Aktionen, beispielsweise dem Ändern der Projekteigenschaften oder dem Bearbeiten einer Seite, vor dem Zugriff durch andere Benutzer gesperrt werden. Dazu setzt der jeweilige Benutzer automatisch (beim Ändern der Projekteigenschaften) oder manuell (beim Bearbeiten einer Seite) ein Lock auf diesem Objekt. Der LockManager verwaltet alle momentan gesperrten Objekte und stellt Informationen zu verschiedenen Locktypen zur Verfügung.

Attribute:

`LockedObjects`: Anzahl aller Objekte (Projekte, Elemente, Pakete, Aufträge), die sich momentan in Bearbeitung befinden und durch einen Benutzer gesperrt wurden.

`ObjectsLocked`: Anzahl aller Objekte (Projekte, Elemente, Pakete, Aufträge), die innerhalb der letzten 60 Sekunden durch einen Benutzer gesperrt wurden.

`LockedPackages`: Anzahl aller Pakete, die sich momentan in Bearbeitung befinden und durch einen Benutzer gesperrt wurden. Dieser Wert wird nur gefüllt, wenn die lizenzabhängige Funktionalität "CorporateContent"



verwendet wird.

`LockedProjects`: Anzahl aller Projekte, die sich momentan in Bearbeitung befinden und durch einen Benutzer gesperrt wurden (Beispiel siehe Kapitel 7.3.13 Seite 269).

`LockedScheduleEntries`: Anzahl aller Aufträge, die sich momentan in Bearbeitung befinden und durch einen Benutzer gesperrt wurden (Beispiel siehe Kapitel 7.5.4 Seite 360).

`LockedStoreElements`: Anzahl aller Elemente aus FirstSpirit Verwaltungsbereichen (z. B. Inhalte-Verwaltung), die sich momentan in Bearbeitung befinden und durch einen Benutzer gesperrt wurden.

`LockedDatasets`: Anzahl der Entities (Datenbank-Entities), die sich momentan in Bearbeitung befinden und durch einen Benutzer gesperrt wurden.

9.10 MediaManager

FirstSpirit stellt innerhalb der Medien-Verwaltung verkleinerte Vorschaubilder (Thumbnails) zur Verfügung. Der MediaManager verwaltet die Informationen zur Darstellung und Berechnung dieser Thumbnails beim Anlegen eines neuen Mediums und ist außerdem zuständig für die Berechnung der Auflösungen, wenn diese angefordert werden.

Attribut:

`ImagesScaled`: Anzahl der bisher skalierten Medien (dazu zählen beispielsweise auch alle neu berechneten Vorschaubilder (Thumbnails)).

9.11 ModuleManager

FirstSpirit unterstützt die Integration von Fremdkomponenten. Zentrale Einheit dieser Integration ist das Modul, welches wiederum aus einer oder mehreren Komponenten besteht. Jede Komponente nutzt Ressourcen (shared, web-local – z.B. Klassen, Jar-Archiv, Properties-Dateien), hat eine bestimmte Sichtbarkeit (scope), besitzt einen Typ (z.B. Projektanwendung, Webanwendung) und bestimmte Eigenschaften. Alle Dateien eines Moduls werden in einer gezippten Datei mit der Endung `.fsm` zusammengefasst und können anschließend auf dem FirstSpirit-Server installiert werden. Der ModuleManager ist zuständig für alle allgemeinen Funktionen zum Verwalten von Modulen und Komponenten auf dem FirstSpirit-Server. Das MBean `ModuleManager` liefert eine Liste aller auf dem Server geladener Module (`fsm-`



Dateien).

Informationen zur Entwicklung, zum Classloading und zur Installation von Module siehe FirstSpirit Modul-Entwicklerhandbuch für Komponenten.

9.12 NIOSocketServer

Die Kommunikation zwischen FirstSpirit-Servern und -Clients läuft über den NIOSocketServer. Das zugehörige MBean zeigt die aktuellen Messwerte des Managers.

Attribute:

Host: Socket-Host, der Hostname für die Bindeadresse des Socket_Ports, um den FirstSpirit-Server, wenn notwendig, auf eine IP-Adresse zu beschränken. Ist kein Wert hinterlegt, bindet der Server an alle IP-Adresse des Hosts. Dieser Wert wird über den Parameter `SOCKET_HOST` in der Konfigurationsdatei `fs-server.conf` definiert (vgl. Kapitel 4.3.1.1).

Port: SocketPort des FirstSpirit-Servers. Dieser Wert wird über den Parameter `SOCKET_PORT` in der Konfigurationsdatei `fs-server.conf` definiert (vgl. Kapitel 4.3.1.1).

ConnectionCount: Anzahl der aktuell offenen Sockets für die Kommunikation zwischen FirstSpirit-Server und den FirstSpirit-Clients.

ConnectionCreationRate: Anzahl der neu geöffneten Socket-Verbindungen innerhalb der letzten 60 Sekunden.

ConnectionGroupCount: Anzahl der aktuell zum Server verbundenen FirstSpirit-Clients (JavaClient, WebClient, Webapplikationen). (entspricht nicht der Anzahl der offenen Sockets).

DanglingCallCount: Der FirstSpirit-Server arbeitet alle Client-Requests nach folgender Strategie ab. Kurze Anfragen werden direkt vom FirstSpirit-Server beantwortet. Bei Anfragen, deren Beantwortung mehr als 30 Sekunden in Anspruch nimmt, wird der Server-Response nicht mehr direkt zurückgesendet, sondern vom Server abgearbeitet und anschließend zur Abholung durch den Client bereitgestellt. Das Attribut `DanglingCallCount` gibt die Anzahl der Anfragen zurück, die vom Server bereits bearbeitet und zur Abholung bereitgestellt wurden.



`LongDurationCallCount`: Anzahl der langandauernden Anfragen (>30 Sek.), die vom FirstSpirit-Server noch nicht komplett abgearbeitet und daher noch nicht zur Abholung durch den Client bereitgestellt wurden.

`ResultWorkerCount`: Anzahl der aktuell bearbeiteten Client-Anfragen.

9.13 PreviewManager

Über FirstSpirit kann jederzeit eine Vorschau der aktuell bearbeiteten Inhalte angefordert werden. Das Anfordern einer Vorschau ist innerhalb der Inhalte-Verwaltung auf Seitenebene oder innerhalb der Struktur-Verwaltung auf Seitenreferenzebene möglich. Der PreviewManager verwaltet alle momentan angeforderten Vorschauen und stellt Informationen dazu bereit.

Attribute:

`AveragePreviewCount`: Anzahl der Seiten bzw. Seitenreferenzen, für die innerhalb der letzten 60 Sekunden eine neue Vorschau berechnet wurde.

`RunningPreviews`: Anzahl der aktuell zur Berechnung angeforderten Vorschauen.



Das Anfordern einer Vorschau ist eine sehr rechenzeitintensive Aktion. FirstSpirit bietet daher Möglichkeiten, das Berechnen einer Vorschau auf vorgelagerte Systeme zu verschieben. In diesem Fall kann der PreviewManager keine Informationen zur Vorschau auswerten.



9.14 ProjectManager

Unter diesem Eintrag werden vorrangig Informationen über die genaue die Anzahl der Objekte (unterteilt nach Objekttypen und Verwaltungsbereichen) eines Projektes bereitgestellt. Dazu besitzt jedes Projekt ein eigenes MBean, das unterhalb des Ordners "ProjectManager" angezeigt wird. Diese Informationen werden nicht dynamisch aktualisiert. Die Berechnung wird erst beim Aufruf der Methode `calculateCounters()` gestartet. Diese aktualisiert die gewünschten Informationen im Bereich Attribute und setzt zugleich den Wert für das Attribut `_Status` auf `CALCULATED`.

9.14.1 Attribute

`All`: Anzahl aller Objekte des Projekts.

`All_ContentStore`: Anzahl aller Objekte in der Datenquellen-Verwaltung des Projekts.

`All_MediaStore`: Anzahl aller Objekte in der Medien-Verwaltung des Projekts.

`All_PageStore`: Anzahl aller Objekte in der Inhalte-Verwaltung des Projekts.

`All_SiteStore`: Anzahl aller Objekte in der Struktur-Verwaltung des Projekts.

`All_TemplateStore`: Anzahl aller Objekte in der Vorlagen-Verwaltung des Projekts.

`ContentStoreDefinitions`: Anzahl der Datenquellen in der Datenquellen-Verwaltung des Projektes.

`ContentStoreFolders`: Anzahl der Ordner in der Datenquellen-Verwaltung des Projektes.

`MediaStoreFolders`: Anzahl der Ordner in der Medien-Verwaltung des Projektes.

`MediaStoreMedia`: Anzahl der Medien (Dateien, Bilder) in der Medien-Verwaltung des Projektes.

`PageStoreBodyNodes`: Anzahl der Inhaltsbereiche in der Inhalte-Verwaltung des Projektes.

`PageStoreSections`: Anzahl der Absätze in der Inhalte-Verwaltung des Projektes.



`PageStoreFolders`: Anzahl der Ordner in der Inhalte-Verwaltung des Projektes.

`PageStorePages`: Anzahl der Seiten in der Inhalte-Verwaltung des Projektes.

`SiteStoreDocumentGroups`: Anzahl der Dokumentengruppen in der Struktur-Verwaltung des Projektes.

`SiteStoreFolders`: Anzahl der Menüebenen in der Struktur-Verwaltung des Projektes.

`SiteStorePageReferences`: Anzahl der Seitenreferenzen in der Struktur-Verwaltung des Projektes.

`TemplateStoreFormatTemplates`: Anzahl der Formatvorlagen in der Vorlagen-Verwaltung des Projektes.

`TemplateStoreLinkTemplates`: Anzahl der Verweisvorlagen in der Vorlagen-Verwaltung des Projektes.

`TemplateStorePageTemplates`: Anzahl der Seitenvorlagen in der Vorlagen-Verwaltung des Projektes.

`TemplateStoreTableTemplates`: Anzahl der Tabellenvorlagen in der Vorlagen-Verwaltung des Projektes.

`TemplateStoreWorkflows`: Anzahl der Arbeitsabläufe in der Vorlagen-Verwaltung des Projektes.

`_Status`: Status der Berechnung `NOT_CALCULATED` | `CALCULATED`. Die Berechnung wird beim Ausführen der Methode `calculateCounters()` gestartet und setzt dann den Wert `CALCULATED` und beim Ausführen der Methode `resetCounters()` zurückgesetzt auf den Wert `NOT_CALCULATED`.

9.14.2 Operations

`void resetProjectStores()`: Beim Ausführen der Methode werden die gecachten, serverseitigen `AccessAPI`-Stores zurückgesetzt. Das kann beispielsweise sinnvoll sein, wenn Auflösungen oder Sprachen im Projekt hinzugefügt oder gelöscht wurden und diese Änderungen im Projekt noch nicht sichtbar sind. Hinweis: Diese Methode sollte nicht im laufenden Serverbetrieb ausgeführt werden!



`void resetCounters()`: Beim Ausführen der Methode wird die Anzahl der zuvor berechneten Attributwerte auf 0 zurückgesetzt. Der Wert des Attributs `_STATUS` ändert sich nach dem Ausführen auf `NOT_CALCULATED`.

`void calculateCounters()`: Die Anzahl der Projekt-Objekte, die als Attributwerte in den einzelnen MBeans bereitgestellt wird, muss zuvor berechnet werden. Dazu muss die Methode `calculateCounters()` ausgeführt werden. Der Wert des Attributs `_STATUS` ändert sich nach dem Ausführen auf `CALCULATED`.

9.15 ReferenceManager

Wesentliche Funktionalitäten von FirstSpirit basieren auf dem Referenzgraph eines Projekts. Der Referenzgraph wird verwendet, um die Abhängigkeiten von Objekten innerhalb komplexer Projekte zu erkennen. Der ReferenceManager ist für das Berechnen und Ausliefern dieser Referenzen zuständig. Die Referenzberechnung erfolgt normalerweise automatisiert, während der laufenden Projektarbeit. In einzelnen Fällen kann aber eine manuelle Neuberechnung der Referenzen erforderlich sein. Die Referenz-Neuberechnung verläuft asynchron, d.h. Server und Projekte können bereits verwendet werden, während die Referenzberechnung noch läuft. Das wird jedoch nicht empfohlen, da einige Operationen, die den Referenzgraph auswerten, möglicherweise unvollständige Informationen zugrundelegen. So können beispielsweise verwendete Objekte in einem Projekt gelöscht werden, wenn die eingehenden Referenzen noch nicht vollständig berechnet wurden.

9.15.1 Attribute

`ActiveCalculated`: Aktuelle Anzahl aller bereits neu berechneten Projektreferenzen auf dem FirstSpirit-Server.

`ActiveDuration`: Aktuelle Dauer der zurzeit laufenden Neuberechnung von Projektreferenzen auf dem FirstSpirit-Server. Diese Information wird auch im ServerMonitoring im Bereich "Übersicht – Aktivitäten" eingeblendet (siehe Kapitel 8.1.2 Seite 431).

`ActiveQueued`: Aktuelle Anzahl der noch zu berechnenden Projektreferenzen auf dem FirstSpirit-Server.



`ActiveRecalculations`: Aktuelle Anzahl der Projektreferenzen, die in diesem Moment neu berechnet werden.

`ActiveRepaired`: Aktuelle Anzahl der reparierten Projektreferenzen auf dem FirstSpirit-Server. Diese Information wird auch im ServerMonitoring im Bereich "Übersicht – Aktivitäten" eingeblendet (siehe Kapitel 8.1.2 Seite 431).

`CalculationRate`: Anzahl der Referenzen, die innerhalb der letzten 60 Sekunden neu berechnet wurden (Referenzen/Minute).

9.15.2 Operations

`void updateProjectReferences()`: Diese Methode startet die Neuberechnung aller veralteten Projektreferenzen auf dem FirstSpirit-Server.

`void startRecalculation(projectID)`: Diese Methode startet die Neuberechnung der Projektreferenzen, in der jeweils aktuellsten, vorliegenden Revision, für ein bestimmtes Projekt. Über die JConsole kann hier die Projekt-ID des Projektes übergeben werden, für die eine Neuberechnung der Referenzen gestartet werden soll.

Die Neuberechnung der Projektreferenzen kann im ServerMonitoring im Bereich "Übersicht – Aktivitäten" verfolgt werden (siehe Kapitel 8.1.2 Seite 431).

9.16 RegistryManager

Ab FirstSpirit 5.0 werden zentrale Registries verwendet, um bestimmte interne Projektdaten, beispielsweise vordefinierte URLs oder auch die sogenannten GIDs (Global Identifier), die beim FirstSpirit-ContentTransport benötigt werden, zu speichern (in einer NoSQL-Datenbank). Jedes Projekt kann mehrere Registries besitzen. Die Registries werden vom RegistryManager verwaltet.

Operations:

`void deleteRegistry (projectID, registry name)`: Diese Methode löscht die Registry eines Projekts mit allen enthaltenen Daten. Dazu muss sowohl die Projekt-ID als auch der Name der Datenbank angegeben werden.



9.17 RepositoryManager

FirstSpirit setzt zur Archivierung und Versionierung der Projektdaten Content-Repositories ein. Unter einem Content-Repository versteht man eine zentrale Stelle, an der die vom Content Management System benötigten Datenstrukturen (Medien, Seiten, Vorlagen usw.) verwaltet werden. Jedes FirstSpirit Projekt besitzt ein eigenes in sich abgeschlossenes Content-Repository und für jedes Repository existiert ein MBean im Ordner "Repository" der JMX-Konsole. Diese Anzeige ist dynamisch. Das heißt, der Eintrag wird erst dann in der JMX-Konsole geladen, wenn das entsprechende Projekt von mindestens einem Benutzer geöffnet wurde. Die Informationen werden dann dynamisch zur Laufzeit aktualisiert.

BackendClass: Klasse, die das entsprechende Repository implementiert (momentan wird nur BerkeleyDB unterstützt).

BasePath: Vollständiger Pfad zum Repository des Projekts.

LastClosedRevision: Nummer der letzten Revision, die zu diesem Projekt freigegeben wurde.

RevisionCounter: Gesamtanzahl der Revisionen für dieses Projekt.

UncommittedRevisions: Gesamtanzahl der Revisionen, die gerade in Bearbeitung, (das heißt noch nicht freigegeben) sind.

9.18 SearchManager

Der SearchManager ist zuständig für das Indizieren und Suchen von Elementen aus den Verwaltungsbereichen von FirstSpirit (z. B. Seite, Absätze, Mediendateien). Die Elemente werden beim Anlegen, Bearbeiten oder Löschen in den Suchindex geschrieben bzw. dort wieder gelöscht.

Die Volltext-Indizierung großer Datenmengen (beispielsweise viele, umfangreiche Dokumente aus der FirstSpirit-Medien-Verwaltung) sind zeit- und rechenintensiv. Die Menge der gleichzeitig indizierten Dokumente, können global über die Einstellungen innerhalb der Konfigurationsdatei `fs-server.conf` geregelt werden:

- `ThreadQueue.DEFAULT.maxRunning:` für Inhalte und Variablen von `StoreElement`-Objekten.
- `ThreadQueue.LOW.maxRunning:` für Dokumente aus der Medien-Verwaltung, (z.B. PDF-Dateien).



