



FirstSpirit™

Your Content Integration Platform

Dokumentation für Administratoren

FirstSpirit Version 4.x

| | |
|----------------|---|
| Version | 2.45 |
| Status | RELEASED |
| Datum | 2011-08-12 |
| Abteilung | FS-Core |
| Autor/ Autoren | B. Ehle, S. Höbbel, D.Reinartz, B.Gutknecht |
| Copyright | 2011 e-Spirit AG |
| Dateiname | ADMI4xDE_FirstSpirit_AdminDocumentation |

e-Spirit AG
Barcelonaweg 14
44269 Dortmund | Germany

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

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

e-Spirit^{AG}

Inhaltsverzeichnis

| | | |
|----------|---|-----------|
| 1 | Einführung | 12 |
| 1.1 | Thema dieser Dokumentation | 17 |
| 1.2 | Gliederung der Dokumentation | 18 |
| 1.3 | Die FirstSpirit-Benutzeroberfläche | 19 |
| 2 | Installation | 21 |
| 3 | Steuerung des FirstSpirit-Servers | 22 |
| 3.1 | Unix..... | 22 |
| 3.1.1 | GNU/Linux und Solaris 9..... | 22 |
| 3.1.2 | Solaris 10 | 22 |
| 3.1.3 | AIX..... | 23 |
| 3.1.4 | Unter normalem Benutzerkonto | 23 |
| 3.1.5 | Stackdump erzeugen | 23 |
| 3.2 | Windows..... | 24 |
| 3.2.1 | Registrieren / Deregistrieren als Systemdienst über Startmenü... .. | 24 |
| 3.2.2 | Deinstallieren über das Startmenü..... | 25 |
| 3.2.3 | Starten/Stoppen als Systemdienst über Startmenü..... | 25 |
| 3.2.4 | Starten/Stoppen im Konsolenmodus..... | 26 |
| 3.2.5 | Konfigurationsdateien über Startmenü editieren | 26 |
| 3.2.6 | Weitere Startmenü-Funktionen | 28 |
| 4 | Konfiguration des FirstSpirit-Servers | 30 |
| 4.1 | Dateisystemorganisation..... | 30 |
| 4.1.1 | Dateinamen | 30 |



| | | |
|-------|--|-----|
| 4.1.2 | Webanwendungen | 31 |
| 4.1.3 | Verzeichnisstrukturen..... | 31 |
| 4.2 | Allgemeines zur Konfiguration..... | 32 |
| 4.3 | Konfigurationsdateien (FirstSpirit-Server)..... | 33 |
| 4.3.1 | Konfiguration des FirstSpirit-Servers (fs-server.conf) | 34 |
| 4.3.2 | Konfiguration der Java-VM und des -Wrappers (fs-wrapper.conf)..... | 64 |
| 4.3.3 | Konfiguration der Datenbankanbindung (fs-database.conf)..... | 75 |
| 4.3.4 | Konfiguration des Anmeldevorgangs (fs-jaas.conf) | 76 |
| 4.3.5 | Konfiguration Lizenz (fs-license.conf) | 92 |
| 4.3.6 | Konfiguration Logging (fs-logging.conf)..... | 96 |
| 4.3.7 | Konfiguration des Webservers (fs-webapp.xml) | 99 |
| 4.4 | Anbindung an einen LDAP-Server | 103 |
| 4.4.1 | Authentifizierung über LDAP | 103 |
| 4.4.2 | LDAP-Attribute an einen FirstSpirit Benutzer binden | 104 |
| 4.4.3 | Verwendung von TLS oder SSL | 105 |
| 4.5 | Einbinden in externen Webserver | 106 |
| 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 | Externe Servlet-Engine (Tomcat) | 115 |
| 4.5.4 | Lastverteilung auf mehrere Servlet-Engines | 121 |
| 4.6 | Einbinden in externen Application-Server..... | 123 |
| 4.6.1 | Einbinden in Websphere Application Server..... | 124 |
| 4.6.2 | Anforderungen an externen Application-Server | 127 |
| 4.7 | Konfiguration des HTTPS-Servers..... | 129 |
| 4.7.1 | Sicherheitszertifikat für einen Testserver installieren | 129 |
| 4.7.2 | Vertrauenswürdigen Sicherheitszertifikat installieren | 130 |
| 4.8 | Datenbankanbindung | 133 |



| | | |
|----------|--|------------|
| 4.8.1 | Ablage der JDBC-Treiberdateien | 133 |
| 4.8.2 | Erstellen eines JDBC-Treiber-Moduls..... | 134 |
| 4.8.3 | Installation und Konfiguration des JDBC-Treiber-Moduls | 140 |
| 4.8.4 | Konfiguration von Datenquellen | 147 |
| 4.8.5 | Notwendige Rechte für Datenbank-Benutzerkonten | 154 |
| 4.8.6 | Hinweise und Einschränkungen zu den einzelnen Datenbanksystemen | 156 |
| 4.8.7 | Beispiele für die Anbindung verschiedener Datenbanksysteme. | 158 |
| 4.8.8 | Vorgehen bei der Anbindung externer Datenbanken..... | 163 |
| 4.9 | Ausrollprozess für native Applikationen (ab V4.2) | 165 |
| 4.9.1 | Ausrollprozess (Server) | 165 |
| 4.9.2 | Ausrollprozess (Arbeitsplatzrechner) | 165 |
| 4.9.3 | Aktualisierung der nativen Systemkomponenten | 167 |
| 4.9.4 | Überschreiben von Dateien beim Ausrollprozess unterbinden ... | 168 |
| 5 | Konfiguration der FirstSpirit Webanwendungen | 169 |
| 5.1 | Konfiguration der FirstSpirit Startseite (fs4root)..... | 169 |
| 5.2 | Konfiguration des WebClients | 169 |
| 5.2.1 | Projektvoraussetzungen für den Einsatz von WebEdit..... | 170 |
| 5.2.2 | WebEdit als projektlokale Anwendung (ab V4.1) | 171 |
| 5.2.3 | Browserkonfiguration für den Einsatz von WebEdit..... | 172 |
| 5.2.4 | Aktivierung von Skripten in WebEdit | 173 |
| 5.2.5 | Unterbinden von "Directory-Browsing" | 173 |
| 5.2.6 | WebEdit Baumdarstellung konfigurieren | 174 |
| 5.2.7 | Konfiguration von Arbeitsabläufen in WebEdit | 175 |
| 5.3 | Single-Sign-On (SSO) konfigurieren | 176 |
| 5.3.1 | SSO in Verbindung mit dem JavaClient | 176 |
| 5.3.2 | WebEdit mit Portal-Authentifizierung..... | 177 |