9.18.1 Attribute:

`IndexingStarted`: Anzahl der Elemente, die innerhalb der letzten 60 Sekunden angelegt, verändert oder gelöscht wurden und über den `SearchManager` indiziert werden müssen. Diese Aktionen werden zwischengespeichert und mit niedriger Priorität abgearbeitet.

`DocumentsIndexed`: Anzahl der Elemente, die innerhalb der letzten 60 Sekunden neu indiziert wurden (Indizierungsvorgang ist abgeschlossen).

`SearchStepsExecuted`: Jede Suche besteht aus einer Menge unterschiedlicher Suchschritte (pro Suche 3-5 Suchschritte). Der hier angegebene Wert enthält die Anzahl der Suchschritte, die innerhalb der letzten 60 Sekunden ausgeführt wurden.



Volltext-Indizierung der Medien-Verwaltung: Dokumente aus der FirstSpirit Medien-Verwaltung werden ebenfalls indiziert, um eine Volltextsuche innerhalb der Dokumente zu ermöglichen. Aus technischen Gründen werden dabei nicht alle Dokumenttypen unterstützt. Eine (vollständige) Indizierung ist daher nicht in allen Fällen möglich.

9.18.2 Operations

`void pause()`: Diese Methode unterbricht die Indizierung von Projekthinhalten. Die Indizierung kann dann zu einem späteren Zeitpunkt durch Ausführen der Methode `resume()` wieder fortgesetzt werden. Beide Methoden können beispielsweise eingesetzt werden, um die Indizierung während des Imports großer Datenmengen in ein FirstSpirit-Projekt kurzzeitig auszusetzen.

`void resume()`: Diese Methode setzt die Indizierung von Projekthinhalten fort, nachdem diese zuvor angehalten wurde (siehe `pause()`).

`void rebuildIndex(projectID)`: Diese Methode startet die manuelle Aktualisierung des Suchindexes für ein Projekt. Dazu muss die entsprechende Projekt-ID übergeben werden. Beim Ausführen der Methode erfolgt kein kompletter Neuaufbau des Suchindexes. Der bestehende Index wird lediglich ergänzt. Ist der Suchindex defekt, sollte daher zunächst die Methode `clearIndex(projectID)` ausgeführt werden.



`boolean clearIndex(projectID)`: Mithilfe dieser Methode kann der Suchindex eines Projektes gelöscht werden, (z.B. weil dieser defekt ist). Anschließend wird automatisch ein neuer leerer Index für das Projekt angelegt. Um das Projekt neu zu indizieren, sollte nachfolgend die Methode `rebuildIndex(projectID)` ausgeführt werden.

9.19 ServerActionManager

Viele rechenintensive Aktionen, wie die Ermittlung der Abhängigkeiten von Objekten während eines Freigabeprozesses, werden von den FirstSpirit-Clients auf den Server verlagert. Der ServerActionManager verwaltet diese ServerActions und sorgt auch dafür das die Aktionen nach ihrer Bearbeitung zyklisch weggeräumt werden.

9.19.1 Attribute

`ActionCount`: Anzahl aller ServerActions die aktuell auf dem FirstSpirit-Server ausgeführt werden. Abhängig von der jeweiligen Implementierung, kann es vorkommen, dass nach dem Ausführen einer ServerAction das zugehörige ServerActionHandle nicht entfernt wird. In diesem Fall erhöhen die liegengebliebenen ServerActionHandles ebenfalls den ActionCount und werden erst nach einer Wartezeit (Standardwert 5 min) zyklisch entfernt.

`Actions`: Zeigt die alle Aktionen des FirstSpirit-Servers (String-Array).

9.19.2 Operations

`java.lang.String getResult(actionID)`: Die Methode liefert das Ergebnis nach Beenden der ServerAction zurück, z.B. `result=true` bei Freigabe-ServerActions. Das ist möglich, wenn die ServerAction bereits beendet wurde, das zugehörige ServerActionHandle aber noch existiert. Für FirstSpirit-interne ServerActions wird das ServerActionHandle automatisch entfernt, nachdem die Aktion beendet wurde.

`java.lang.String cancelAction (actionID)`: Beim Ausführen dieser Methode wird die ServerAction mit der übergebenen actionID abgebrochen. Das Ergebnis der Aktion wird zurückgeliefert.

`boolean isActionRunning(actionID)`: Die Methode liefert zurück, ob die ServerAction mit der übergebenen actionID, aktuell noch auf dem Server ausgeführt wird oder nicht.



9.20 ServerManager

Der ServerManager verwaltet allgemeine Daten und Methoden für die Steuerung und Konfiguration des FirstSpirit-Servers.

9.20.1 Attribute

`IgnoreStopServerRequests`: Zeigt ob die Steuerung des FirstSpirit-Servers über die Methode `ignoreStopServerRequests(pwd)` deaktiviert wurde (`true`) oder nicht (`false`).

`Version`: Zeigt die Version des FirstSpirit-Servers nach dem Schema: `major.minor.build.revision`

9.20.2 Operations

`java.lang.String dumpThreads()`: Beim Ausführen der Methode wird ein ThreadDump des Servers erzeugt und zurückgeliefert.

`boolean logThreadDump()`: Beim Ausführen der Methode wird ein ThreadDump des Servers erzeugt und mit dem Loglevel INFO in die Server-Log-Datei (`fs-server.log`) geschrieben.

`boolean enableVerboseMemoryLogging()`: Die Methode leitet alle Logausgaben des GarbageCollectors (zur Speicherauslastung der Java VM) (siehe Kapitel 4.3.2.1) auf die Konsole (Kommandozeile/Shell) um, von der FirstSpirit gestartet wurde.

`boolean disableVerboseMemoryLogging()`: Die Methode deaktiviert das Umleiten der Logausgaben des GarbageCollectors auf die Konsole wieder (siehe `enableVerboseMemoryLogging()`) und stellt den Zustand wieder her, der ursprünglich für die Logausgaben definiert wurde (siehe Kapitel 4.3.2.4 Seite 73).

`void clearLoggerCache()`: Die Methode setzt den Cache für die Log-Instanzen zurück.

`void restartManager(manager)` Die Methode startet einen Manager neu. Dazu muss der Name des gewünschten Managers übergeben werden.

`void ignoreStopServerRequests(pwd)`: Die Methode verhindert das Stoppen



des Servers über die herkömmlichen Serversteuerung (vgl. Kapitel 3) und wurde für die interne Testinfrastruktur eingeführt.

`void resetIgnoreStopServerRequests(pwd)` Die Methode stellt den herkömmlichen Zustand zur Steuerung des FirstSpirit-Servers wieder her (vgl. `ignoreStopServerRequests(pwd)`).

9.21 ServiceManager

Der ServiceManager verwaltet und steuert die FirstSpirit Systemdienste. Ein Dienst ist eine Serverkomponente, die über eine öffentliche Schnittstelle aus Eingabekomponenten oder Skripten angesprochen werden kann. (Beispiele sind die Rechtschreibprüfung oder der Dienst der Rechte-Eingabekomponente `CMS_INPUT_PERMISSION`). Die Konfiguration und die Steuerung von Diensten kann auch über das FirstSpirit-ServerMonitoring (Konfiguration siehe Kapitel 8.6.1.7 und Steuerung siehe Kapitel 8.6.2.4) oder die FirstSpirit Server- und Projektkonfiguration erfolgen (siehe Kapitel 7.3.11 Seite 254).

9.21.1 Attribute

`Services`: Zeigt die alle Dienste des FirstSpirit-Servers (String-Array).

9.21.2 Operations

`java.lang.String stopService(serviceName)`: Stoppt den Dienst mit dem übergebenen Namen.

`java.lang.String startService(serviceName)`: Startet den Dienst mit dem übergebenen Namen.

`java.lang.String restartService(serviceName)`: Neustart des Dienstes mit dem übergebenen Namen. Anders als beim einfachen "Starten" oder "Stoppen" wird dabei ein bereits gestarteter Dienst zuerst gestoppt und anschließend automatisch neu gestartet.

`java.lang.String setAutostartOn(serviceName)`: Aktiviert den automatischen Start für den Dienst mit dem übergebenen Namen. Der Dienst wird nun bei jedem Server-Neustart automatisch gestartet.

`java.lang.String setAutostartOff(serviceName)`: Deaktiviert den automatischen Start für den Dienst mit dem übergebenen Namen. Bei



einem Server-Neustart, muss der Dienst nun manuell neu gestartet werden.

9.22 SessionManager

Der SessionManager ist zuständig für die Verwaltung der Sessions auf dem Server. Beim Einloggen eines Benutzers wird eine neue Session auf dem Server angelegt. Der SessionManager überprüft in regelmäßigen Intervallen, ob eine Session weiterhin gültig oder bereits abgelaufen ist (vgl. Kapitel 9.8 Seite 487). Eine Übersicht dazu kann auch über das FirstSpirit Server-Monitoring angefordert werden (siehe Kapitel 8.1.3 Seite 432).

9.22.1 Attribute

`ActiveProjectsCount`: Anzahl der Projekte mit aktiven Sitzungen.

`ActiveUsersCount`: Anzahl der Benutzer, die eine aktive Sitzung belegen..

`LicenseSessionCount`: Anzahl der Sessions, die nach den FirstSpirit-Lizenzbedingungen eine FirstSpirit-Lizenz beanspruchen. Temporäre Sessions, d.h. vom Server angelegte Sessions, werden nicht angezeigt.

`MaxSessionCount`: Anzahl der maximal möglichen Sessions basierend auf der FirstSpirit-Lizenz (siehe Kapitel 8.6.1.2 Seite 451).

`SessionCount`: Anzahl der aktuell gültigen Sessions. Temporäre Sessions, d.h. vom Server angelegte Sessions, werden nicht angezeigt. Hinweis: Aktualisierungen, z.B. beim manuellen Löschen von Cookies durch den Benutzer, werden potentiell erst nach dem Timeout von 20 Min. angezeigt.

`SessionCreationRate`: Anzahl der neu geöffneten Sitzungen innerhalb der letzten 60 Sekunden.

`TicketCount`: Anzahl der Tickets auf dem FirstSpirit-Server.

9.22.2 Operations

`void dumpSessions()`: Beim Ausführen dieser Methode werden die Sessioninformationen in die Server-Log-Datei (`fs-server.log`) geschrieben.



10 Sichere Veröffentlichung über rsync und ssh

Für eine Veröffentlichung (Deployment) über ungesicherte Internetverbindungen oder Netze mit geringer Bandbreite empfiehlt sich die Verwendung einer Kombination aus den externen Dienstprogrammen `rsync`³⁶ und `ssh`³⁷.

`ssh` stellt dabei die verschlüsselte Verbindung zwischen FirstSpirit-Server und Webserver bereit und `rsync` reduziert die Menge der übertragenen Daten auf die Änderungen gegenüber der vorherigen Veröffentlichung.

Zur Nutzung dieses Veröffentlichungs-Verfahrens wird auf dem System, auf dem der FirstSpirit-Server installiert ist, der `rsync`- und `ssh`-Client benötigt, sowie die Generierung eines `ssh`-Schlüsselpaares (privater und öffentlicher Schlüssel). Auf dem System des Webserver ist ein `ssh`-Server notwendig sowie der `rsync`-Client. Außerdem muss der öffentliche `ssh`-Schlüssel auf dieses System kopiert werden.

In den folgenden Abschnitten wird die Konfiguration in Abhängigkeit zum Betriebssystem beschrieben. Dabei wird als Beispiel das Benutzerkonto "web" auf dem Webserver mit dem Hostnamen `www.mydomain.net` verwendet und das Dokumentenverzeichnis `/var/www` bzw. `c:\www`.

10.1 Webserver unter Unix

Der `ssh`-Server wird in den meisten Fällen bereits installiert sein. Ist das nicht der Fall, muss er über das Paketsystem des Betriebssystems installiert und aktiviert werden. Für `rsync` wird nur die Client-Installation benötigt (nicht `rsyncd`).

Zunächst muss auf dem Webserver ein Unix-Benutzerkonto "web" angelegt werden, das Lese- und Schreibrechte auf dem Dokumentenverzeichnis (z. B. `/var/www`) des Webserver erhält. Dazu kann z. B. eine Unix-Gruppe angelegt werden, die das Benutzerkonto des Webserver und das neu erstellte "web" enthält.

Anschließend sollte das Anmelden über `ssh` am Benutzerkonto "web" (mit Passwort) von einem anderen Rechner (Unix oder Windows) aus geprüft werden. Dabei ist darauf zu achten, dass der `ssh`-Client das `ssh`-Protokoll 2 nutzt. Unter Windows kann der `ssh`-Client `putty.exe` verwendet werden.

³⁶ `rsync`: <http://rsync.samba.org/>

³⁷ `ssh`: <http://www.openssh.com/>



10.2 Webserver unter Windows



Unter Windows öffnen die meisten Webserver die Dateien anzuzeigender Seiten im exklusiven Lesemodus. Das bedeutet, während der Auslieferung einer Seite an einen Webbrowser kann die zugehörige Datei auf dem Dateisystem des Webserver nicht geändert oder ersetzt werden. Vor dem Einsatz dieses Veröffentlichungs-Verfahrens auf Windows-Servern sollte daher überprüft werden, ob dieses Problem auftritt.

1. Anmeldung als Administrator auf dem Webserver.
2. `setup.exe` von <http://cygwin.com> herunterladen und starten.
3. Als Zielverzeichnis der Installation z. B. `c:\cygwin` eintragen und die Standardoptionen auf "Install for all Users" sowie "Text File Type Unix" belassen. Als "Local Package Directory" beispielsweise `c:\cygwin\cache` wählen.
4. Bei der Abfrage nach zu installierenden Paketen die Standardeinstellung beibehalten und zusätzlich in der Kategorie "Net" die Pakete "openssh" und "rsync" zur Installation auswählen. Dazu die Pfeilsymbole, links neben dem Paketnamen von "Skip", auf die Versionsnummer umstellen (durch klicken). Bei dieser Auswahl werden automatisch andere, benötigte Pakete selektiert.
5. Durch Weiterschalten zum nächsten Fenster wird die Installation abgeschlossen.
6. Zum Abschluss die Cygwin-Shell über das Cygwin-Icon auf dem Desktop oder über das Startmenü aufrufen. Falls dabei, in Abhängigkeit vom Windows-System, eine Abfrage zum Aktualisieren von `/etc/group` und `/etc/passwd` erfolgt, diese bestätigen und die dort angegebene Anleitung befolgen. Anschließend die Shell über Eingabe von "exit" verlassen, erneut aufrufen und geöffnet lassen. Dabei sollte nun keine Abfrage bezüglich `/etc/passwd` mehr erscheinen.
7. In der Cygwin-Shell `ssh-host-config` eingeben und die Parameterabfrage wie folgt beantworten:

Overwrite existing <code>/etc/ssh_config_file</code> ?	Yes
Overwrite existing <code>/etc/sshd_config_file</code> ?	Yes
Create local user <code>sshd_server</code> ?	Yes
Should privilege separation be used?	No
Do you want install <code>sshd</code> as service?	Yes
Value for environment variable <code>CYGWIN</code> =?	ntsec tty
8. Den Systemdienst "CYGWIN sshd" über die Windows-Verwaltung starten.
9. Ein lokales Benutzerkonto "web" anlegen, das Schreibrechte im Dokumentenverzeichnis des Webserver erhält.
10. Die Cygwin-Benutzerdatenbank aktualisieren durch Eingabe folgender Kommandos in der Cygwin-Shell:


```
mkpasswd -l > /etc/passwd
```



```
mkpasswd -l > /etc/group
```

11. Das Anmelden über ssh am Benutzerkonto "web" (mit Passwort) von einem anderen Rechner (Unix oder Windows) aus überprüfen. Dabei darauf achten, dass der ssh-Client das ssh-Protokoll 2 nutzt. Unter Windows kann der ssh-Client `putty.exe` verwendet werden.
12. Um die zu konfigurierende ssh-Anmeldung über ein Schlüsselpaar zu ermöglichen, wird bei der Anmeldung unter dem Benutzerkonto "web" nun Folgendes eingegeben: `ssh-user-config`
(Dabei die Abfrage zur Erzeugung der identity-files mit "no" verneinen.)

10.3 FirstSpirit-Server unter Unix

Die Installation der Clients für ssh und rsync sollte über das Paketsystem des Betriebssystems erfolgen.