- 5.3.3 Anmeldung mit Klartext-Passwort 177
- 5.3.4 Einschränkungen für den Einsatz von WebEdit 178
- 5.4 Deaktivieren von Popup-Blockern 178

- 6 FirstSpirit Startseite 180**
 - 6.1 Automatische Anmeldung über Single Sign-on 180
 - 6.2 Anmeldung mit Benutzername und Passwort 181
 - 6.3 FirstSpirit-Startseite 182
 - 6.3.1 Überarbeitete Startseite in FirstSpirit Version 4.1 183
 - 6.3.2 Schnellstart 184
 - 6.3.3 Clientstart 184
 - 6.3.4 Administration 184
 - 6.3.5 Benutzer 185
 - 6.4 Starten der Anwendungen 191
 - 6.4.1 JavaClient 191
 - 6.4.2 WebClient 192
 - 6.4.3 Server-Monitoring 192
 - 6.4.4 Server- und Projektkonfiguration 192
 - 6.5 FirstSpirit-Client als JAVA-Anwendung starten 193
 - 6.5.1 Socket-Modus 193
 - 6.5.2 HTTP (Internet)-Modus 195

- 7 FirstSpirit Server- und Projektkonfiguration 197**
 - 7.1 Server- und Projektadministratoren 197
 - 7.2 Elemente der Menüleiste 198
 - 7.2.1 Datei 198
 - 7.2.2 Server 198
 - 7.2.3 Projekt 202



| | | |
|--------|-----------------------------------|-----|
| 7.2.4 | Benutzer | 219 |
| 7.2.5 | Extras | 224 |
| 7.2.6 | Hilfe | 225 |
| 7.3 | Servereigenschaften..... | 226 |
| 7.3.1 | Globale Server-Eigenschaften..... | 226 |
| 7.3.2 | Präsentationskanäle | 231 |
| 7.3.3 | Vorschau-Konfigurationen | 233 |
| 7.3.4 | Konvertierungs-Regeln | 234 |
| 7.3.5 | Installierte Fonts..... | 236 |
| 7.3.6 | Datenbanken..... | 238 |
| 7.3.7 | Sprach-Vorlagen..... | 242 |
| 7.3.8 | Webstart..... | 245 |
| 7.3.9 | Startseite | 246 |
| 7.3.10 | Auftragsübersicht..... | 248 |
| 7.3.11 | Auftragsverwaltung | 249 |
| 7.3.12 | Aktionsvorlagen..... | 249 |
| 7.3.13 | JAAS-Konfiguration..... | 250 |
| 7.3.14 | Module | 251 |
| 7.3.15 | Webserver..... | 258 |
| 7.3.16 | Web-Applikationen | 264 |
| 7.3.17 | Clustering (ab V4.1)..... | 270 |
| 7.4 | Projekteigenschaften..... | 278 |
| 7.4.1 | Projekt..... | 279 |
| 7.4.2 | Optionen..... | 280 |
| 7.4.3 | Ersetzungen | 287 |
| 7.4.4 | Fonts..... | 289 |
| 7.4.5 | Sprachen..... | 290 |
| 7.4.6 | Auflösungen | 294 |



| | | |
|--------|---|-----|
| 7.4.7 | Benutzer | 297 |
| 7.4.8 | Gruppen..... | 299 |
| 7.4.9 | Auftragsübersicht..... | 306 |
| 7.4.10 | Auftragsverwaltung | 307 |
| 7.4.11 | Aktionsvorlagen..... | 307 |
| 7.4.12 | Datenbanken..... | 308 |
| 7.4.13 | Vorlagensätze..... | 309 |
| 7.4.14 | WebEdit Einstellungen..... | 311 |
| 7.4.15 | Quoten | 314 |
| 7.4.16 | Berechtigungen | 315 |
| 7.4.17 | Projekt-Komponenten | 317 |
| 7.4.18 | Web-Komponenten | 319 |
| 7.4.19 | Remote-Projekte..... | 325 |
| 7.4.20 | Medien-Restriktionen (ab V4.1) | 330 |
| 7.4.21 | Client-Applikationen..... | 333 |
| 7.5 | Auftragsplanung..... | 342 |
| 7.5.1 | Auftragsübersicht..... | 343 |
| 7.5.2 | Auftragsverwaltung | 346 |
| 7.5.3 | Aktionsvorlagen..... | 349 |
| 7.5.4 | Hinzufügen / Bearbeiten eines Auftrags (Register Eigenschaften)..... | 351 |
| 7.5.5 | Hinzufügen / Bearbeiten eines Auftrags (Register Aktionen) | 356 |
| 7.5.6 | Hinzufügen von Aktionen zu einem Auftrag | 358 |
| 7.5.7 | Kopieren von Aktionen aus einem anderen Auftrag | 359 |
| 7.5.8 | Hinzufügen von Aktionen über Aktionsvorlagen | 360 |
| 7.5.9 | Projektbezogene Aktionen..... | 363 |
| 7.5.10 | Serverbezogene Aktionen | 386 |
| 7.6 | Clustering: Lastverteilung bei der Generierung (ab V4.1) | 387 |