Anschließend erfolgt die Konfiguration der rsync/ssh-Verbindung zum Webserver:

1. Unter dem Benutzerkonto des FirstSpirit-Servers (fs5) anmelden bzw. als root "`su - fs5`" verwenden.
2. Ein ssh-Schlüsselpaar erzeugen: `ssh-keygen -t rsa`
Dabei die Standardvorgaben akzeptieren und kein Passwort eintragen.
3. Den öffentlichen Schlüssel auf dem Benutzerkonto "web" des Webserver installieren. Dazu über ssh als "web" anmelden und das Verzeichnis `.ssh` im Home-Directory erstellen: `mkdir .ssh`
Auf dem FirstSpirit-Server als Benutzer "fs5" dann eingeben:
`scp .ssh/id_rsa.pub web@www.mydomain.net:~/.ssh/authorized_keys`
4. Zum Testen der Anmeldung über das Schlüsselpaar (ohne Passwort) eingeben:
`ssh web@www.mydomain.net ls -la /var/www`
5. Ohne Eingabe eines Passworts sollte nun das Dokumentenverzeichnis des Webserver aufgelistet werden.
6. Zum Testen der rsync-Verbindung eingeben:
`rsync -n -vcrzt -e "ssh -l web" www.mydomain.net:/var/www`
7. Ohne Eingabe eines Passworts sollte das Dokumentenverzeichnis des Webserver erneut aufgelistet werden. Aufgrund der Option "-n" wurden keine Dateien übertragen.
Falls als Webserver ein Windows-System eingesetzt wird, die Angabe von `/var/www` durch `/cygdrive/c/www` ersetzen.



10.4 FirstSpirit-Server unter Windows

Als ssh-Client und rsync unter Windows sind die Installations-Pakete von Cygwin³⁸ notwendig. Cygwin bietet eine komfortable Paketverwaltung, die ein späteres Aktualisieren der einzelnen Pakete aus dem Internet ohne weiteren Konfigurationsaufwand ermöglicht. Nachfolgend wird die Installation und Konfiguration von OpenSSH und rsync auf einem Windows-basierten Webserver beschrieben:

1. Anmeldung als Benutzer mit Administrator-Rechten.
2. `setup.exe` von <http://cygwin.com> herunterladen und starten.
3. Als Zielverzeichnis der Installation z. B. `c:\cygwin` eintragen und die Standardoptionen auf "Install for all Users" sowie "Text File Type Unix" belassen. Als "Local Package Directory", z. B. `c:\cygwin\cache` wählen. Das Zielverzeichnis darf keine Leerzeichen enthalten, da ansonsten das im folgenden Abschnitt erwähnte Beanshell-Skript nicht funktioniert!
4. Bei der Abfrage nach zu installierenden Paketen die Standardeinstellung beibehalten und zusätzlich in der Kategorie "Net" die Pakete "openssh" und "rsync" zur Installation auswählen. Dazu die Pfeilsymbole links neben dem Paketnamen von "Skip" auf die Versionsnummer umstellen (durch Klicken). Bei dieser Auswahl werden automatisch andere, benötigte Pakete selektiert.
5. Durch Weiterschalten zum nächsten Fenster wird die Installation abgeschlossen.
6. Zum Abschluss die Cygwin-Shell über das Cygwin-Icon auf dem Desktop oder über das Startmenü aufrufen. Falls dabei, in Abhängigkeit vom Windows-System, eine Abfrage zum Aktualisieren von `/etc/group` und `/etc/passwd` erfolgt, diese bestätigen und die dort angegebene Anleitung befolgen. Anschließend die Shell über Eingabe von "exit" verlassen, erneut aufrufen und geöffnet lassen. Dabei sollte nun keine Abfrage bezüglich `/etc/passwd` mehr erscheinen.

Anschließend erfolgt die Konfiguration der rsync/ssh-Verbindung zum Webserver:

1. Anmeldung als Benutzer mit Administrator-Rechten auf dem Windows-System des FirstSpirit-Servers.
2. Die Cygwin-Shell über das Startmenü aufrufen.
3. Ein ssh-Schlüsselpaar erzeugen:

```
ssh-keygen -t rsa
```

Die Standardvorgaben akzeptieren und kein Passwort eintragen.
4. Mittels des Windows-Explorers die Zugriffsrechte auf die Datei
`c:\cygwin\home\benutzername\.ssh\id_rsa`

³⁸ <http://cygwin.com/>



so ändern, dass zusätzlich das Benutzerkonto SYSTEM Leserechte erhält. Das ist notwendig, da der FirstSpirit-Server unter dem Benutzerkonto SYSTEM läuft. "benutzername" ist hier der aktuelle Benutzername, unter dem im vorherigen Schritt "ssh-keygen" aufgerufen wurde

5. Den öffentlichen Schlüssel auf dem Benutzerkonto "web" des Webservers installieren. Dazu über ssh als "web" anmelden und das Verzeichnis `.ssh` im Home-Directory erstellen:

```
mkdir .ssh
```

In der Cygwin-Shell auf dem FirstSpirit-Server dann eingeben:

```
scp .ssh/id_rsa.pub web@www.mydomain.net:~/.ssh/authorized_keys
```

6. Zum Testen der Anmeldung über das Schlüsselpaar (ohne Passwort) nun die Eingabeaufforderung auf dem Windows-System des FirstSpirit-Servers aufrufen (`cmd.exe`) und eingeben:

```
c:\cygwin\bin\ssh web@www.mydomain.net ls -la /var/www
```

7. Ohne Eingabe eines Passworts sollte nun das Dokumentenverzeichnis des Webservers aufgelistet werden.

8. Zum Testen der rsync-Verbindung nun eingeben:

```
c:\cygwin\bin\rsync -n -vrtz -e "c:\cygwin\bin\ssh -l web" www.mydomain.net:/var/www
```

9. Ohne Eingabe eines Passworts sollte das Dokumentenverzeichnis des Webservers erneut aufgelistet werden. Aufgrund der Option "-n" wurden keine Dateien übertragen.

10. Falls als Webserver ein Windows-System eingesetzt wird, die Angabe von `/var/www` durch `/cygdrive/c/www` ersetzen.



10.5 Konfiguration des FirstSpirit-Projekts

Über den FirstSpirit-Client zur Projekt- und Server-Konfiguration erfolgt nun der Eintrag der ssh/rsync-Veröffentlichung:

1. Einen neuen Auftrag über die Projektkonfiguration erstellen (siehe Kapitel 7.5.4 Seite 360).
2. Zu dem Auftrag die Aktion "Projekt generieren" hinzufügen (siehe Kapitel 7.5.10.2 Seite 385).
3. Zu dem Auftrag die Aktion "Skript ausführen" unter der vorherigen Aktion hinzufügen (siehe Kapitel 7.5.9.4 Seite 373).
4. Das "ssh/rsync-Deployment-Skript" ist Bestandteil des Zip-Archivs (admi5x_files.zip) zum Handbuch für Administratoren und kann über die FirstSpirit Online Dokumentation³⁹ (ODFS) heruntergeladen werden.
5. Als Skript-Parameter werden die Parameter des Webservers eintragen:

```
webhost=www.mydomain.net  
webuser=web  
webpath=/var/www
```

Falls FirstSpirit unter Windows läuft, zusätzlich folgende Parameter hinzufügen:

```
ssh=c:\cygwin\bin\ssh  
rsync=c:\cygwin\bin\rsync  
privkey=/home/benutzername/.ssh/id_rsa
```

"benutzername" ist hier der aktuelle Benutzername, unter dem zuvor "ssh-keygen" aufgerufen wurde. "/home" wird hier tatsächlich unter Windows verwendet, weil Cygwin dieses auf das Windows-Profilverzeichnis abbildet.

6. Danach über den Button "Testen" die eingegebene Konfiguration testen und gegebenenfalls, die angezeigte Protokollierung auf Fehlermeldungen kontrollieren (siehe Kapitel 7.5.9.4 Seite 373). Erst wenn der Test fehlerfrei durchläuft, kann diese Veröffentlichung verwendet werden.

³⁹ .../fs5odfs/de/dokumentation/administratoren/administratoren.html



11 Konfiguration von Benutzerrechten

11.1 Einführung

Im Rahmen des vorliegenden Kapitels werden die von FirstSpirit zur Verfügung gestellten Mechanismen zur Rechtevergabe und -prüfung beschrieben und ihr konkreter Einsatz skizziert. Es geht in den folgenden Kapiteln ausschließlich um die Rechtevergabe für die generierte Site (also um die Vergabe von Benutzerrechten) und weder um die Vergabe von Projektrechten (also um Redaktionsrechte) noch um die Vergabe von Rechten zur Ausführung von Arbeitsabläufen. (Weitere Informationen zur Rechtevergabe siehe FirstSpirit Handbuch für Redakteure).

In FirstSpirit wird streng zwischen Redaktions- und Benutzerrechten unterschieden. Während die Redaktionsrechte für alle Operationen gelten, die ein Redakteur ausführen kann (z. B. Seiten anlegen/ändern/löschen), gelten die Benutzerrechte nur für den "Besucher" der generierten Site und sind damit immer mit dem eingesetzten Personalisierungssystem verknüpft. Wird als Personalisierungssystem FirstSpirit Personalization⁴⁰ eingesetzt (was nicht zwangsläufig der Fall sein muss), so kann ein sehr enger Zusammenhang hergestellt werden. (siehe Kapitel 11.2.3 Seite 522)

Im Bereich der Redaktionsrechte wird die Operationsmenge (Anlegen/Ändern/Löschen/Freigeben usw.) von FirstSpirit fest vorgegeben. Diese Operationen können für Personen oder Gruppen mit Rechten versehen werden. Die Verwaltung von Personen/Gruppen erfolgt ebenfalls nach Vorgabe von FirstSpirit (auch wenn ggf. ein LDAP-System angebunden werden kann). Somit liegen im Bereich der Redaktionsrechte sowohl die Operationen, als auch die Gruppen (relativ) fest.

Im Gegensatz zu den Redaktionsrechten, die sich auf Abläufe im FirstSpirit-Projekt beziehen, beziehen sich die Benutzerrechte ausschließlich auf die generierte und veröffentlichte Site. Die Verwendung von Benutzerrechten in einem Projekt lässt sich in der Regel am Einsatz einer Login-Seite erkennen.

Im Kontext der Benutzerrechte definiert FirstSpirit weder die Operations- noch die Gruppen-Struktur, da jedes mit FirstSpirit implementierte Projekt im Bereich der Benutzerrechte völlig unterschiedliche Anforderungen stellt. In vielen Fällen reicht es aus, wenn Benutzerrechte als "Recht ein Objekte zu betrachten" interpretiert

⁴⁰ Siehe Modul-Dokumentation zu FirstSpirit Personalization



werden. Es sind aber auch Fälle denkbar, in denen neben "ansehen" auch die Operationen "drucken" bzw. "speichern" relevant sind. In diesem Fall muss im Rahmen der Benutzerrechte zwischen den Operationen "ansehen" und "drucken" unterschieden werden.

Wichtig ist in diesem Zusammenhang, dass es in genau zwei Fällen einen Zusammenhang zwischen den Redaktionsrechten und den Benutzerrechten gibt:

- 1) Bei der Vorschau einer Seite:
In diesem Fall ist der Redakteur auch Benutzer – hier treffen die Redaktionsrechte "ansehen" und die Benutzerrechte "ansehen" aufeinander und müssen geeignet verknüpft werden.
- 2) Bei einer Datenänderung aus der Live-Site⁴¹:
In diesem Fall ist der Benutzer auch Redakteur – hier gilt analog, dass die Benutzerrechte "ändern" und die Redaktionsrechte "ändern" geeignet verknüpft werden müssen.

In der Regel wird die Verknüpfung über einen zusätzlichen Login-Request hergestellt, das bedeutet, der Benutzer meldet sich als Redakteur an bzw. umgekehrt.

Es besteht jedoch außerdem die folgende Login-Möglichkeit⁴²:

- SSO: Bei Verwendung des SSO-Moduls erfolgt der Übergang zwischen Redakteur und Benutzer transparent ohne Passwort-Abfrage (Login-Modul: FS SSO)

11.1.1 Definition von Benutzerrechten

Die Vergabe von Benutzerrechten erfolgt immer auf der Basis von Gruppen, da erfahrungsgemäß eine Verwaltung auf Benutzerebene zu großen Problemen, beispielsweise bei der Vertreter-Regelung, führt.

Um die Rechtevergabe zu strukturieren und damit zu vereinfachen, wird davon ausgegangen, dass Gruppen hierarchisch aufgebaut sein können. Eine Gruppe kann also mehrere Untergruppen enthalten.

FirstSpirit bietet eine Unterstützung, um diese Gruppenstrukturen entweder neu zu definieren oder aber aus einem bestehenden System (z. B. LDAP) zu übernehmen.

⁴¹ Siehe Modul-Dokumentation FirstSpirit DynamicDatabaseAccess

⁴² Siehe Modul-Dokumentation FirstSpirit Personalization



Unabhängig von der Herkunft wird die Gruppenhierarchie dem Redakteur in Form einer Baumansicht präsentiert, in der die Rechtekonfiguration vorgenommen werden kann. Bei der Rechtedefinitions-Komponente handelt es sich um eine spezielle Eingabekomponente, mit deren Hilfe Rechte auf der Basis einer hierarchischen Gruppendifinition vergeben werden können. Diese Rechtedefinition bezieht sich ausschließlich auf das Laufzeitsystem und nicht auf das Redaktionssystem selbst, es handelt sich nicht um Redaktionsrechte. In der Regel wird daher die Rechtedefinitions-Komponente im Rahmen der Metadaten verwendet. Ein Einsatz in der Inhalte-Verwaltung oder in der Datenquellen-Verwaltung ist technisch aber auch möglich.

Neben der Gruppenhierarchie existiert bei den Benutzerrechten zusätzlich ein Bezug zur Baumstruktur der FirstSpirit Verwaltungen, die ebenfalls als Hierarchie interpretiert wird.

Die Baumstruktur der FirstSpirit Verwaltungen stellt für die Rechtevergabe eine Vererbungsbeziehung dar. Es gilt daher immer: Wenn in einem Baum-Objekt keine Benutzerrechte definiert sind, so gelten die Rechte des Vater-Objektes. Durch diese Vererbungsdefinition ist es einfach möglich, z. B. auf der Ebene eines Ordners, Rechte für alle darunter liegenden Seiten zu definieren.

Die Vererbung ist als "nicht additiv" definiert – das bedeutet, dass eine Rechte-Definition in einem Objekt alle "darüber liegenden" Definitionen überschreibt.

Da dieses extrem einfache Vererbungsmodell nicht in allen Fällen angemessen ist, gibt es eine Reihe von Möglichkeiten um projektspezifisch "Plausibilitätsregeln" für die Rechtevergabe zu definieren (z. B. "Wenn für eine Obergruppe etwas erlaubt ist kann es für eine Untergruppe nicht verboten sein", oder "Wenn jemand ein Objekt sehen darf, so muss er auch den darüber liegenden Teilbaum betreten können, da er sonst nie bis zum Objekt gelangen kann").

11.1.2 Prüfung von Benutzerrechten

Nachdem nun skizziert wurde, wie Benutzerrechte definiert werden können, werden nun die unterschiedlichen Formen der Prüfung von Benutzerrechten vorgestellt.

Die Definition von Rechten, gleich welcher Art, macht nur Sinn, wenn diese Rechte auch ausgewertet und bei der Auslieferung von Dokumenten berücksichtigt werden.

Dies setzt eine Laufzeitkomponente voraus, die diese Überprüfung ermöglicht. Um eine Überprüfung vornehmen zu können, ist zunächst die Identifikation des Benutzers und eine Ermittlung seiner Gruppen notwendig. Diese Funktionalität wird



vom Modul FirstSpirit Personalization zur Verfügung gestellt.

Eine weitere sich logisch ergebende Konsequenz ist die Notwendigkeit, eine Wirkung aus der Auswertung der Rechte zu erzeugen – also eine Reaktion. In einigen Fällen erfolgt die Reaktion im Rahmen der Personalisierung.

Dabei werden durch spezielle Tags (FirstSpirit-Personalization-Tags⁴³) Teile einer Seite oder aber auch die komplette Seite geschützt. Diese Form der Reaktion ist aber nur bei JSP-Seiten möglich. Reine HTML-Seiten, PDF-Dokumente oder Bilder sind durch dieses Verfahren nicht zu schützen. Ergänzend zu diesem Konzept ist daher das Modul FirstSpirit Security vorhanden (siehe Kapitel 11.3.2 Seite 525). Darüber wird der auf der Ebene des HTTP-Servers, je nach Rechtekonfiguration, die Auslieferung von Dokumenten bzw. Dateien unterbunden. Eine auf Medien aus der Medien-Verwaltung eingeschränkte und mit der Rechtedefinitions-Komponente verknüpfte Lösung ist in Form des "Sichere Medien" Konzepts vorhanden (siehe Kapitel 11.3.3 Seite 527). Mehr zum Thema "Schutz personalisierter Inhalte" siehe Kapitel 11.3 Seite 523.

Es gibt eine Entsprechung der Rechtedefinitions-Komponente im Laufzeitsystem (Soll-/Ist-Vergleich der Rechtekonfiguration eines Objekts mit der Konfiguration des Benutzers) (siehe Kapitel 11.3.2 Seite 525). Wesentlich ist dabei, dass die gleichen Gruppenhierarchie-Beziehungen verwendet werden.

Zu diesem Zweck wird ein Filter eingesetzt, der in der Lage ist, die Auslieferung von nicht-aktiven Dokumenten zu kontrollieren. Dieser "Multi-Access-Control-Filter" muss projektspezifisch konfiguriert werden (siehe Kapitel 11.3.2.2 Seite 526).

Im Detail werden diese Mechanismen in Kapitel 11.2.3 ab Seite 522 beschrieben.