| | | |
|----------|--|------------|
| 7.6.1 | Einleitung (ab V4.1) | 387 |
| 7.6.2 | Konzept (ab V4.1) | 388 |
| 7.6.3 | Lizenzdatei prüfen (ab V4.1) | 389 |
| 7.6.4 | Konfiguration der Clusterknoten (ab V4.1) | 390 |
| 7.6.5 | Konfiguration des Generierungsauftrags (ab V4.1) | 391 |
| 7.7 | Konfiguration der Rechtschreibprüfung | 393 |
| 7.7.1 | SpellService installieren / deinstallieren (Servereigenschaften) | 393 |
| 7.7.2 | SpellService aktualisieren (Servereigenschaften) | 395 |
| 7.7.3 | Globalen SpellService konfigurieren | 396 |
| 7.7.4 | Globale Wörterbücher konfigurieren | 398 |
| 7.7.5 | Dienst "SpellService" starten und konfigurieren | 401 |
| 7.7.6 | SpellService als Projekt-Komponente hinzufügen | 401 |
| 7.7.7 | Projektspezifische Spell-Service-Konfiguration | 403 |
| 7.7.8 | Projektspezifische Wörterbücher hinzufügen | 405 |
| 7.8 | Unterstützung für Apache FOP (ab V4.1) | 406 |
| 7.9 | Projektarchivierung | 408 |
| 7.9.1 | Versionshistorie | 408 |
| 7.9.2 | Revisionen | 408 |
| 7.9.3 | Mindestanforderung an die Projektarchivierung | 409 |
| 7.9.4 | Versionshistorie nach Archivierung | 411 |
| 8 | FirstSpirit Server-Monitoring | 413 |
| 8.1 | Übersicht | 414 |
| 8.1.1 | Übersicht – Status | 414 |
| 8.1.2 | Übersicht – Aktivitäten | 416 |
| 8.1.3 | Übersicht – Sitzungen | 417 |
| 8.2 | Projekte | 418 |
| 8.2.1 | Projekte – Übersicht | 418 |



| | | |
|----------|--|------------|
| 8.2.2 | Projekte – Statistik..... | 420 |
| 8.3 | Log-Dateien..... | 423 |
| 8.3.1 | Log-Dateien – kompletter Server..... | 423 |
| 8.3.2 | Log-Dateien – nach Projekt..... | 428 |
| 8.3.3 | Log-Datei – nach Veröffentlichung..... | 430 |
| 8.4 | Zeitplanung..... | 433 |
| 8.4.1 | Zeitplanung – Übersicht..... | 433 |
| 8.4.2 | Zeitplanung – geplante Aufträge..... | 434 |
| 8.4.3 | Zeitplanung – ausgeführte Aufträge | 435 |
| 8.5 | Benutzer | 436 |
| 8.5.1 | Benutzer – Suche | 436 |
| 8.6 | FirstSpirit..... | 437 |
| 8.6.1 | FirstSpirit – Konfiguration..... | 437 |
| 8.6.2 | FirstSpirit – Steuerung | 444 |
| 8.6.3 | FirstSpirit – Nachricht..... | 447 |
| 8.6.4 | FirstSpirit – Datenbanken | 447 |
| 8.6.5 | FirstSpirit – Monitoring..... | 449 |
| 8.6.6 | FirstSpirit – Clustering (ab V4.1)..... | 458 |
| 9 | FirstSpirit JMX-Konsole | 460 |
| 9.1 | Start der JMX-Konsole..... | 460 |
| 9.2 | Register | 463 |
| 9.3 | AuthenticationManager..... | 464 |
| 9.4 | BackendManager | 465 |
| 9.4.1 | Attribute | 465 |
| 9.4.2 | Operation | 466 |
| 9.5 | CacheSizeManager | 468 |
| 9.5.1 | Attribute (gesamt)..... | 468 |



| | | |
|-----------|--|------------|
| 9.5.2 | Operation | 469 |
| 9.5.3 | Attribute (projektbezogen)..... | 469 |
| 9.6 | Content Manager..... | 470 |
| 9.7 | Event Manager..... | 471 |
| 9.8 | ExecutionManager | 473 |
| 9.8.1 | Klassifizierung der Thread Queues..... | 473 |
| 9.8.2 | Verarbeitung innerhalb des ExecutionManagers..... | 474 |
| 9.8.3 | Attribute | 475 |
| 9.9 | LockManager | 477 |
| 9.10 | MediaManager | 478 |
| 9.11 | PreviewManager..... | 479 |
| 9.12 | ReferenceManager | 480 |
| 9.13 | RepositoryManager | 481 |
| 9.14 | SearchManager | 482 |
| 9.15 | SessionManager..... | 484 |
| 10 | Sichere Veröffentlichung über rsync und ssh | 485 |
| 10.1 | Webserver unter Unix | 485 |
| 10.2 | Webserver unter Windows | 486 |
| 10.3 | FirstSpirit-Server unter Unix..... | 487 |
| 10.4 | FirstSpirit-Server unter Windows..... | 488 |
| 10.5 | Konfiguration des FirstSpirit-Projekts..... | 490 |
| 11 | Konfiguration von Benutzerrechten | 491 |
| 11.1 | Einführung | 491 |
| 11.1.1 | Definition von Benutzerrechten | 493 |
| 11.1.2 | Prüfung von Benutzerrechten..... | 494 |
| 11.1.3 | Permission-Modul..... | 495 |



| | |
|--|------------|
| 11.2 Architektur..... | 496 |
| 11.2.1 Einleitung | 496 |
| 11.2.2 Übersicht | 497 |
| 11.2.3 Berechtigungsprüfung mit FirstSpirit | 501 |
| 11.3 Schutz personalisierter Projekthinhalte in FirstSpirit | 502 |
| 11.3.1 Personalisierung | 502 |
| 11.3.2 Prüfung der Zugriffsrechte über die Access-Control-Datenbank | 504 |
| 11.3.3 Konzept Sichere Medien | 506 |
| 11.3.4 Abgrenzung | 507 |
| 11.3.5 Berechtigungsdefinition | 508 |
| 11.3.6 Definitionen..... | 511 |
| 11.4 Konfiguration..... | 512 |
| 11.4.1 Einleitung | 512 |
| 11.4.2 Aktivierung des Permission Service | 512 |
| 11.4.3 Konfiguration der Server-Komponente | 512 |
| 11.4.4 Konfiguration für die Veröffentlichung | 515 |
| 11.4.5 Konfiguration der Berechtigungen über die Projekteigenschaften..... | 515 |
| 11.5 Einsatz im Live-System | 516 |
| 11.5.1 Konfiguration des Servlet-Servers..... | 516 |
| 11.6 Einsatz im Projekt..... | 517 |
| 11.6.1 Manuelle Gruppendifinition und ID-Vergabe | 517 |
| 11.6.2 Konfigurationssemantik | 519 |
| 12 Anhang: Konfigurationsdateien..... | 521 |
| 12.1 fs-server.conf | 521 |
| 12.2 fs-wrapper.conf | 521 |
| 12.3 fs-jaas.conf | 525 |
| 12.4 fs-webapp.xml | 526 |



12.5 ssh/rsync-Deployment-Skript..... 528



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.