11.1.3 Vergabe von Benutzerrechten

Die Benutzerrechte-Vergabe und -Prüfung in FirstSpirit erfolgt über zwei Komponenten:

- Editor (Rechtedefinitions-Komponente) – (siehe Kapitel 11.1.3.1 Seite 516)
- Service (PermissionService) – (siehe Kapitel 11.1.3.2 Seite 516)

Diese beiden Komponenten sind ein Bestandteil des System-Moduls, welches bereits bei der Installation eines FirstSpirit-Servers enthalten ist und nicht separat installiert werden muss.

⁴³ Siehe Modul-Dokumentation FirstSpirit Personalization



11.1.3.1 Rechtedefinitions-Komponente (CMS_INPUT_PERMISSION)

Die Rechtedefinitions-Komponente ist eine Kombination aus GUI- und Render-Komponente. Über diese Komponente kann der FirstSpirit-Client um benutzerspezifische Eingabemöglichkeiten, in diesem Fall die Definition von Rechten, erweitert werden. Die Rechtedefinitions-Komponente kommt in der Regel als Eingabekomponente im Metadaten-Reiter zum Einsatz. Die Komponente wird (projektspezifisch im Metadaten-Formular) mit einem eindeutigen Gruppendokument parametrisiert und arbeitet mit dem passenden Service zusammen, der die Gruppendefinitionen vom Server lädt und bereitstellt (siehe Kapitel 11.1.3.2).

Zur Benutzung der Rechtedefinitionskomponente siehe auch FirstSpirit Online Dokumentation.

11.1.3.2 Permission-Service

Der Permission-Service ist eine Serverkomponente, die über die Rechtedefinitions-Komponente angesprochen werden kann. Der Permission-Service ist ein spezieller Dienst des FirstSpirit-Servers, dessen Aufgabe in der Verwaltung von Gruppen- und Benutzerkonfigurationen besteht. Als Systemdienst kann der Permission-Service über das FirstSpirit Server-Monitoring aktiviert werden (Konfiguration der Datei `fs-server.conf` über das FirstSpirit Server-Monitoring siehe Kapitel 11.4.2 "Aktivierung des Permission Service" Seite 533).



11.2 Architektur

11.2.1 Einleitung

Wie bereits in der Einleitung ersichtlich ist, handelt es sich bei dem Thema der Berechtigungsprüfung um einen komplexen Aspekt, der viele Teile des FirstSpirit-Systems betrifft. In diesem Kapitel werden daher der Zusammenhang und die Abhängigkeiten der Einzelteile auf der Architektur-Ebene vorgestellt. Die wesentlichen Bestandteile der Berechtigungsprüfung in FirstSpirit sind:

- a) Rechtedefinitions-Komponente (Client): Definiert die Soll-Berechtigung für Objekte (i. d. R. über Metadaten).
- b) Rechtedefinitions-Komponente (Server): Definiert die Gruppen- und Benutzerstrukturen, die als Grundlage für die Berechtigungsdefinition dienen
- c) Web-Applikationen (Preview/Staging/Live): Dienen dazu, die Berechtigungsprüfung bei der Dokumentenauslieferung durchzuführen
- d) Externe Datenquellen: Liefern die Grundlage für die Definition der Gruppenstrukturen. Hierbei wird eine (aus Sicht von FirstSpirit) externe Datenquelle (z. B. ein LDAP-Server) verwendet.
- e) Interne Datenquellen: Liefern die Grundlage für die Definition der Gruppenstrukturen. Hierbei wird eine (aus Sicht von FirstSpirit) interne Datenquelle verwendet.

Wesentlich für das Verständnis des Zusammenspiels der einzelnen Komponenten sind die Beziehungen und die Datenflüsse zwischen den Bestandteilen.



11.2.2 Übersicht

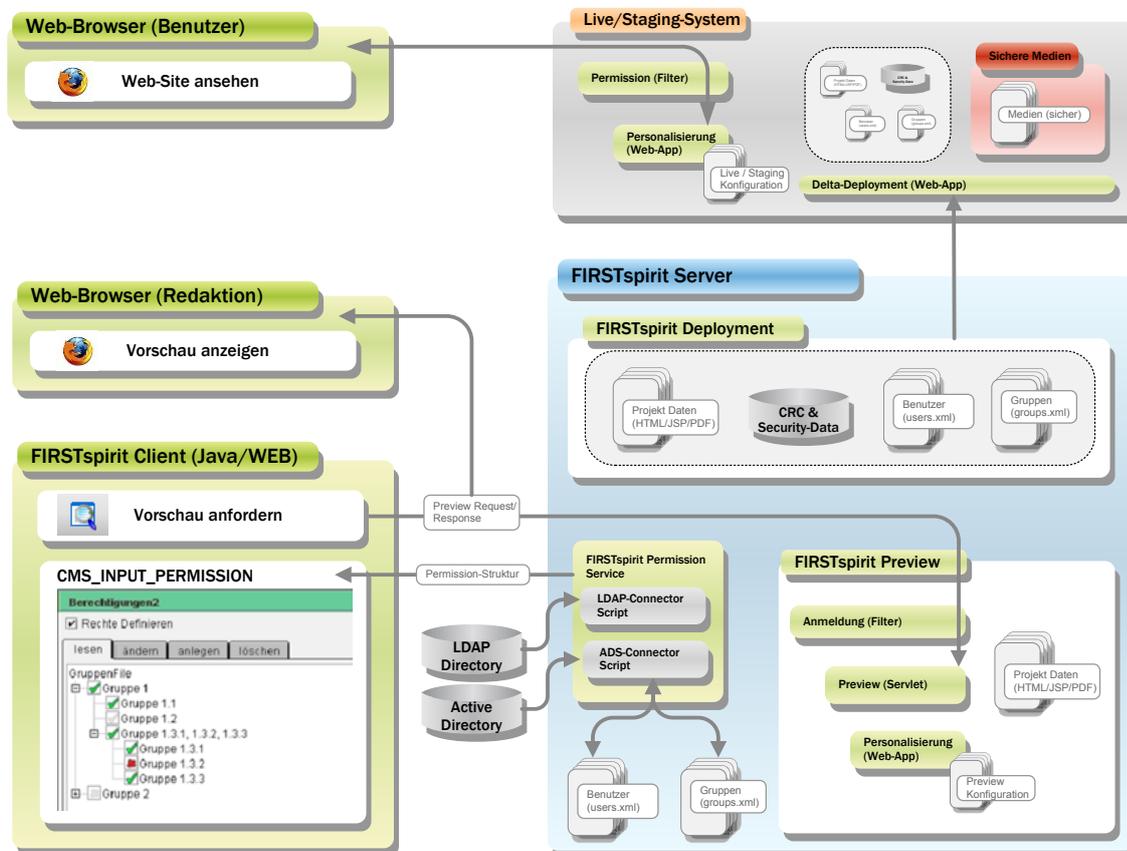


Abbildung 11-1: Übersicht Live-System

FirstSpirit-Clients: Kommunizieren ausschließlich mit dem FirstSpirit-Server, nie direkt z. B. mit einem LDAP-Server oder einer Datenbank (weitere Informationen siehe Kapitel 11.2.2.1 Seite 519).

FirstSpirit-Server: Bearbeitet alle Client-Anfragen, verwaltet Zugriffe auf externe Ressourcen (Datenbanken usw.), erzeugt Vorschau-Seiten, bedient das Staging- und Live-System (weitere Informationen siehe Kapitel 11.2.2.1 Seite 519).

Staging-System: (Auch "Generierungsverzeichnis") dient zur Endüberprüfung des Gesamtsystems inkl. ggf. erfolgter Applikationsintegration. Hier müssen auch alle im Live-System vorhandenen Webanwendungen und Module laufen. Im Gegensatz zum Preview-System sind die Verzeichnis- und Dateinamen identisch mit den Strukturen des späteren Live-Systems. Die Konfigurationen aller Webanwendungen entsprechen im Wesentlichen denen des Live-Systems, abgesehen von absoluten Pfaden und URL-Präfixen (weitere Informationen siehe Kapitel 11.2.2.2 Seite 520).

Live-System: Das für den Endbenutzer sichtbare System. Hierhin werden alle relevanten Daten im Rahmen der Veröffentlichungen übertragen (weitere



Informationen siehe Kapitel 11.2.2.3 Seite 522).

11.2.2.1 FirstSpirit-Client, -Server und -Vorschauerzeugung

Die Rechtedefinitions-Komponente kommt in der Regel als Eingabekomponente im Metadaten-Reiter zum Einsatz. Die Komponente wird (projektspezifisch im Metadaten-Formular) mit einem eindeutigen Gruppendokument parametrisiert. Mit diesem Namen ermittelt die Rechtedefinitions-Komponente beim FirstSpirit-Server die zugehörige Struktur. Dies erfolgt über den "Permission Service". Dieser Service ist ein spezieller Dienst des FirstSpirit-Servers, dessen Aufgabe in der Verwaltung von Gruppen- und Benutzerkonfigurationen besteht (Konfiguration in der Datei `fs-server.conf` siehe Kapitel 11.4.2 "Aktivierung des Permission Service" Seite 533).

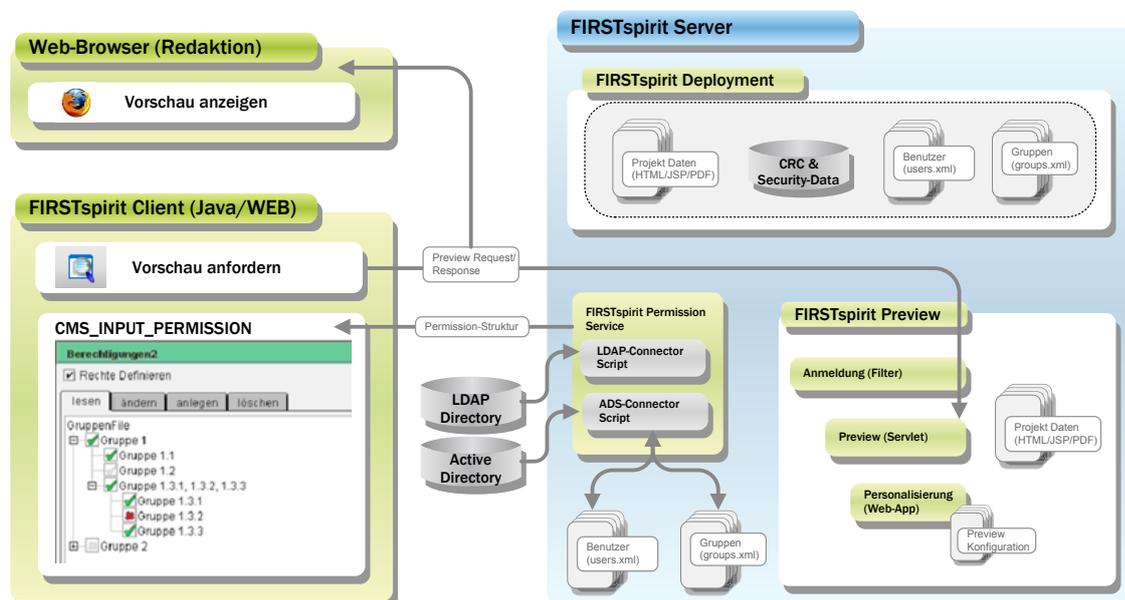


Abbildung 11-2: Ausschnitt Live-System

Die vom "Permission Service" verwalteten Gruppen- und Benutzerkonfigurationen werden in der Servicekonfigurationsdatei ("`service.ini`") konfiguriert (siehe Kapitel 11.4.3.1 "Aufbau der Servicekonfigurationsdatei `service.ini`" Seite 534). Die zugehörigen XML-Dateien (siehe FirstSpirit Handbuch für Entwickler) können entweder manuell erzeugt werden (über das FirstSpirit Server-Monitoring siehe Kapitel 8.6.1.7 Seite 455) oder aber automatisch über ein Connector-Skript auf Basis eines bestehenden Benutzer/Gruppen-Verwaltungssystems (z. B. LDAP oder Active-Directory) (siehe FirstSpirit Handbuch für Entwickler).

Mithilfe einer Gruppen-Struktur-Definition kann anschließend vom Redakteur eine Rechtedefinition für ein Objekt im FirstSpirit-Client vorgenommen werden.



Wie in Kapitel 11.6 "Einsatz im Projekt" beschrieben ist, kann die Rechtedefinition entweder als "Soll-Berechtigung" für die Personalisierung dienen (siehe Kapitel 11.3.1 Seite 523) oder im Rahmen des Zugriffsschutzes über die Access-Control-Datenbank und den Multi-Access-Control-Filter (siehe Kapitel 11.3.2.2 Seite 526) verwendet werden.

An dieser Stelle greift ein weiterer Sicherheits-Mechanismus, der zwar in keinem Zusammenhang mit der Benutzerrechte-Definition steht aber hier dennoch der Vollständigkeit halber erwähnt werden soll: Das "Preview-Servlet". Die Aufgabe des Preview-Servlets ist die Überwachung der Zugriffsberechtigungen bei der Anforderung einer Vorschau (siehe Kapitel 11.3.3.1 Seite 527). Dabei erfolgt die Content-Auslieferung durch das Preview-Servlet, wobei der Request intern an den Webserver weitergeleitet wird. Diese Architektur ist notwendig, wenn eine HTTP-Server-spezifische Erweiterung genutzt werden soll, beispielsweise ASP- oder PHP-Extensions in der Vorschau.

Während das Preview-Servlet nur auf der Zugriffsebene greift, arbeitet der "Multi-Access-Control-Filter" auf der Objektebene (siehe Kapitel 11.3.2.2 Seite 526). Der wesentliche Unterschied zur Konfiguration im Live-System ist, dass in der Vorschau direkt auf die von den FirstSpirit Web-Applikationen benötigten Dateien (z. B. Content-Schema für OR-Runtime bzw. `user.xml` und `groups.xml` für die Personalisierung) zugegriffen werden kann, während im Live-System Kopien der Dateien veröffentlicht werden müssen.

11.2.2.2 Staging-System (Generierung)

Das Staging-System unterscheidet sich vom FirstSpirit Vorschau-System vor allem durch die Organisation der Dateiablage. Während im Vorschau-System aus Performancegründen alle Dateien mit "künstlichen" Namen und ohne Verzeichnisse erzeugt werden, entspricht die Struktur im Staging-System der des Live-Systems. Dadurch wird es möglich auch Applikationen zu integrieren, die innerhalb der Vorschau (auf Grund der genannten Unterschiede) prinzipbedingt nicht zum Laufen zu bringen sind.

Der grundlegende Aufbau der HTTP-Server-Infrastruktur entspricht der des Vorschau-Systems (siehe Kapitel 11.2.2.1 Seite 519)



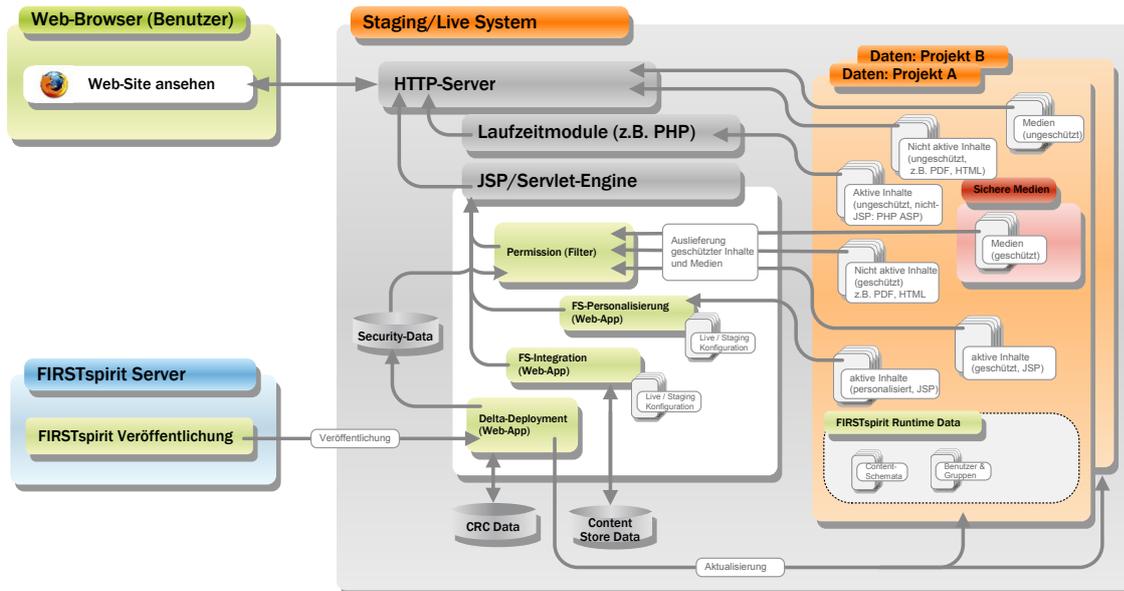


Abbildung 11-3: Live- und Staging-System

Der Multi-Access-Control-Filter ist nur dann wirksam, wenn das Objekt auch wirklich durch die Servlet-Engine ausgeliefert wird. Dieser Aspekt soll noch etwas eingehender am Beispiel einer PDF-Datei verdeutlicht werden, die z. B. aus einer Seite der Inhalte-Verwaltung erzeugt wurde, aber kein "Sicheres Medium" ist. Im ungünstigen Fall wird das PDF-Dokument direkt durch den HTTP-Server ausgeliefert. Damit hat der Multi-Access-Control-Filter, der Teil der Servlet-Engine ist, keine "Chance" die Auslieferung zu verhindern. Es muss sichergestellt werden, dass der Request vom HTTP-Server an die Servlet-Engine weitergegeben und dort bearbeitet wird.

Zu beachten ist hier, dass die Content-Auslieferung durch die Servlet-Engine erfolgt. Dieses Verfahren ist weniger effizient als die direkte Auslieferung durch den HTTP-Server. Zusätzlich kommt im Staging- und im Live-System erschwerend hinzu, dass direkt mit dem Access-Control-Datenbank kommuniziert wird, um die Berechtigung zu prüfen.



11.2.2.3 Live-System

Im Gegensatz zum Staging-System hat der FirstSpirit-Server keine Kontrolle über die Konfiguration der Web-Applikationen im Live-System. Daher muss die Konfiguration manuell erfolgen. Eine Möglichkeit ist, die automatisch erzeugten Konfigurationsdateien der FirstSpirit-Web-Applikationen als Ausgangsbasis zu verwenden. In der Regel müssen nur einige Pfade auf die Gegebenheiten des Live-Systems angepasst werden (siehe Abbildung 11-3).

Die Aktualisierung des Live-Systems erfolgt vom FirstSpirit-Server aus über den Deployment-Manager. Dieser sorgt für die Zusammenstellung aller relevanten Daten und den Upload auf das Live-System. Im Live-System erfolgt die Verwaltung der Zugriffsprüfung entweder eigenständig oder aber auf Basis der Informationen aus der Access-Control-Datenbank (siehe Kapitel 11.3.2 Seite 525).

11.2.3 Berechtigungsprüfung mit FirstSpirit

11.2.3.1 Schnelleinstieg

Der Schnelleinstieg gibt einen Ausblick darüber, welche Schritte in welcher Reihenfolge durchgeführt werden müssen, um die Berechtigungsprüfung in FirstSpirit zu aktivieren:

1. Permission-Service starten: Der Service ist Bestandteil des System-Moduls, welches bereits initial auf dem Server existiert und nicht extra installiert werden muss. Allerdings enthält der Service Dienste, die eventuell gestartet werden müssen, wenn sie nicht bereits automatisch gestartet wurden. Das Starten bzw. Stoppen der Dienste erfolgt entweder über das FirstSpirit Server-Monitoring (siehe Kapitel 8.6.2.4 Seite 463) oder über die Server- und Projektkonfiguration (siehe Kapitel 7.3.11 Seite 254).
2. Systemkonfigurationsdateien (`service.ini`, `groups.xml` und `users.xml`) des Permission-Service anpassen (siehe Kapitel 11.4.3 Seite 533)
3. Metadatenvorlage erstellen (siehe FirstSpirit Handbuch für Entwickler)
`CMS_INPUT_PERMISSION` und `sichere Medien`
4. Web-Komponenten über die Server- und Projektkonfiguration konfigurieren (siehe Kapitel 7.4.18 Seite 327).



5. Berechtigungen über die Server- und Projektkonfiguration konfigurieren (siehe Kapitel 11.4.5 Seite 539)
 - Angabe der Eingabekomponente für sichere Medien
 - Angabe der Eingabekomponente für Rechteprüfung
6. Konfiguration des Multi-Access-Control-Filters (siehe Kapitel 11.3.2.2 Seite 526)

Siehe auch Modul-Dokumentation FirstSpirit Security.

In den nachfolgenden Kapiteln werden die Einsatzszenarien und die einzelnen Konfigurationsmöglichkeiten ausführlich beschrieben.

11.3 Schutz personalisierter Projektinhalte in FirstSpirit

FirstSpirit kennt unterschiedliche Konzepte um Projektinhalte vor unberechtigtem Zugriff zu schützen:

- Personalisierung von Inhalten (siehe Kapitel 11.3.1 Seite 523)
- Prüfung der Zugriffsrechte auf ein Objekt über die Access-Control-Datenbank (siehe Kapitel 11.3.2 Seite 525)
- Zugriffsschutz für Medien innerhalb der FirstSpirit Vorschau ("Sichere Medien") (siehe Kapitel 11.3.3 Seite 527)

11.3.1 Personalisierung

Das Modul FirstSpirit Personalization⁴⁴ besteht aus einer Java Tag-Library, die sehr einfach genutzt werden kann, um JSP-Seiten zu personalisieren, d.h. es kann entweder das Anzeigen der Seite komplett oder teilweise unterbunden werden. Ob ein Teilbereich einer Seite sichtbar ist oder nicht hängt davon ab, ob der angemeldete Benutzer über eine ausreichende Berechtigung verfügt. Dies wird entschieden, indem die Soll-Berechtigung der Seite mit der Ist-Berechtigung des Benutzers verglichen wird.

Die Definition der Soll-Berechtigung (Wer darf ein Dokument oder einen Teil des Dokumentes sehen?) erfolgt über die Rechtedefinitions-Komponente. Die Auswertung, d.h. der Vergleich der Ist-Konfiguration, die sich aus dem Login-Kontext des Benutzers ergibt und der Soll-Konfiguration wird durch die Personalisierung durchgeführt. Dabei besteht ein sehr enger (logischer) Zusammenhang zwischen

⁴⁴ Modul FirstSpirit Personalization (siehe Dokumentation PERS40DE_FirstSpirit_Modules_Personalization)



den Gruppen, die in der Rechtedefinitions-Komponente verwendet werden und den Gruppen, auf die sich das "GroupModule" der Personalisierung bezieht.

Beispiel: Wenn ein Benutzer zu einer Gruppe gehört, die nicht in der Rechtedefinitions-Komponente auftaucht, können für ihn unter Umständen keine Rechte gesetzt werden.

Dieser Zusammenhang zwischen Gruppen-Modell in der Rechtedefinitions-Komponente und im Personalisierungs-Modul kann auf folgenden zwei Wegen hergestellt werden:

1. Die Erzeugung der Gruppen-Definitionsdatei für die Rechtedefinitions-Komponente und das "GroupModule" der Personalisierung werten die gleichen externen Datenquellen aus (z. B.: LDAP-Server oder AD-Server). In diesem Fall liegt die Datenhoheit komplett extern.
2. Die Rechtedefinitions-Komponente erzeugt eine Gruppen-Definitionsdatei und das Personalisierungs-Modul verwendet diese Daten bzw. bezieht sich auf diese Datei. Die Datei kann von der Rechtedefinitions-Komponente durchaus durch die Abfrage eines LDAP-Servers erzeugt werden. Der Unterschied ist, dass das Personalisierungsmodul selbst nicht den LDAP-Server abfragt, sondern die von der Rechtedefinitions-Komponente erzeugte Datenbasis verwendet.

Während also im ersten Fall das "GroupModule" der Personalisierung auf die Modi "ContentStore" oder "LDAP" konfiguriert wird (und damit die Laufzeitumgebung eine permanente Verbindung zur Datenquelle benötigt), ist für den zweiten Fall das "GroupModule" auf "GroupService" zu konfigurieren und verwendet dann die gleichen XML-Dateien wie die Rechtedefinitions-Komponente. Im zweiten Fall ist daher kein permanenter Zugriff auf eine externe Ressource erforderlich, aber es ist zu beachten, dass bei der Deployment-Konfiguration die Gruppen-Konfigurationsdateien auch auf das Live-System veröffentlicht werden (siehe Kapitel 11.4.4 Seite 539).

Durch den Einsatz der Personalisierung in FirstSpirit können Berechtigungsprüfungen ausschließlich auf JSP-Seiten erzwungen werden. Durch das personalisierte Ausblenden von Links kann damit ein Zugriff auf nicht JSP-Dokumente (z. B. PDF-Dateien oder Bilder) erschwert werden. Dies stellt aber keinen zuverlässigen Schutz dar, da die Auslieferung durch die Direkteingabe des Links erzwungen werden kann. Das Modul sollte daher immer mit dem Standard-Modul FirstSpirit Security kombiniert werden (Konzept siehe Kapitel 11.3.2 Seite 525).



Beim Modul FirstSpirit Personalization handelt es sich um eine kostenpflichtige Zusatzfunktionalität.

Für weitere Informationen siehe Modul-Dokumentation FirstSpirit Personalization.

11.3.2 Prüfung der Zugriffsrechte über die Access-Control-Datenbank

Um einen zuverlässigen Schutz für Objekte zu realisieren, die eine Berechtigungsprüfung nicht selbst durchführen können, muss daher bereits auf Ebene der Dateiauslieferung angesetzt werden. Dafür steht in FirstSpirit das Modul FirstSpirit Security zur Verfügung. Bei der Generierung eines Projekts wird eine lokale Access-Control-Datenbank angelegt, die alle Berechtigungsinformationen für die einzelnen Projektinhalte verwaltet. Der Abgleich dieser lokalen Access-Control-Datenbank mit dem Live-System erfolgt über das Modul FirstSpirit Security (über das CRC-Transfer-Servlet – siehe Modul-Dokumentation FirstSpirit Security). Über das Modul kann u.a. ein Filter definiert werden ("Multi-Access-Control-Filter"), der die Auslieferung aller Objekte, auf die die Filterkriterien passen, zuverlässig unterbindet (Der Server liefert die geschützten Dateien nicht aus.).

Das Konzept "Sichere Medien" kann im Live-System ebenfalls über das Modul FirstSpirit Security und einen individuell angepassten Filter auf das gewünschte "Sichere Medien" Verzeichnis (bzw. die Medien-Datei) realisiert werden⁴⁵.

Die Konfiguration erfolgt über die FirstSpirit Server- und Projektkonfiguration (siehe Kapitel 7.4.16 Seite 323).

Für weitere Informationen siehe Modul-Dokumentation FirstSpirit Security.

11.3.2.1 Access-Control-Datenbank

Ein Teilbereich des Moduls FirstSpirit Security ist die Access-Control-Datenbank. Eine Access-Control-Datenbank besteht aus mehreren "Access Control Lists" (ACL). Die Access-Control-Datenbank verwaltet beliebige Informationen zu Projektdateien (z. B. Seiten, Medien). Neben vielen anderen Informationen, beispielsweise die CRC-32-Prüfsumme des Objektes, die für den differentiellen Upload benötigt wird (vgl. Kapitel 7.5.10.6.3 Seite 399), können über die Access-Control-Datenbank auch die Zugriffsrechte auf ein Objekt gespeichert werden (d.h. Rechte eines Besuchers der generierten Site – "Benutzerrechte"). Mithilfe von individuell konfigurierten

⁴⁵ Modul FirstSpirit Security (siehe Dokumentation SECU40DE_FirstSpirit_Modules_Security)



Filtern, die ebenfalls über das Modul zur Verfügung gestellt werden, kann so der Zugriff auf generierte Projekthinhalte geregelt werden (siehe "Multi-Access-Control-Filter", Kapitel 11.3.2.2)

Die lokale Access-Control-Datenbank eines Auftrags wird bei der ersten Generierung angelegt und bei jeder weiteren Generierung aktualisiert.

Für weitere Informationen siehe Modul-Dokumentation FirstSpirit Security.

11.3.2.2 Multi-Access-Control-Filter

Über den Multi-Access-Control-Filter können unterschiedliche Filter für Projekthinhalte definiert werden, für die, innerhalb der generierten bzw. veröffentlichten Inhalte, ein spezieller Zugriffsschutz gewünscht ist. Die Aufgabe des Multi-Access-Control-Filters ist es, eine bestimmte Klasse von Objekten vor der Auslieferung hinsichtlich ihrer Berechtigungskonfiguration zu prüfen. Der Filter nutzt für die Prüfung die Informationen aus der Access-Control-Datenbank.

Der Multi-Access-Control-Filter ist als Servlet-Filter realisiert, der im Staging- und Live-System in der entsprechenden Webanwendung konfiguriert werden muss. Dabei kann ein bestimmtes Generierungsverzeichnis (oder ein einzelnes Objekt) angegeben werden, das durch den Filter geprüft werden soll. Im Rahmen des Mappings können die Geltungsbereiche des Filters definiert werden. Dazu wird ein URL-Pattern definiert mit dem URL-Klassen und bestimmte Datei-Filter beschrieben werden können. Der Multi-Access-Control-Filter kann über Parameter weiter angepasst werden, indem beispielsweise nur bestimmte Zugriffsrechte zur Überprüfung angegeben werden, oder die Filterung auf bestimmte Dateien (z. B. nur PDF-Dateien) eingeschränkt wird. Innerhalb einer Konfiguration können mehrere unterschiedliche Filter konfiguriert werden.

Bei der Definition des URL-Patterns sollte berücksichtigt werden, dass die Ausführung des Servlets vor der Auslieferung eines Objektes, auf das ein Filter-Kriterium passt, einen zusätzlichen Rechenaufwand zur Folge hat. Daher sollten URL-Pattern immer so "klein" wie möglich gehalten werden. Es sollte also keinesfalls "/" gemappt werden, wenn "*.pdf" bereits ausreichen würde.

Für weitere Informationen siehe Modul-Dokumentation FirstSpirit Security.



11.3.3 Konzept Sichere Medien

Ein bekanntes Konzept sind die sogenannten "Sicheren Medien". Dabei werden bestimmte Dateien der Medien-Verwaltung vor unberechtigtem Zugriff geschützt:

- Innerhalb der FirstSpirit Vorschauerzeugung über ein Servlet (Preview-Servlet) (siehe Kapitel 11.3.3.1 Seite 527)
- Innerhalb der Live-Systems über einen Filter (Multi-Access-Control-Filter) (siehe Kapitel 11.3.3.2 Seite 528)

Beide Mechanismen, sowohl das Servlet als auch der Filter, prüfen vor der Auslieferung die Berechtigung des Benutzers.

11.3.3.1 Sichere Medien innerhalb der FirstSpirit Vorschau

Innerhalb der FirstSpirit-Vorschau kann der Zugriffsschutz für genau ein Medien-Verzeichnis innerhalb der Projekteigenschaften im Bereich "Berechtigungen" definiert werden (siehe Kapitel 7.4.16 Seite 323). Dort wird ein Ordner der Medien-Verwaltung als Sichere-Medien-Ordner definiert. Alle Inhalte innerhalb des Ordners sind dann innerhalb der FirstSpirit-Vorschau (nur dort!) vor unberechtigtem Zugriff geschützt.

Die Auslieferung der (innerhalb des Ordners liegenden) Medien wird über ein Servlet ("Preview-Servlet") innerhalb der Vorschauerzeugung unterbunden. Die Aufgabe des Preview-Servlets ist die Überwachung der Zugriffsberechtigungen bei der Anforderung einer Vorschau (siehe Kapitel 11.2.2.1 Seite 519). Dabei werden die Benutzerrechte ausgewertet, die für ein Objekt in der Eingabekomponente für Benutzerrechte hinterlegt sind. Dabei prüft das Preview-Servlet gegen den PermissionService. Die hier hinterlegten Berechtigungen entsprechen den Informationen, die für ein Objekt in der Access-Control-Datenbank hinterlegt werden, sind aber aktueller ("Current"-Stand). Sofern diese Informationen nicht vorhanden sind, werden die internen "Redaktionsrechte" von FirstSpirit ausgewertet (für Medien: "can see").

Das Preview-Servlet steht über die Standard-Webanwendung fs5preview zur Verfügung. Konfigurationseinstellungen, die über die Definition des Sicheren-Medien-Ordners hinausgehen, sind nicht notwendig.

Die Installation der Webanwendung FirstSpirit Security WebApp ist im Webbereich "Vorschau (Preview)" nicht notwendig.



11.3.3.2 Sichere Medien innerhalb des Live-Systems



Das in einer früheren FirstSpirit-Version eingeführte Modul FirstSpirit Security erfordert Änderungen für die Konfiguration (im Live-Stand), da sich der Zugriffsschutz sich hier seitdem nicht mehr länger nur auf Medien beschränkt, sondern alle Projektinhalte einschließt. Das Prinzip der Sicheren Medien bezieht sich nur auf die Preview des Projekts. Sollen Medien auch im Live-Stand geschützt sein, sind weitere Schritte notwendig.

Für den Schutz der Sicheren Medien innerhalb des Live-Systems ist die Installation des Moduls FirstSpirit Security und die Konfiguration eines individuellen, auf das Sichere-Medien-Verzeichnis angepassten Filters (siehe Kapitel 11.3.2.2 Seite 526) erforderlich⁴⁶. Die Rechte werden in diesem Fall nicht über das Preview-Servlet kontrolliert, sondern basieren auf Informationen aus einer Access-Control-Datenbank (siehe Kapitel 11.3.2.1 Seite 525).

Für weitere Informationen siehe Modul-Dokumentation FirstSpirit Security.

11.3.4 Abgrenzung

Die in den vorangehenden Kapiteln beschriebenen Mechanismen dienen alle dem Ziel, die Prüfung einer Berechtigungskonfiguration durchzuführen. Daher stellt sich die Frage, wann welche Verfahren zu verwenden sind und in welcher Beziehung diese Verfahren zueinander stehen. Zwingende Voraussetzungen für die Berechtigungsprüfung sind:

- Verwendung der FirstSpirit Rechtedefinitions-Komponente
- Verwendung des Moduls FirstSpirit Personalization

Im Einzelfall könnte zwar auch eine andere Rechtedefinitions-Komponente oder ein anderes Personalisierungsmodul verwendet werden, aber in beiden Fällen sind Anpassungen erforderlich.