Die nachfolgende Abbildung zeigt die Gesamtsystemarchitektur:

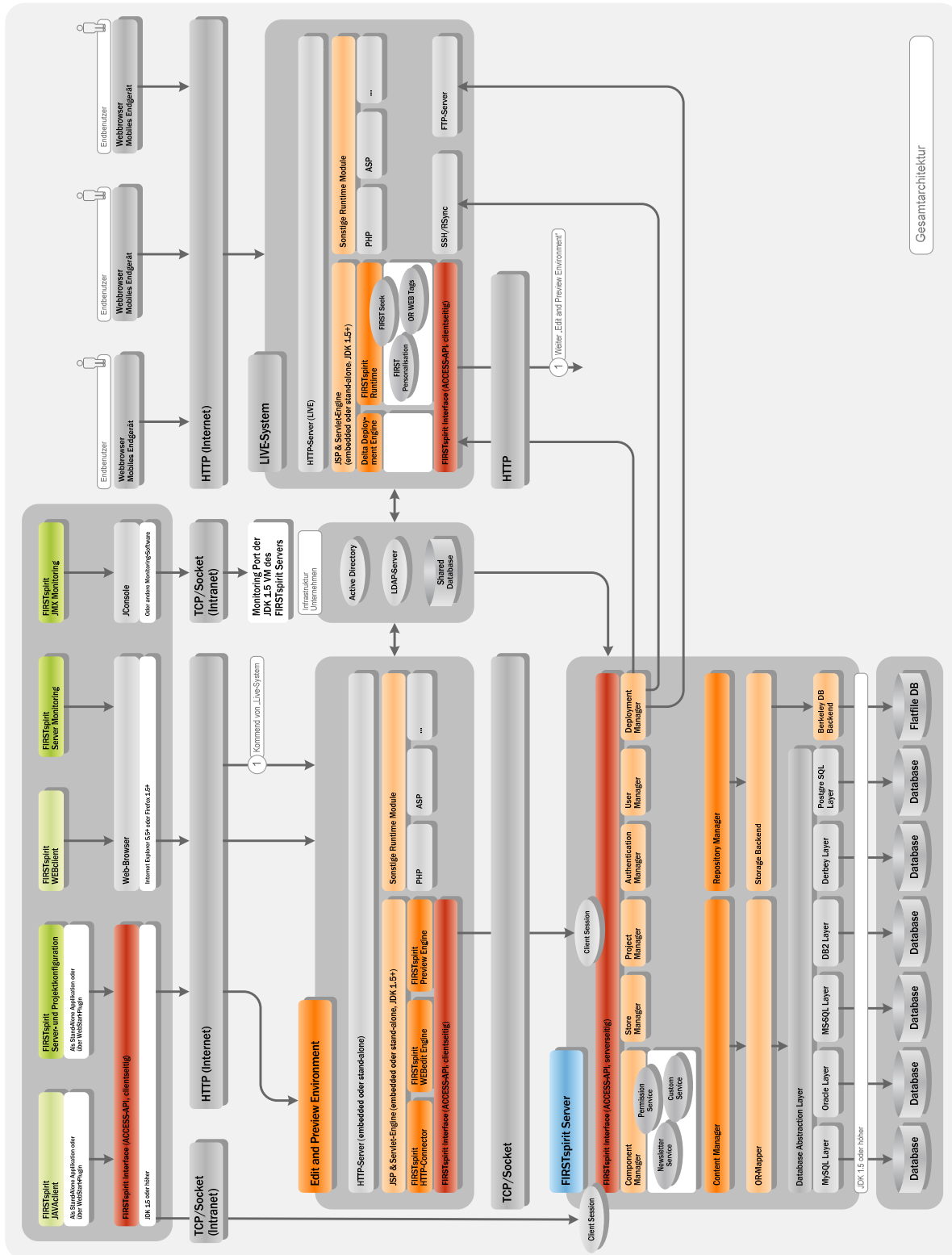


Abbildung 1-1: Gesamtarchitektur



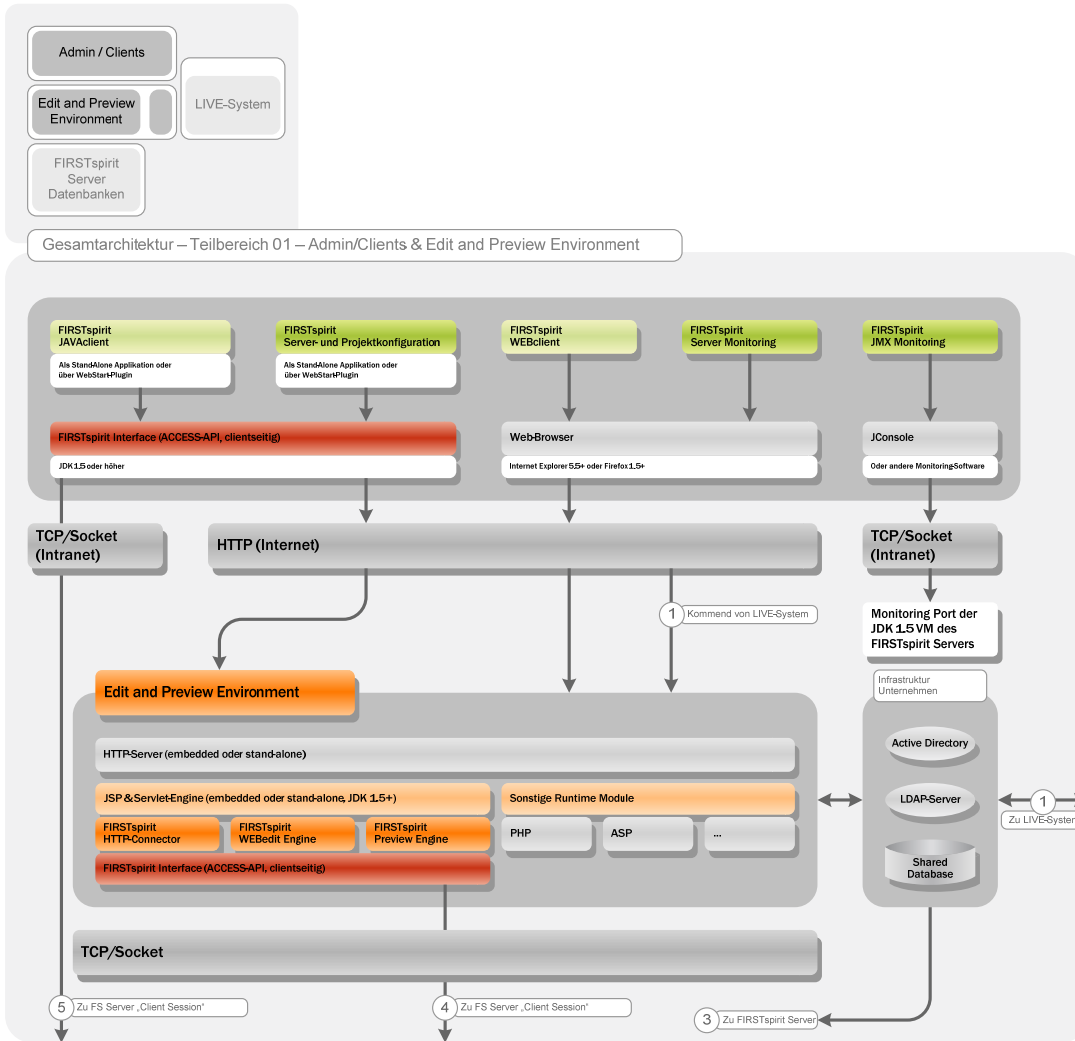


Abbildung 1-2: Gesamtarchitektur – Detail: Edit- and Preview Environment



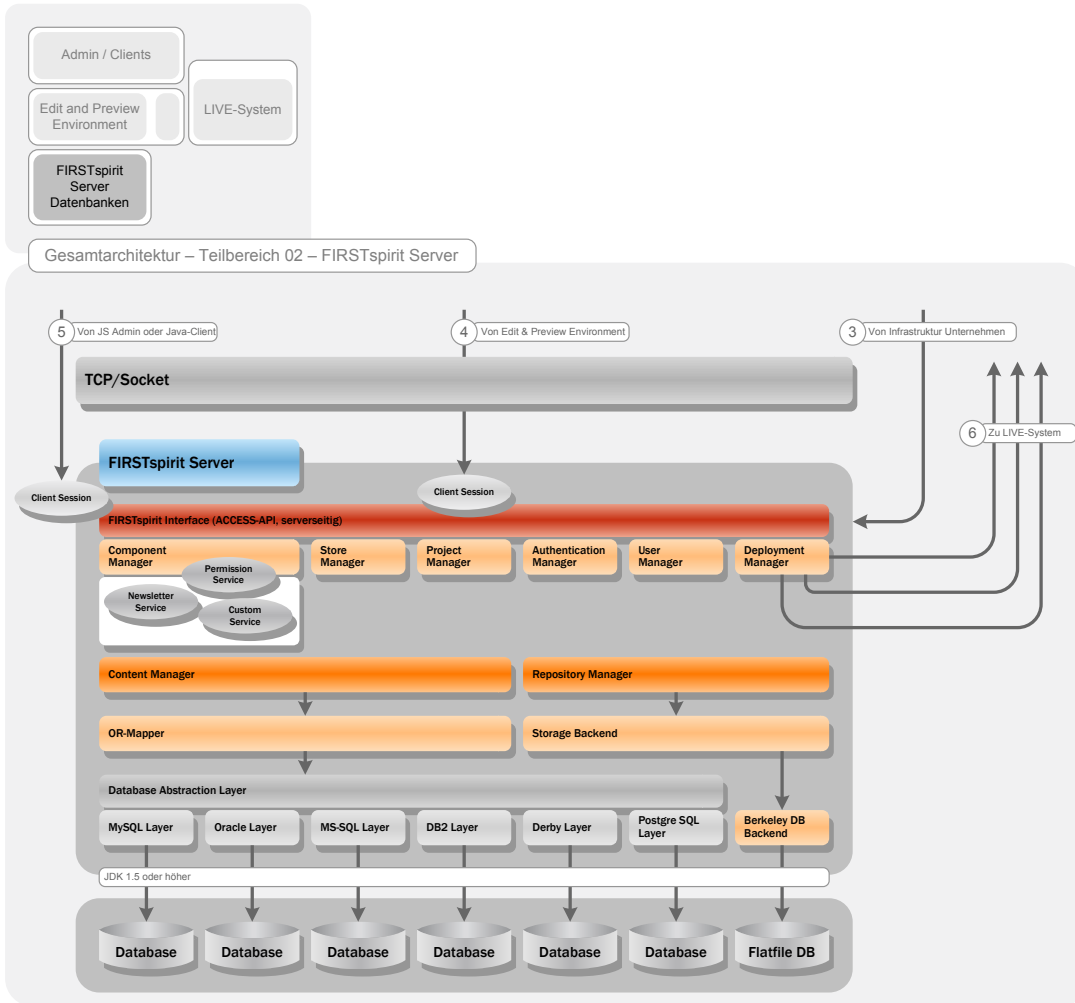


Abbildung 1-3: Gesamtarchitektur – Detail: FirstSpirit™-Server



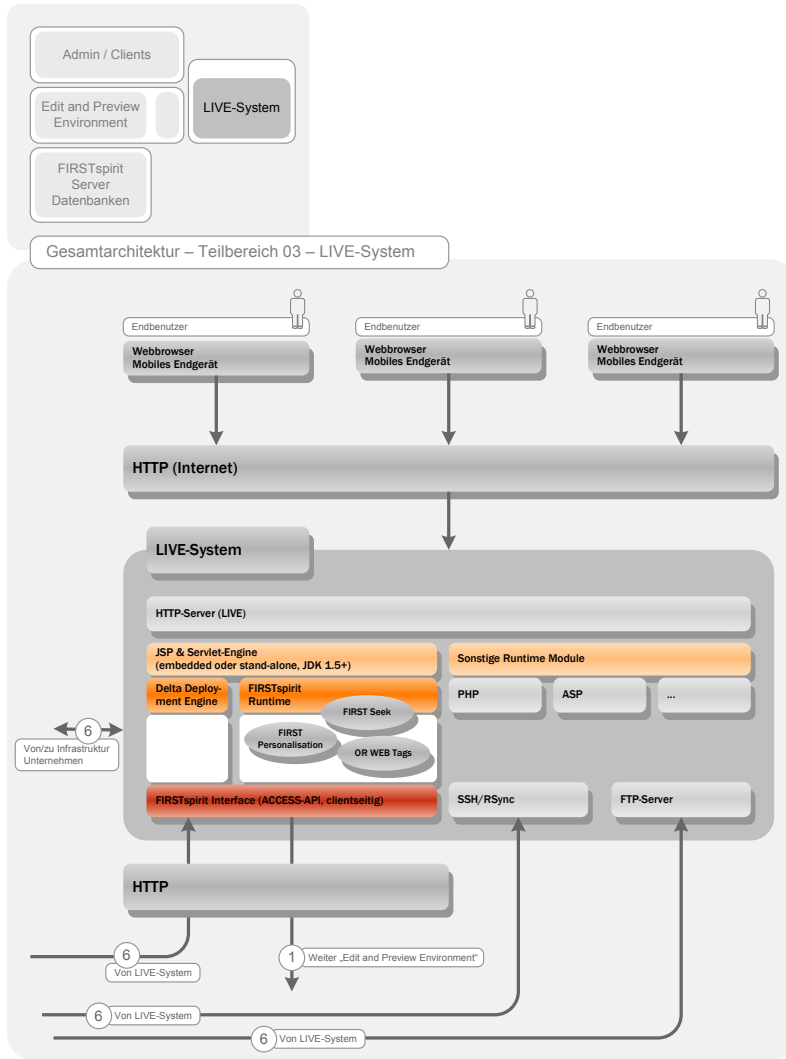


Abbildung 1-4: Gesamtarchitektur – Detail: Live-System



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.

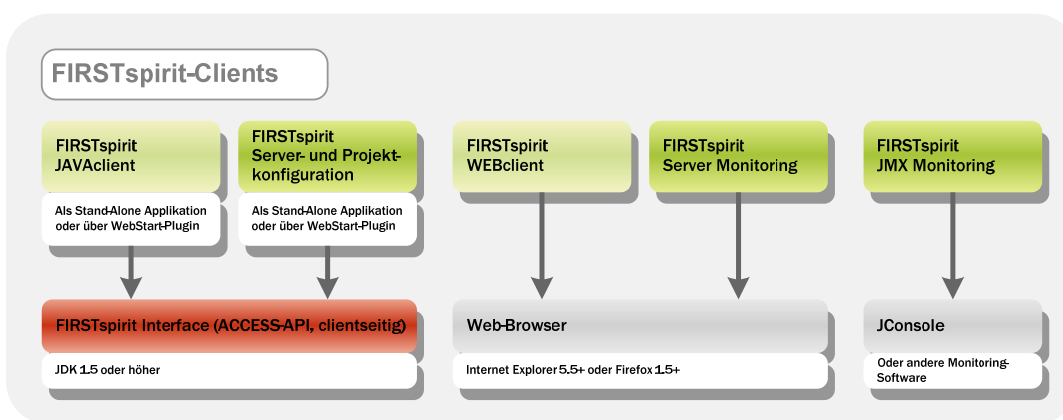


Abbildung 1-5: FirstSpirit-Redaktionsumgebungen

Die **Dokumentation für Administratoren** beschreibt alle Aspekte zur Administration von FirstSpirit V4.x 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 Web-Browser 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 deutlich 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.