Grundsätzlich kann man sagen, dass die Berechtigungsprüfung auf der Ebene von Absätzen (also innerhalb einer Seite) ausschließlich mit dem FirstSpirit Personalisierungs-Modul möglich ist. Sollen eigenständige Objekte (z. B. Seiten, Medien oder PDF-Dokumente) geschützt werden, so kann ergänzend zur Personalisierung der Mechanismus "Sichere Medien" (siehe Kapitel 11.3.3 Seite

⁴⁶ siehe Dokumentation SECU40DE_FirstSpirit_Modules_Security.pdf



527) oder die "Prüfung der Zugriffsrechte über die Access-Control-Datenbank (siehe Kapitel 11.3.2 Seite 525) eingesetzt werden.

11.3.5 Berechtigungsdefinition

11.3.5.1 Hierarchische Semantik

Die grundsätzliche Problematik einer Eingabekomponente für die Rechtedefinition liegt dabei in den folgenden Aspekten:

1. Die Gruppen bilden in der Regel eine Hierarchie, die geeignet normalisiert werden muss. Dies heißt im Umkehrschluss: Benutzer mit Sonderfunktionen sind als separate Gruppe zu normalisieren (siehe Kapitel 11.3.5.2 Seite 529).
2. Die Verwaltungen bilden ebenfalls eine Hierarchie, die häufig für die Vererbung von Rechtedefinitionen genutzt wird (siehe Kapitel 11.3.5.3 Seite 531).
3. Es wird eine Modifikation von vererbten Rechten gewünscht

Die Verbindung dieser drei Punkte führt zu einer erheblichen, nicht auf den ersten Blick ersichtlichen Komplexität bei der Rechtedefinition, da:

1. zu definieren ist, was es bedeutet, wenn eine Untergruppe etwas darf, was die Obergruppe nicht darf. Analog: Verbot
2. zu definieren ist, welche Semantik die Verwaltung hat:
 - a) keine (z. B. Medien-Verwaltung/Inhalte-Verwaltung) d.h. nur Vererbung.
 - b) Hierarchie (z. B. Navigation über die Struktur-Verwaltung) d.h. Vererbung und Einschränkung
3. eine Unterscheidung zwischen "erlaubt", "verboten", "vererbt" erforderlich ist.

11.3.5.2 Hierarchische Semantik (Gruppen)

Unter einer Gruppenhierarchie versteht man eine Organisationsform bei der Gruppen wieder aus einzelnen Gruppen bestehen können.

In FirstSpirit ist es möglich, auf jeder (in der Rechtedefinitions-Komponente) unterhalb des Root-Knotens liegenden Ebene der Gruppenhierarchie Rechte zu definieren. Diese Funktionalität bietet speziell bei der Konfiguration von Benutzerrechten eine Reihe von Vorteilen, da es so möglich ist auf einer



Obergruppe Rechte zu definieren, die dann für alle Untergruppen gelten. So kann die Anzahl der notwendigen Konfigurationen auch bei vielen Gruppen gering gehalten werden.

In der FirstSpirit Rechtedefinitions-Komponente ist es möglich auf jeder unterhalb des Root-Knotens liegenden Ebene der Gruppenhierarchie einen der folgenden Werte zu definieren:

- Erlaubt
- Verboten
- Vererbt (d.h. entspricht der Einstellung der Obergruppe)

Es können auf allen Ebenen parallel Rechtedefinitionen erfolgen – hinsichtlich der "Sinnhaftigkeit" der Konfiguration trifft die FirstSpirit Rechtedefinitions-Komponente keine Annahmen. Plausibilitätsprüfungen bzw. Validierungen können projektspezifisch umgesetzt werden.

Mit der Einführung von Gruppenhierarchien geht aber auch eine Reihe von Problemen einher:

- 1) Was passiert bei "Widersprüchen", beispielsweise wenn eine Untergruppe etwas darf, was die Obergruppe nicht darf?
- 2) Ist eine Obergruppe eine eigenständige Entität?
- 3) Was passiert, wenn sich die Gruppenhierarchie ändert?

Da diese Fragen nicht universell für jeden Anwendungsfall zu beantworten sind, bietet das Konzept der Benutzerrechte Ausdrucksmittel, um möglichst alle möglichen Antworten abbilden zu können. Konkret bedeutet dies:

- 1) Die Liste der erlaubten und der verbotenen Gruppen wird getrennt verwaltet und kann einzeln ausgewertet werden. So können individuelle Vorrangstrategien umgesetzt werden.
- 2) Eine Obergruppe ist genau dann eine eigenständige Entität, wenn ihr eine ID zugeordnet wird. Wird keine ID definiert, so "berechnet" sich die ID der Gruppe als die Vereinigungsmenge aller IDs der Untergruppen.
- 3) Veränderungen im Gruppengeflecht werden über ID-Erhaltungen und über Benachrichtigungsmechanismen abgedeckt.



11.3.5.3 Hierarchische Semantik (Verwaltungen)

Die Rechtedefinition ist eng verknüpft mit der Semantik der Verwaltungen. Es gibt zwei Möglichkeiten:

1. keine hierarchische Semantik, ausgenommen Vererbung (z. B. Medien-Verwaltung/Inhalte-Verwaltung)
2. Vererbungs- und Einschränkungshierarchie (z. B. Navigation über die Struktur-Verwaltung)

Im ersten Fall definiert der Verwaltungsbaum keine hierarchische Semantik außer der Vererbung (z. B. Medien-Verwaltung /Inhalte-Verwaltung). In diesem Fall können vielfältige Anomalien auftreten, beispielsweise:

- ein Benutzer hat zwar die erforderlichen Berechtigungen für ein Dokument, kann aber nicht auf das Dokument zugreifen, da er keine Berechtigung für die Seite besitzt, auf der das Dokument verlinkt ist.
- Umgekehrt kann ein Verweis auch auf ein Dokument zeigen, für das der Benutzer keine Berechtigungen hat.

Beide Fälle sind eigentlich Fehlkonfigurationen, allerdings sind diese schwer zu erkennen und damit kaum zu vermeiden.

Im zweiten Fall definiert die Verwaltung neben der Vererbungshierarchie eine Einschränkungshierarchie. Damit kann die Menge der berechtigten Personen entlang des Baumes nur eingeschränkt, aber nie erweitert werden. Dies macht Sinn, wenn die Baum-Hierarchie sich in Form eines hierarchischen Menüs widerspiegelt, da in diesem Fall der Zugriff auf ein "tieferes Element im Baum" nur über den "darüberliegenden Knoten" erfolgen kann. Damit ist eine Erweiterung der Berechtigungen sinnlos, da der darüber liegende Einstiegspunkt fehlt.

Zusammenfassend kann gesagt werden, dass das Verhalten im zweiten Fall in einigen Fällen die Möglichkeiten zur Fehlkonfiguration reduzieren kann, allerdings steht dem gegenüber eine erhöhte Anforderung bei der Konfigurationsvisualisierung.

Die FirstSpirit Implementierung bietet die notwendige Infrastruktur, um beide Varianten realisieren zu können.



11.3.6 Definitionen

Für die Festlegung von Benutzerberechtigungen gelten die folgenden Definitionen:

1. Die Verwaltungsstruktur hat ausschließlich eine hierarchische Semantik für die Vererbung. Eine Einschränkung der Rechtekonfiguration entlang des Verwaltungsbaums wird nicht vorgesehen. Es wurden aber Mechanismen geschaffen, die eine Validierung der Rechtedefinition mit projektspezifischer Semantik erlauben (z. B. Script-Hooks für den Button "Prüfen" oder "on-load-/on-save"-Skripte).
2. Eine modifizierende Vererbung wird grundsätzlich ausgeschlossen. Jede Rechtedefinition hat den Charakter eines Rechtedefinitionspunktes (d.h. die Rechte werden kopiert und dann modifiziert). Um dieses Konzept in der Praxis einsetzen zu können, werden einige zusätzliche Hilfsfunktionen realisiert (siehe Convenience-Methoden Kapitel 11.6.2.1 auf Seite 544).
3. Es werden im Gruppenbaum folgende drei Zustände definiert:
 -  Erlaubt = grünes Häkchen
 -  Verboten = rotes Kreuz
 -   wie der Vaterknoten = entweder graues Häkchen oder graues Kreuz.

Durch Klicken wird durch die drei Zustände geschaltet. Dabei werden gegebenenfalls auch die grauen Symbole im darunter liegenden Teilbaum geändert.

4. Abhängigkeitspropagation: Default-Strategie (grau), das heißt die Kinder haben per Default den gleichen Zustand wie der Vater, es kann aber eine explizite Veränderung vorgenommen werden.

Insgesamt führen diese Festlegungen zu einem für den Benutzer überschaubaren Verhalten der Rechtevergabe-Komponenten. Ursache und Wirkung sind vorhersehbar, da der gesamte Zustand der Konfiguration auf einen Blick zu sehen ist. Die Einschränkungen hinsichtlich der Flexibilität, speziell bei der Pflege, werden durch eine Reihe von Convenience-Methoden reduziert.



11.4 Konfiguration

11.4.1 Einleitung

Die Rechtedefinitions-Komponente besteht aus einer Eingabekomponente, die im Client angezeigt wird und vom Redakteur zur Definition der Rechte verwendet wird. Die Eingabekomponente ermittelt die für das konkrete Projekt definierte Gruppenkonfiguration, indem sie mit einer FirstSpirit-Serverkomponente kommuniziert. Die Serverkomponente übernimmt den Aufbau der Gruppenhierarchie (z. B. auf Basis eines LDAP-Baums) und übermittelt die Daten an den FirstSpirit-Client.

11.4.2 Aktivierung des Permission Service

Bei der Rechtedefinitions-Komponente und dem Permission Service handelt es sich um ein FirstSpirit-System-Modul, welches bereits bei der Installation eines FirstSpirit-Servers enthalten ist und nicht separat installiert werden muss.

Starten und Stoppen des Permission-Service erfolgen entweder über das FirstSpirit Server-Monitoring (siehe Kapitel 8.6.2.4 Seite 463) oder über die Server- und Projektkonfiguration (siehe Kapitel 7.3.11 Seite 254).

11.4.3 Konfiguration der Server-Komponente

Die Konfiguration der Server-Komponente erfolgt durch eine Servicekonfigurationsdatei `service.ini`, in der die allgemeinen Einstellungen vorgenommen und optional einer Menge von Gruppenspezifischen-Dateien im XML-Format (siehe Kapitel 11.4.3.1) angegeben werden. Das Servicekonfigurationsverzeichnis für Module befindet sich unterhalb des FirstSpirit-Verzeichnisses im Unterverzeichnis "conf/modules". Die einzelnen Servicekonfigurationsdateien befinden sich in einem Unterverzeichnis, welches den Namen der Komponente (hier: `System.PermissionService`) trägt. Für die Rechtedefinitions-Komponente befindet sich die Datei `services.ini` in folgendem Verzeichnis:

```
conf/modules/System.PermissionService
```

Die Datei `service.ini` muss so erweitert werden, dass es möglich ist, auf Basis von LDAP-Queries eine Gruppenhierarchie aufzubauen. Da dies vermutlich nicht durch eine statische Parametrisierung möglich ist, muss gegebenenfalls ein "erzeugendes BeanShell-Skript" als Parameter angegeben werden. Diesem Skript



kann dann ein geeigneter initialer LDAP-Kontext zur Verfügung gestellt werden.

11.4.3.1 Aufbau der Servicekonfigurationsdatei `service.ini`

Die `service.ini`-Datei konfiguriert den Permission-Service. Die zentrale Aufgabe des Permission-Service ist es, Gruppenshierarchien an die jeweiligen Eingabekomponenten in FirstSpirit zu liefern. Dabei können unterschiedliche Gruppenshierarchien definiert werden, die anhand eines eindeutigen Namens identifiziert werden können. So ist es möglich, dass jedes Projekt eine eigene Gruppenshierarchie definiert oder mehrere Gruppenshierarchien in einem Projekt verwendet werden können.

Grundsätzlich können die Gruppenshierarchien aus zwei unterschiedlichen Quellen kommen:

- 1) Aus einer explizit angegebenen XML-Datei (die manuell oder extern automatisiert erstellt wird). Manuell können diese XML-Dateien über das FirstSpirit Server-Monitoring (siehe Kapitel 8.6.1.7 Seite 455) erzeugt werden.
- 2) Über ein Skript: Automatisch können die Gruppenshierarchien über ein Connector-Skript auf Basis eines bestehenden Benutzer/Gruppen-Verwaltungssystems (z. B. LDAP oder Active-Directory) erzeugt werden.

Im zweiten Fall wird für das Zwischenspeichern der Ergebnisse ebenfalls eine XML-Datei erzeugt, allerdings vom Skript. Das Skript wird vom Service in definierten Intervallen aufgerufen und kann die XML-Datei ggf. verändern. Ein typischer Einsatzfall für ein Skript ist die Erzeugung einer Gruppensdatei aus einem LDAP-Server.

Folgende globale Parameter sind in der INI-Datei enthalten:

`interval` = Zeitraum in Sekunden, in dem die ini-Datei auf Änderungen geprüft wird.

`documents` = kommaseparierte Liste der symbolischen Namen der verfügbaren Gruppenshierarchien.

`NAME.path` = Pfad zur Gruppen-XML-Datei (siehe auch Kapitel 11.4.3.2 Seite 536).
Dabei ist `NAME` ein Platzhalter für einen Wert von `documents`.

Soll die Gruppenshierarchie über ein Skript erzeugt werden, so ist eine Reihe von Parametern notwendig:

`NAME.path` = Pfad zur Gruppen-XML-Datei (siehe auch Kapitel 11.4.3.2 Seite 536)

`NAME.users` = Pfad zur User-XML-Datei (siehe auch Kapitel 11.4.3.3 Seite 538)



`NAME.script` = Pfad zum BeanShell-Skript (z. B. für die automatische Erzeugung von Benutzer- und Gruppendateien – `users.xml` und `group.xml`).

`NAME.script.intervall` = Intervall in Sekunden, in denen das Skript aufgerufen werden soll.

(Bei den oben genannten `NAME.`-Parametern handelt es sich um mögliche Parameter zu jedem `documents`-Eintrag.)

Wenn eine Gruppenhierarchie aus einer XML-Datei erzeugt werden soll, wird nur der Parameter `NAME.path` benötigt.

Standardkonfiguration der Datei `service.ini`:

```
## global params
# -----
# check each x seconds for changes
interval=20

# symbolic names for documents
documents=GroupsFile

#
# document specific params
# -----
GroupsFile.path=groups.xml
```

Weiterhin können Parameter für die LDAP-Verbindung angegeben werden:

`NAME.ldap.url` = URL des LDAP-Servers

`NAME.ldap.userDN` = Login für den LDAP-Lookup

`NAME.ldap.password` = Passwort für den LDAP-Lookup

`NAME.ldap.version` = 2 (LDAP-Protokoll-Version)

`NAME.ldap.ssl` = 0|1 legt fest, ob die LDAP-Verbindung über SSL hergestellt wird.

Aus den hier angegebenen LDAP-Parametern wird ein LDAP-Kontext (`javax.naming.directory.*`) erzeugt, der dem Skript zur Verfügung gestellt wird.

Exemplarisches Beispiel einer "service.ini" mit LDAP-Konfiguration:

```
# global params
# -----
# check each x seconds for changes
interval=20

# symbolic names for documents
documents=GruppenFile, GruppenLdap

# document specific params
# -----
GruppenFile.path=groups.xml
GruppenFile.users=users.xml
GruppenLdap.path=gruppenl.xml
GruppenLdap.script=gruppenl.bsh
```



```
GruppenLdap.ldap.URL=ldap://osiris:389/o=e-Spirit
# optional attributes
#GruppenLdap.script.interval=60
GruppenLdap.ldap.userDN=cn=extern1,cn=Recipients,ou=E-SPIRIT,o=e-Spirit
GruppenLdap.ldap.password=geheim
GruppenLdap.ldap.version=2
GruppenLdap.ldap.SSL=0
```

11.4.3.2 Gruppen-XML-Datei (groups.xml)

Standardmäßig steht die Gruppendatei `groups.xml` für die Definition von Gruppen zur Verfügung. Sie muss in der Permission-Servicekonfigurationsdatei `service.ini` mit dem Parameter `NAME.path` bekannt gemacht werden (siehe Kapitel 11.4.3.1 Seite 534) und kann über das FirstSpirit Server-Monitoring bearbeitet werden (siehe Kapitel 8.6.1.7 Seite 455, "Konfiguration eines Dienstes", Punkt "Dateiname").

Es wird empfohlen, die Datei über ein Skript zu erstellen, um die Fehleranfälligkeit zu reduzieren.