Die vorliegende Dokumentation gilt prinzipiell auch für FirstSpirit Version 4.2 Release 2 (4.2R2), soweit nicht anders angegeben.

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 21).

Kapitel 3: In diesem Kapitel wird die Steuerung des FirstSpirit-Servers unter Unix- und Windows-Betriebssystemen erläutert (ab Seite 22).



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 30).

Kapitel 5: Die Konfiguration der FirstSpirit-Webanwendungen – insbesondere die Projektvoraussetzungen, Konfiguration und Einschränkungen des WebClients – wird in diesem Kapitel beschrieben (ab Seite 169).

Kapitel 6: Die FirstSpirit Startseite ist eine Webanwendung zum Starten der Redaktions- und Administrationsumgebungen von FirstSpirit (ab Seite 180).

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 197).

Kapitel 8: Das browserbasierte FirstSpirit Server-Monitoring dient zur Überwachung des FirstSpirit-Servers und zeigt aktuelle Betriebsparameter an (z. B. Benutzeranzahl, Speicherauslastung) (ab Seite 413).

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 460).

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 485).

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 491).

1.3 Die FirstSpirit-Benutzeroberfläche

Für FirstSpirit Version 4.1 erfolgt die offizielle Freigabe des neuen "Look & Feels". Die FirstSpirit-Anwendung zur Server- und Projektkonfiguration sowie der JavaClient sind dann in zwei Darstellungsvarianten verfügbar: im Look & Feel "Classic" und im neuen Look & Feel "LightGray" (Standardeinstellung).

Kunden, die auch in Version 4.1 lieber mit der klassischen "Look & Feel"-Version arbeiten möchten, können diese projektspezifisch oder serverweit über die FirstSpirit Server- und Projektkonfiguration umschalten (Das Umschalten des "Look & Feels" ist nur möglich, wenn die Anwendungen über die FirstSpirit-Startseite gestartet



werden).

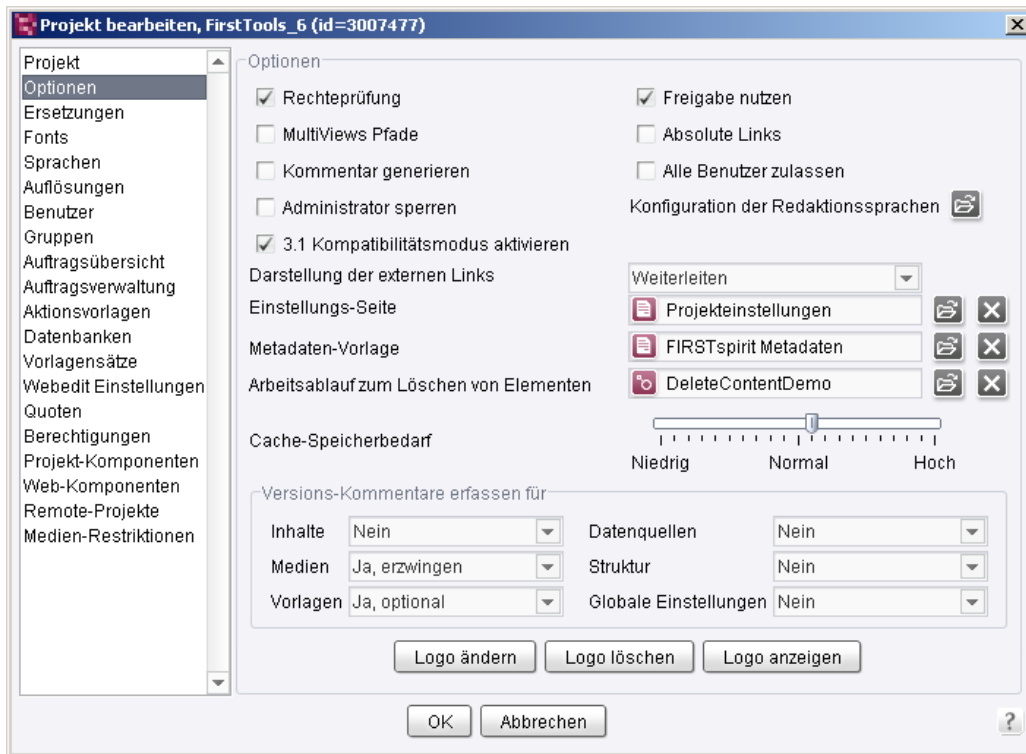


Abbildung 1-6: Look & Feel "LightGray" für die Server- und Projektkonfiguration

Die projektspezifische Umschaltung ist über den zusätzlichen Parameter `lookAndFeel=classic` bzw. `lookAndFeel=lightgrey` bei der Konfiguration der Schnellstart-Einträge über die Server- und Projektkonfiguration möglich (siehe Kapitel 7.3.9 Seite 246).

Die serverweite Umschaltung ist über den zusätzlichen Parameter `lookAndFeel=classic` bzw. `lookAndFeel=lightgrey` der Webstart-Einstellungen für den JavaClient oder die Server- und Projektkonfiguration (Menüeintrag "Webstart") möglich (siehe Kapitel 7.3.8 Seite 245).

Außerdem kann beim Starten der Anwendung der Parameter `-DlookAndFeel=classic` verwendet werden.



Neue Funktionen, die ab FirstSpirit-Version 4.1 freigegeben sind, werden in dieser Dokumentation im neuen Look & Feel dargestellt, die Darstellung bereits bestehender bleibt zunächst im alten Look & Feel.

Ab FirstSpirit Version 4.2 ist das klassische "Look & Feel" nicht mehr verfügbar.



2 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-Webstart verwaltet wird (JavaClient) oder als Web-Anwendung über einen Web-Browser 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.

Der FirstSpirit-Server in Version 4.x ist für folgende Betriebssysteme verfügbar:

- Red Hat Enterprise Linux
- Debian/GNU Linux
- SUSE Linux Enterprise Server
- LSB-konforme Linux-Distributionen
- Solaris
- AIX
- Windows

Alle Betriebssysteme werden in 32- und 64Bit-Architektur unterstützt.

Über die Paketsysteme RPM, Debian-Package oder Solaris-Package kann immer nur genau 1 FirstSpirit-Server unter einer Betriebssystem-Instanz installiert werden. Bei einer erneuten Installation wird der bestehende FirstSpirit-Server aktualisiert, unter Beibehaltung aller Nutz- und Konfigurationsdaten, – besteht also bereits eine FirstSpirit-Installation der Version 4.1, wird beim Ausführen der Installation von FirstSpirit Version 4.2 ein Update des bestehenden Servers von Version 4.1 auf Version 4.2 durchgeführt. Falls mehrere FirstSpirit-Server unter einer Betriebssystem-Instanz installiert werden sollen, muss die Installation unter einem normalen Benutzerkonto über das tar-Archiv erfolgen.

Im Gegensatz dazu wird bei einer Installation unter Windows-Betriebssystemen explizit zwischen einem Update und einer Neuinstallation unterschieden:

- Bei einer Neuinstallation von FirstSpirit Version 4.2 wird ein neuer FirstSpirit-Server (mit separatem Dienst, Startmenü, usw.) installiert.
- Bei einem Update wird die vorhandene Installation aktualisiert.

Weiterführende Informationen zur Installation und zur Aktualisierung von FirstSpirit-



Servern siehe "FirstSpirit Installationsanleitung Version 4.2".