Die Standardkonfiguration der `groups.xml` lautet wie folgt:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<GROUPS name="GroupsFile" version="1">

    <GROUP id="2" name="Anonyme Besucher"/>

    <GROUP id="3" name="Registrierte Mitglieder"/>

    <GROUP id="4" name="Kunden"/>

</GROUPS>
```

Dem Wurzelement der Gruppendefinition `GROUPS` kann im Attribut `name` der Name der Gruppendefinition übergeben werden. Zusätzlich kann das optionale Attribut `version` angegeben werden, um die jeweilige Versionsnummer der Gruppendefinition festzuhalten.

Innerhalb von `GROUPS` können beliebig viele und beliebig ineinander geschachtelte `GROUP`-Elemente angegeben werden, wobei es sich bei jedem `GROUP`-Element um eine Gruppe handelt. Jedes `GROUP`-Element benötigt das Attribut `name` (Name der Gruppe). Zusätzlich kann das optionale Attribut `id` (ID der Gruppe) angegeben werden. Diese `id` wird benötigt, um einer Gruppe in der Eingabekomponente Rechte zuzuordnen zu können. Besitzt eine Gruppe keine `id`, wird sie in der Eingabekomponente ohne einen Knoten angezeigt, auf dem die Rechte gesetzt



werden.

Beispielkonfiguration ohne das Attribut `id` für die Gruppe "Kunden":

```
<?xml version="1.0" encoding="ISO-8859-1" ?>

  <GROUPS name="GroupsFile" version="1">

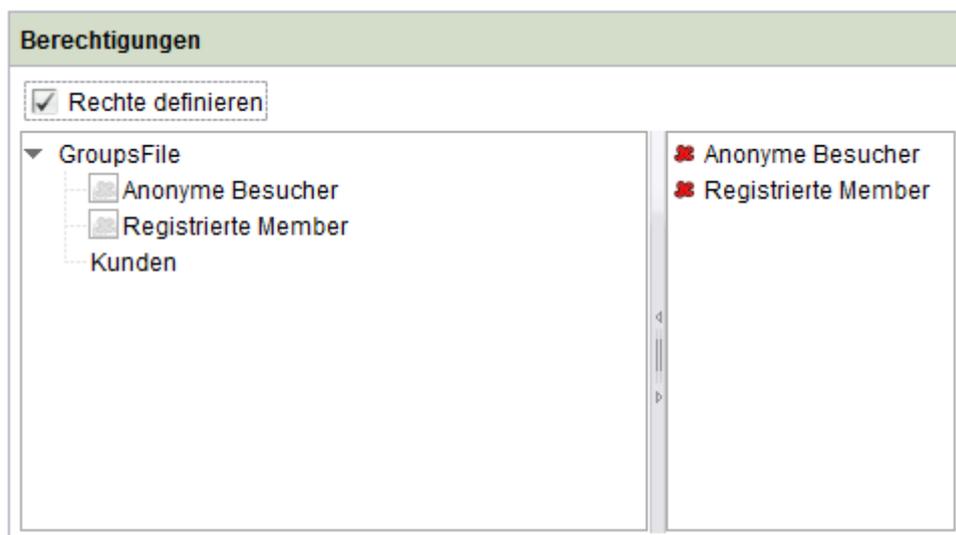
    <GROUP id="2" name="Anonyme Besucher"/>

    <GROUP id="3" name="Registrierte Member"/>

    <GROUP name="Kunden"/>

  </GROUPS>
```

Darstellung im JavaClient:



Mit dem Parameter `selectable` kann festgelegt werden, ob eine Gruppe in der Eingabekomponente auswählbar sein soll oder nicht. Als Werte kann 1 (auswählbar) oder 0 (nicht auswählbar) übergeben werden. Ist keine ID mithilfe des Parameters `id` vergeben worden, so ist die Angabe von `selectable="1"` nicht erlaubt.

Für `GROUPS`- und `GROUP`-Elemente kann zusätzlich das Attribut `timestamp` (Zeitstempel der Erstellung) angegeben werden.



11.4.3.3 Benutzer-XML-Datei (users.xml)

Für die Benutzerdefinition muss eine eigene XML-Datei angelegt werden. Dies kann über das FirstSpirit Server-Monitoring vorgenommen werden (siehe Kapitel 8.6.1.7 Seite 455, "Konfiguration eines Dienstes", Schaltfläche "Datei erstellen"). Diese Benutzer-XML-Datei muss in der Permission-Servicekonfigurationsdatei `service.ini` mit dem Parameter `NAME.users` bekannt gemacht werden (siehe Kapitel 11.4.3.1 Seite 534). Es wird empfohlen, die Datei über ein Skript zu erstellen, um die Fehleranfälligkeit zu reduzieren.

Eine Beispielkonfiguration einer Benutzerdatei kann folgendermaßen lauten:

```
<?xml version="1.0" encoding="UTF-8"?>
  <USERS>
    <USER login="visitor_1"
          realname="Besucher 1"
          password="password_visitor_1"
          active="1"
          groups="2" />

    <USER login="member_1"
          realname="Mitglied 1"
          password="password_member_1"
          active="1"
          groups="3" />

    <USER login="client_1"
          realname="Kunde 1"
          password="password_client_1"
          active="1"
          groups="3,4" />
  </USERS>
```

Dem Wurzelement der Benutzerdefinition `USERS` kann das Attribut `timestamp` (Zeitstempel der Benutzerdefinition) übergeben werden.

Innerhalb von `USERS` können beliebig viele `USER`-Elemente angegeben werden, wobei es sich bei jedem `USER`-Element um einen Benutzer handelt. Jedes `USER`-Element benötigt folgende Attribute:

- `login` (Benutzer-Login)
- `realname` (vollständiger Benutzername)
- `password` (Benutzerpasswort)
- `active` (aktiver/nicht aktiver Benutzer, mögliche Werte: 1 und 0) und
- `groups` (kommaseparierte Liste der Gruppen-IDs, zu der der Benutzer gehört).

Zusätzlich kann das optionale Attribut `timestamp` (Zeitstempel) angegeben werden.



11.4.4 Konfiguration für die Veröffentlichung

Die Veröffentlichung der benötigten Dateien wird über ein Veröffentlichungs-Servlet realisiert (siehe Kapitel 7.5.10.6.3 Seite 399). Dazu wird in der Server- und Projektkonfiguration in den Projekteigenschaften ein Veröffentlichungsauftrag für das Projekt angelegt (siehe Kapitel 7.5.4 Seite 360).

Das Veröffentlichungs-Servlet ist ein fester Bestandteil der Webanwendung im Webserver. Durch dieses Servlet ist es möglich, bei einer Veröffentlichung, die lokale Dateiliste mit der Webanwendung abzugleichen.

Durch die CRC-32-Prüfsummenberechnung für jede generierte Datei können:

- nur geänderte und neue Dateien übertragen werden.
- nicht mehr erzeugte Dateien in der Webanwendung gelöscht werden (optional).

So wird sichergestellt, dass alle aktuellen Daten erzeugt, aber nur veränderte Daten auch übertragen werden. Der Dateiabgleich erfolgt auf Basis der berechneten CRC-32-Prüfsummen.

Für weitere Informationen siehe Modul-Dokumentation FirstSpirit Security.

11.4.5 Konfiguration der Berechtigungen über die Projekteigenschaften

In der Server- und Projektkonfiguration können unter dem Menüpunkt "Berechtigungen" in den Projekteigenschaften die projektspezifischen Einstellungen für die Auswertung der Berechtigung vorgenommen werden (siehe Kapitel 7.4.16 Seite 323).



11.5 Einsatz im Live-System

11.5.1 Konfiguration des Servlet-Servers

Nachdem alle benötigten Dateien und die veröffentlichten Dateien auf dem Live-System vorhanden sind, müssen noch die Konfigurationsdateien des Servlet-Servers (z. B. Jetty, Tomcat usw.) angepasst werden.

Für die Konfiguration ist es wichtig, dass sich das Verzeichnis "WEB-INF" entweder:

1. Im "Zielverzeichnis" aus den Veröffentlichungseinstellungen befindet (vgl. Kapitel 7.5.10.6, Seite 393 ff.). Dann ist das "Zielverzeichnis" das Web-Applikationsverzeichnis, welches gemappt werden muss:

Jetty:

```
<Call name="addWebApplication">
  <Arg>/</Arg>
  <Arg>ZIELVERZEICHNIS</Arg>
  <Set name="extractWAR">>false</Set>
</Call>
```

Tomcat:

```
<Context path="" docBase="ZIELVERZEICHNIS" />
```

Oder:

2. Neben dem "Zielverzeichnis" aus den Veröffentlichungseinstellungen befindet. Im zweiten Fall muss das "Projektverzeichnis" aus den Veröffentlichungseinstellungen gemappt werden.

Jetty:

```
<Call name="addWebApplication">
  <Arg>/</Arg>
  <Arg>PROJEKTVERZEICHNIS</Arg>
  <Set name="extractWAR">>false</Set>
</Call>
```

Tomcat:

```
<Context path="" docBase="PROJEKTVERZEICHNIS" />
```



11.6 Einsatz im Projekt

11.6.1 Manuelle Gruppendifinition und ID-Vergabe

Die Implementierung der Permission-Infrastruktur geht davon aus, dass jeder Knoten im Gruppenbaum eindeutig identifiziert werden kann oder keine Bedeutung hat. Dazu dient das ID-Attribut in der "groups.xml"-Datei, die vom Server bereitgestellt wird.

Grundsätzlich kann jeder Gruppe eine eigene, eindeutige ID zugeordnet werden. Diese Zuordnung muss erfolgen, wenn eine Gruppe keine weiteren Untergruppen hat (d.h. jedes Blatt muss eine ID haben). Sollte einer Gruppe keine ID zugeordnet werden, so definiert sich die Gruppen-ID implizit aus der Vereinigungsmenge aller Untergruppen-IDs (transitiv).

Die Semantik einer ID-Vergabe besteht darin, dass Personen nur in Gruppen mit einer ID vorhanden sein können. Wenn also eine Gruppe nur Untergruppen und keine Personen enthält, so muss keine ID vergeben werden. Eine solche Gruppe dient nur der Veranschaulichung. Wenn mindestens eine Person in einer Gruppe enthalten ist, so muss eine ID vergeben werden. Bei der ID-Vergabe sollte sichergestellt werden, dass die IDs mit den im Rahmen der Personalisierung ermittelten Gruppen-IDs übereinstimmen. Dabei ist besonders zu beachten, dass die Gruppenhierarchie im Personalisierungs-Modul nicht aufgelöst wird, da die Normalisierung bereits von der Rechtedefinitions-Komponente durchgeführt wird!

Bei der Auswertung der Rechtekonfiguration wird die zusammengesetzte ID zurückgeliefert.



Beispiel:

```
Alle (ID=, eff:1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
+--group 1 (ID=, eff: 7, 10)
+---+--group 1.1 (ID 7)
+---+-- group 1.2 (ID 10)
+--group 2 (ID=9, eff:1, 2, 3, 4, 5, 6,9)
+---+--group 2.1 (ID 4, eff: 1, 2, 3, 4)
+---+---+--group 2.1.1(ID 1)
+---+---+--group 2.1.2(ID 2)
+---+---+--group 2.1.3(ID 3)
+---+--group 2.2 (ID 6, eff: 5,6)
+-----+--group 2.2.1 (ID 5)
+--group 3 (ID=8)
```

In diesem Beispiel hat "group 1" keine explizite ID, daher ergibt sich die ID aus der Vereinigungsmenge aller IDs der Untergruppen. In diesem Fall sind das "7" und "10". Wird also auf der Ebene von "group 1" eine Konfiguration zurückgeliefert, so wird als Ergebnis "7, 10" zurückgeliefert. Logisch bedeutet dies, dass in "group 1" keine Personen enthalten sein können, sondern nur in "group 1.1 (7)" und "group 1.2 (10)".

Im Fall von "group 2" ist dies anders. Diese Gruppe hat die ID 9 und zusätzlich diverse Untergruppen. Es gibt damit Personen, die direkt in "group 2" sind oder in den Untergruppen.



11.6.2 Konfigurationssemantik

Die Implikationen einer Berechtigungskonfiguration in Verbindung mit dem gewählten ID-Schema ist nicht in allen Aspekten trivial und wird im FirstSpirit Handbuch für Redakteure anhand eines Beispiels ausführlicher erläutert.

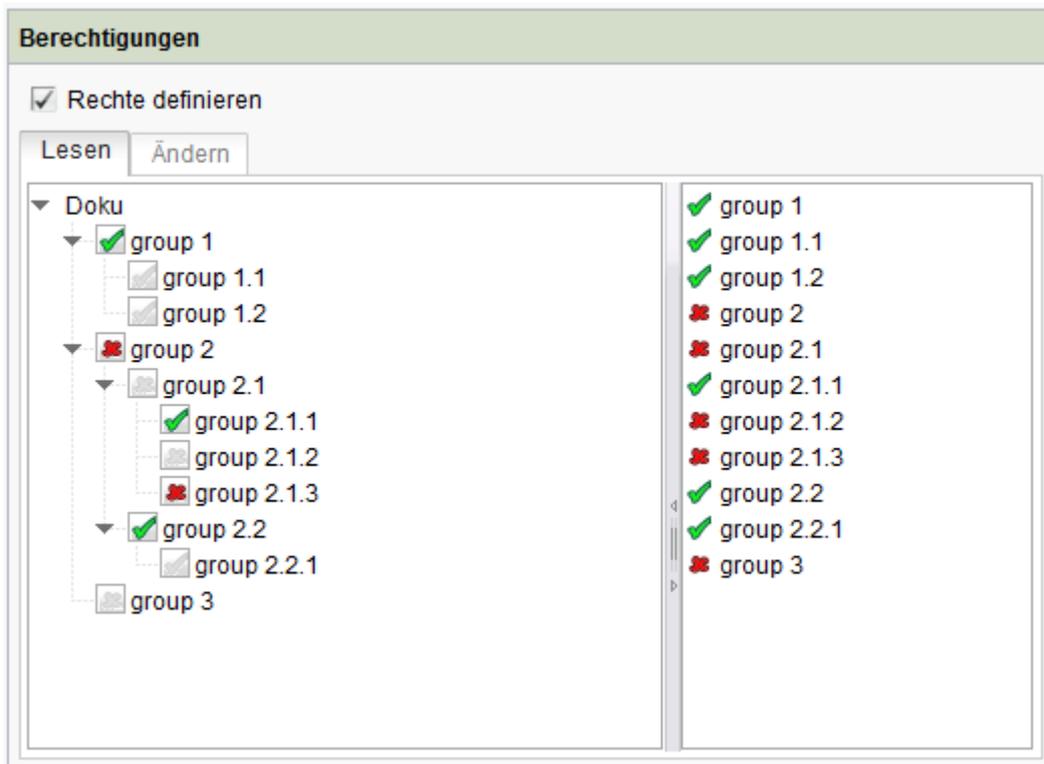


Abbildung 11-4: Beispiel Rechtedefinitions-Komponente

Die Rechtedefinitions-Komponente ist in der Lage, Rechte für unterschiedliche Operationen zu vergeben. Die Operationen sind grundsätzlich orthogonal, das heißt, es gibt kein Szenario "Operation A erfordert B" oder "aus A folgt B". Sollen solche Funktionen realisiert werden, so erfolgt dies über projektspezifische Validations/Korrektur-Skripte, die zu geeigneten Zeitpunkten aufgerufen werden (siehe Beschreibung im "FirstSpirit Handbuch für Entwickler").

Die Darstellung der zusätzlichen Operationen erfolgt in der Form von Registern (siehe Abbildung 11-4 "Lesen" und "Ändern").



11.6.2.1 Convenience-Methoden

Offensichtlich ist die Funktion "ererbte Rechte kopieren" eine sinnvolle Vorbelegung, wenn die Rechte in einem Knoten konfiguriert werden sollen.

Die Entscheidung eine "modifizierende Vererbung" nicht zu unterstützen, führt zu folgendem Problem bei der "Wartung" von Projekten:

Änderungen von Rechtedefinitionen: Änderungen von Rechten-Definitionen haben keinen Einfluss auf in der Baumstruktur hierarchisch untergeordnete Teilbäume, die mit einem Rechtedefinitions-Knoten beginnen ("Copy-on-Change"-Semantik des Rechtedefinitions-Punktes). Somit müssen diese Klasse von Änderungen auf jeden Rechtedefinitions-Knoten in darunter liegenden Teilbäumen angewendet werden. Um dies zu erleichtern, kann bei jeder Konfigurationsänderung über ein Kontextmenü auf einem Gruppenknoten der Zustand auf die untergeordneten Knoten übertragen werden. Nachdem das Kontextmenü "Rechte propagieren" gewählt wurde, erscheint ein Fenster, in dem eine Liste aller untergeordneten Rechtedefinitionspunkte angezeigt wird. In diesem, können die entsprechenden Knoten ausgewählt werden und nach Bestätigung wird der Zustand des Teilbaumes, auf dem das Kontextmenü geöffnet wurde, an alle Unterknoten übertragen.



12 Anhang

12.1 fs-wrapper.conf



Innerhalb der Konfigurationsdatei fs-wrapper.conf dürfen Kommentare nicht in einer Zeile mit Parameter-Werten stehen, z. B.:

```
wrapper.startup.timeout=30 # Kommentar
```

```
*****
# FirstSpirit-Server Java-Wrapper Properties
# Documentation available in FirstSpirit Administration Manual
# and http://wrapper.tanukisoftware.com/doc/english/properties.html
# *****

# Java command (JDK needed, not JRE)
# absolute path or just "java" when environment variable PATH is set
# correctly
wrapper.java.command=C:\Program Files (x86)\Java\jdk1.7.0_05\bin\java.exe

# Maximum Java Heap Size
# set in MByte with maxmemory=MBYTESNUMBER
# set in percent of total physical RAM with
# maxmemory.percent=PERCENTNUMBER
wrapper.java.maxmemory=700

# Initial Java Heap Size.
# set to same value as wrapper.java.maxmemory to prevent unnecessary Full
GCs
wrapper.java.initmemory=700

# Java parameters.
# Gaps in parameter enumeration are allowed since FirstSpirit 5 /
# Wrapper 3.3.6.
wrapper.java.additional.1=-Djava.awt.headless=true
wrapper.java.additional.2=-Djava.security.auth.login.config=conf/fs-
jaas.conf
wrapper.java.additional.3=-Djava.security.policy=conf/fs-server.policy
wrapper.java.additional.4=-Dfile.encoding=UTF-8
wrapper.java.additional.5=-Xshare:off
wrapper.java.additional.6=-Djava.net.preferIPv4Stack=true
wrapper.java.additional.7=-Djava.io.tmpdir=work
wrapper.java.additional.8=
wrapper.java.additional.9=
wrapper.java.additional.10=

# parameter is automatically set by 64bit JVM
# explicitly set as needed by Berkeley-DB:
wrapper.java.additional.11=#-XX:+UseCompressedOops

# Java parameters for garbage collection
# set -Xmn to 40% of wrapper.java.maxmemory
wrapper.java.additional.12=-Xmn280M
wrapper.java.additional.13=-XX:PermSize=200M
```



```
wrapper.java.additional.14=-XX:MaxPermSize=200M
wrapper.java.additional.15=-XX:InitialCodeCacheSize=128M
wrapper.java.additional.16=-XX:ReservedCodeCacheSize=128M
wrapper.java.additional.17=-XX:SurvivorRatio=1
wrapper.java.additional.18=-XX:SoftRefLRUPolicyMSPerMB=1
wrapper.java.additional.19=-XX:+DisableExplicitGC
wrapper.java.additional.20=-XX:+UseConcMarkSweepGC
wrapper.java.additional.21=-XX:+UseParNewGC
wrapper.java.additional.22=-XX:+CMSParallelRemarkEnabled
wrapper.java.additional.23=-XX:+CMSClassUnloadingEnabled
wrapper.java.additional.24=-XX:+NeverTenure
wrapper.java.additional.25=-XX:-UseLargePages
wrapper.java.additional.26=-Djava.rmi.dgc.leaseValue=3600000
wrapper.java.additional.27=
wrapper.java.additional.28=
wrapper.java.additional.29=
wrapper.java.additional.30=
wrapper.java.additional.31=
wrapper.java.additional.32=
wrapper.java.additional.33=
wrapper.java.additional.34=
wrapper.java.additional.35=

# Garbage Collector Log
# Logfile fs-gc.log is automatically rotated by FirstSpirit.
wrapper.java.additional.36=-verbose:gc
wrapper.java.additional.37=-XX:+PrintGCTimeStamps
wrapper.java.additional.38=-XX:+PrintGCDetails
wrapper.java.additional.39=-XX:+PrintGCDateStamps
wrapper.java.additional.40=-Xloggc:log/fs-gc.log

# Enable JMX-Connector via fs-server.conf, parameter jmx.port.
# For more JMX parameters see FirstSpirit Administration Manual.

# Timeout parameters in seconds for controlling the Java process.
# Before changing read Wrapper documentation as all timeout parameters
# are related to each other.
# ping.timeout disabled to allow enough time for large heapdumps,
# can be set to 300 if automatic restarts in case of JVM deadlock
# is needed.
wrapper.startup.timeout=30
wrapper.shutdown.timeout=180
wrapper.jvm_exit.timeout=30
wrapper.cpu.timeout=20
wrapper.ping.timeout=0
wrapper.timer_slow_threshold=3
wrapper.successful_invocation_time=35

# Unix: umask for creating files, to prevent access by others use 0027
wrapper.umask=0022

# set TRUE on Solaris with SMF, FALSE on all other systems
wrapper.disable_restarts=FALSE

# disable automatic restarts after Java-VM failures
wrapper.disable_restarts.automatic=TRUE

*****
****
# stdout/stderr-logging (log/fs-wrapper.log)
# for configuration of log/fs-server.log edit conf/fs-logging.conf
*****
****

# Log Level for log file output. (DEBUG, INFO, STATUS, ERROR, FATAL,
```



```
NONE)
wrapper.logfile.loglevel=INFO

# Log Level for console mode. (DEBUG, INFO, STATUS, ERROR, FATAL, NONE)
wrapper.console.loglevel=INFO

# Show Java options in fs-wrapper.log
wrapper.java.command.loglevel=INFO

# Format of output for the console. (See docs for formats)
wrapper.console.format=PM

# Log file to use for wrapper output logging.
wrapper.logfile=log/fs-wrapper.log

# Format of output for the log file. (See docs for formats)
wrapper.logfile.format=LPTM

# Maximum size that the log file will be allowed to grow to before
# the log is rolled. Size is specified in bytes. The default value
# of 0, disables log rolling. May abbreviate with the 'k' (kb) or
# 'm' (mb) suffix. For example: 10m = 10 megabytes.
wrapper.logfile.maxsize=10m

# Maximum number of rolled log files which will be allowed before old
# files are deleted. The default value of 0 implies no limit.
wrapper.logfile.maxfiles=9

# Unix: Log Level for syslog (DEBUG, INFO, STATUS, ERROR, FATAL, NONE)
wrapper.syslog.loglevel=NONE

# Unix: Log Level for syslog/event log output. (USER, LOCAL0-7)
wrapper.syslog.facility=USER

# Unix: Identity entry for syslog
wrapper.syslog.ident=FirstSpirit

*****
****
# Windows Service
*****
****

# WARNING - Do not modify any of these properties when an application
# using this configuration file has been installed as a service.
# Please uninstall the service before modifying this section. The
# service can then be reinstalled by starting the FIRSTspirit Installer.

# Name of the service
wrapper.ntservice.name=FirstSpiritServer5 Instance 2

# Display name of the service
wrapper.ntservice.displayname=FirstSpirit 5.0 Instance 2

# Description of the service
wrapper.ntservice.description=FirstSpirit Content Management Server

# Service dependencies. Add dependencies as needed starting from 1
wrapper.ntservice.dependency.1=

# Mode in which the service is installed. AUTO_START or DEMAND_START
wrapper.ntservice.starttype=AUTO_START

# Allow the service to interact with the desktop.
wrapper.ntservice.interactive=false
```



```
# Use a console window. This is needed as workaround for Threaddumps
# on Windows. The window will only be visible if
# wrapper.ntservice.interactive=true
wrapper.ntservice.console=true

#*****
****
# Do not change parameters below as FirstSpirit depends on them
#*****
****

# Directory to launch FirstSpirit-Server relative to wrapper(.exe)
wrapper.working.dir=../

# Application parameters. Add parameters as needed starting from 1
#wrapper.app.parameter.1=

# Java Classpath (includes wrapper.jar)
wrapper.java.classpath.1=shared/classes
wrapper.java.classpath.2=shared/lib/*.jar
wrapper.java.classpath.3=shared/lib/*.zip
wrapper.java.classpath.4=server/classes
wrapper.java.classpath.5=server/lib/*.jar

# Java Library Path (includes libwrapper.so or wrapper.dll)
wrapper.java.library.path.1=shared/lib
wrapper.java.library.path.2=server/lib

# FirstSpirit-Server Main class
wrapper.java.mainclass=de.espirit.firstspirit.server.CMSServer

# Java Service Wrapper is licensed for redistribution by e-Spirit AG
#include ../conf/fs-wrapper-license.conf

# Do not restart Java VM on failure
wrapper.max_failed_invocations=1

# only one instance of this server with name wrapper.ntservice.name is
# allowed
wrapper.single_invocation=TRUE

# write thread dump if server failed to exit
wrapper.request_thread_dump_on_failed_jvm_exit=TRUE

# return codes used by Webmonitor
wrapper.on_exit.default=SHUTDOWN
# normal restart:
wrapper.on_exit.23=RESTART
# web-update and restart:
wrapper.on_exit.42=RESTART

# reload this file on Wrapper restart
wrapper.restart.reload_configuration=TRUE

# File which will be monitored every 5 seconds for wrapper commands
wrapper.commandfile=conf/fs-control
wrapper.command.poll_interval=5

# Title to use when running in console mode
wrapper.console.title=FirstSpirit

# no continuous enumeration is needed for wrapper.java.additional.<n>
# and other numbered parameters!
wrapper.ignore_sequence_gaps=true
```



```
# set -d64 automatically on operating systems where required
wrapper.java.additional.auto_bits=true

# Server update include file
#include ../server/update/fs-update.conf

*****
****
# end of file
*****
****
```

12.2 fs-jaas.conf

```
/*
 * JAAS Login Configurations.
 * (for app-to-config mappings see fs-server.conf, JAAS.*)
 */

/* access api authentication (e.g., for remote projects) */
system {
de.espirit.firstspirit.server.authentication.FSUserLoginModule sufficient
hash="true";
de.espirit.firstspirit.server.authentication.FSTicketLoginModule
sufficient;
};

/* JavaClient without Webstart, without SSO */
plain {
// de.espirit.firstspirit.server.authentication.LdapLoginModule optional
section="LDAP";
de.espirit.firstspirit.server.authentication.FSUserLoginModule optional;
};

/* JavaClient without Webstart, with SSO */
sso {
de.espirit.firstspirit.server.authentication.FSTicketLoginModule
sufficient;
// de.espirit.firstspirit.server.authentication.LdapLoginModule optional
section="LDAP";
de.espirit.firstspirit.server.authentication.FSUserLoginModule optional;
};

/* FirstSpirit start page without SSO: WebClient, JavaClient with
Webstart */
webplain {
// de.espirit.firstspirit.server.authentication.LdapLoginModule optional
section="LDAP";
de.espirit.firstspirit.server.authentication.FSUserLoginModule optional;
};

/* FirstSpirit start page with SSO: WebClient, JavaClient with Webstart
*/
websso {
de.espirit.firstspirit.server.authentication.FSTicketLoginModule
sufficient;
// de.espirit.firstspirit.server.authentication.KerberosLoginModule
optional useFullPrincipal="false";
// de.espirit.firstspirit.server.authentication.LdapLoginModule optional
section="LDAP";
de.espirit.firstspirit.server.authentication.FSUserLoginModule optional;
};
```

```
};  
/* required additional configuration for KerberosLoginModule: */  
/*  
com.sun.security.jgss.accept {  
    com.sun.security.auth.module.Krb5LoginModule required  
        principal=HTTP/fs4host.mydomain.net@MYDOMAIN.NET  
        keyTab="/opt/firstspirit4/conf/fs4host-HTTP.keytab"  
        useKeyTab="true"  
        storeKey="true"  
        isInitiator="false"  
        doNotPrompt="true"  
        debug="false";  
};  
*/
```

12.3 fs-webapp.xml

```
<?xml version="1.0"?>  
  
<!DOCTYPE Configure PUBLIC "-//Jetty//Configure//EN"  
    "http://www.eclipse.org/jetty/configure.dtd">  
  
<!--  
  
Configuration of FirstSpirit Web-Server.  
  
For parameter details see http://wiki.eclipse.org/Jetty/ and FirstSpirit  
Administration Manual.  
  
This file is only used when INTERNAL_SERVLET_ENGINE=1 is set in conf/fs-  
server.conf.  
  
-->  
  
<Configure id="Server" class="org.eclipse.jetty.server.Server">  
  
<!-- Server Thread Pool -->  
<!-- ===== -->  
  
<Set name="ThreadPool">  
    <New class="org.eclipse.jetty.util.thread.QueuedThreadPool">  
        <Set name="minThreads">5</Set>  
        <Set name="maxThreads">250</Set>  
    </New>  
</Set>
```



```

<!-- HTTP-Connector -->
<!-- ===== -->
<Call name="addConnector">
    <Arg>
        <New
class="org.eclipse.jetty.server.nio.SelectChannelConnector">
            <Set
name="port"><SystemProperty name="HTTP_PORT" /></Set>
            <Set
name="maxIdleTime">30000</Set>
            <Set
name="Acceptors">1</Set>
            <Set
name="statsOn">>false</Set>
            <Set
name="lowResourcesConnections">1000</Set>
            <Set
name="lowResourcesMaxIdleTime">500</Set>
        </New>
    </Arg>
</Call>

<!-- HTTPS-Connector -->
<!-- ===== -->
    <!-- if NIO is not available, use
org.eclipse.jetty.server.ssl.SslSocketConnector -->
    <!--
        <New id="sslContextFactory"
class="org.eclipse.jetty.http.ssl.SslContextFactory">
            <Set name="KeyStore"><SystemProperty name="cmsroot" />/conf/fs-
keystore.jks</Set>
            <Set name="KeyStorePassword">changeit</Set>
            <Set name="KeyManagerPassword">changeit</Set>
        </New>

        <Call name="addConnector">
            <Arg>

```



```

        <New class="org.eclipse.jetty.server.ssl.SslSelectChannelConnector">

            <Arg><Ref id="sslContextFactory"/></Arg>

            <Set name="Port">8443</Set>

            <Set name="maxIdleTime">30000</Set>

            <Set name="Acceptors">2</Set>

            <Set name="AcceptQueueSize">100</Set>

        </New>

    </Arg>

</Call>

-->

<!-- Request Log -->
<!-- ===== -->
<!--
<Ref id="Handlers">

    <Call name="addHandler">

        <Arg>

            <New id="RequestLog"
class="org.eclipse.jetty.server.handler.RequestLogHandler">

                <Set name="requestLog">

                    <New id="RequestLogImpl"
class="org.eclipse.jetty.server.NCSARequestLog">

                        <Set name="filename"><Property name="jetty.logs"
default="./logs"/>/yyyy_mm_dd.request.log</Set>

                        <Set name="filenameDateFormat">yyyy_MM_dd</Set>

                        <Set name="retainDays">90</Set>

                        <Set name="append">true</Set>

                        <Set name="extended">>false</Set>

                        <Set name="logCookies">>false</Set>

                        <Set name="LogTimeZone">GMT</Set>

                    </New>

                </Set>

            </New>

        </Set>

    </New>

```



```

        </Arg>

        </Call>

    </Ref>

    -->

    <!-- Set handler Collection Structure -->

    <!-- ===== -->

    <Set name="handler">

        <New id="Handlers"
class="org.eclipse.jetty.server.handler.HandlerCollection">

            <Set name="handlers">

                <Array type="org.eclipse.jetty.server.Handler">

                    <Item>

                        <New id="Contexts"
class="org.eclipse.jetty.server.handler.ContextHandlerCollection"/>

                    </Item>

                    <Item>

                        <New id="DefaultHandler"
class="org.eclipse.jetty.server.handler.DefaultHandler"/>

                    </Item>

                </Array>

            </Set>

        </New>

    </Set>

    <!-- FirstSpirit Web Applications -->

    <!-- ===== -->

    <New
class="de.espirit.firstspirit.server.jetty.JettyManagerImpl$FailSafeWebAppContext">

        <Arg><Ref id="Contexts" /></Arg>

        <Arg><SystemProperty name="WEBAPP_ROOT_PATH" /></Arg>

        <Arg><SystemProperty name="WEBAPP_ROOT_URL" /></Arg>

    </New>

<New

```



```

class="de.espirit.firstspirit.server.jetty.JettyManagerImpl$FailSafeWebAppContext">
    <Arg><Ref id="Contexts" /></Arg>
    <Arg><SystemProperty name="WEBAPP_WEBMON_PATH" /></Arg>
    <Arg><SystemProperty name="WEBAPP_WEBMON_URL" /></Arg>
</New>
<New
class="de.espirit.firstspirit.server.jetty.JettyManagerImpl$FailSafeWebAppContext">
    <Arg><Ref id="Contexts" /></Arg>
    <Arg><SystemProperty name="WEBAPP_WEBEDIT5_PATH" /></Arg>
    <Arg><SystemProperty name="WEBAPP_WEBEDIT5_URL" /></Arg>
</New>
<New
class="de.espirit.firstspirit.server.jetty.JettyManagerImpl$FailSafeWebAppContext">
    <Arg><Ref id="Contexts" /></Arg>
    <Arg><SystemProperty name="WEBAPP_STAGING_PATH" /></Arg>
    <Arg><SystemProperty name="WEBAPP_STAGING_URL" /></Arg>
</New>
<New
class="de.espirit.firstspirit.server.jetty.JettyManagerImpl$FailSafeWebAppContext">
    <Arg><Ref id="Contexts" /></Arg>
    <Arg><SystemProperty name="WEBAPP_PREVIEW_PATH" /></Arg>
    <Arg><SystemProperty name="WEBAPP_PREVIEW_URL" /></Arg>
</New>
</Configure>

```