Ab FirstSpirit Version 4.1 ist die parallele Installation eines (oder mehrerer) FirstSpirit-Servers Version 4.0 und eines (oder mehrerer) FirstSpirit-Servers 4.1 unter Windows-Betriebssystemen möglich (siehe "FirstSpirit Installationsanleitung" für Version 4.1).

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 444).

3.1 Unix

3.1.1 GNU/Linux und Solaris 9

Starten als root:

```
/etc/init.d/fs4 start
```

Stoppen als root:

```
/etc/init.d/fs4 stop
```

3.1.2 Solaris 10

Unter Solaris 10 verwendet FirstSpirit die Service Management Facility, sofern verfügbar.

Starten als root:

```
svcadm enable fs4
```

Stoppen als root:

```
svcadm disable -s -t fs4
```



Prozesse auflisten als `root`:

```
svcs -p fs4
```

3.1.3 AIX

Unter AIX dient zum Starten des FirstSpirit-Servers ein Eintrag in der Datei `/etc/inittab`, der mit Kennung "fs4" während der Installation erfolgt.

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

Starten als `root`:

```
/opt/firstspirit4/bin/fs4.init start
```

Stoppen als `root`:

```
/opt/firstspirit4/bin/fs4.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 `fs4` an, deaktiviert aber die Anmeldung für dieses Benutzerkonto. Zum Aktivieren der Möglichkeit als `fs4` über SSH oder Telnet anzumelden, ist es ausreichend, das Passwort für das Benutzerkonto `fs4` einzutragen. Dazu als `root` aufrufen:

```
passwd fs4
```

Anschließend können nach Anmeldung mit dem Benutzerkonto `fs4` zur Steuerung des FirstSpirit-Servers folgende Aufrufe verwendet werden:

Starten als Benutzer `fs4`:

```
firstspirit4/bin/fs4 start
```

Stoppen als Benutzer `fs4`:

```
firstspirit4/bin/fs4 stop
```

3.1.5 Stackdump erzeugen

Über den Befehl:

```
fs4 dump
```

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

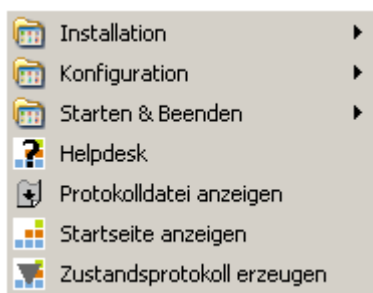


Eine gute Analysemöglichkeit bietet das FirstSpirit Server-Monitoring. Die über die Funktionalität "Threads" erstellten Thread Dumps können hier analysiert und in einer aufbereiteten Ansicht angezeigt werden (siehe Kapitel 8.6.5.5 Seite 452).

3.2 Windows

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

Start / Programme / FirstSpirit4 das folgende Menü 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
- Log-Datei ansehen: siehe Kapitel 3.2.6.2
- Startseite anzeigen: siehe Kapitel 3.2.6.3
- Zustandsprotokoll erzeugen: siehe Kapitel 3.2.6.4

3.2.1 Registrieren / Deregistrieren als Systemdienst über Startmenü

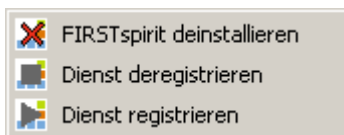


Abbildung 3-1: Dienst registrieren / deregistrieren

Administratoren haben die Möglichkeit, den FirstSpirit-Server als Systemdienst zu 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 26).



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 (vgl. Abbildung 3-1) (weiterführende Informationen zur Deinstallation von FirstSpirit siehe FirstSpirit Installationsanleitung).

3.2.3 Starten/Stoppen als Systemdienst über Startmenü

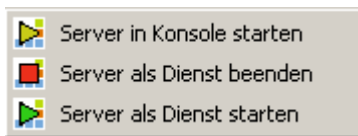


Abbildung 3-2: FirstSpirit-Server starten

Es besteht die Möglichkeit, den FirstSpirit-Server über das Windows Startmenü 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 26).



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 444).



3.2.4 Starten/Stoppen im Konsolenmodus

Mit einem Klick auf den Eintrag "Server in Konsole starten" öffnet sich ein Konsolenfenster in dem der FirstSpirit-Server gestartet wird (vgl. Abbildung 3-2).

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 444).

3.2.5 Konfigurationsdateien über Startmenü editieren

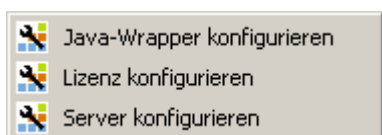


Abbildung 3-3: Konfiguration über das Startmenü

Über das Startmenü 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 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 64). Die Datei sollte, wenn möglich, über das FirstSpirit Server-Monitoring konfiguriert werden (siehe Kapitel 8.6.1.5 Seite 441).

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 92). 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 438).



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 34). Die Datei sollte, wenn möglich, über das FirstSpirit Server-Monitoring konfiguriert werden (siehe Kapitel 8.6.1.1 Seite 437). Bei einigen Änderungen der Konfigurationsdatei `fs-server.conf` ist ein Neustart des Servers erforderlich.



3.2.6 Weitere Startmenü-Funktionen

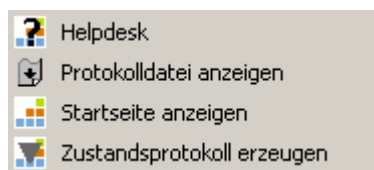


Abbildung 3-4: Weitere Startmenü-Funktionen

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 72).
- fs-server.log: Log-Datei für die Ausgabemeldungen des FirstSpirit-Servers (Konfiguration siehe Kapitel 4.3.6 Seite 96).

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 180).

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 27).

¹ <https://helpdesk.e-spirit.de/>



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 452).



4 Konfiguration des FirstSpirit-Servers

4.1 Dateisystemorganisation

Mit der Einführung von FirstSpirit Version 4.0 wurden die Verzeichnisstrukturen der Serverinstallation, die Benennung der Konfigurations-Dateien und auch die Struktur der Web-Applikationen neu strukturiert und an bestehende Standards angeglichen.

4.1.1 Dateinamen

Einige der wichtigsten Änderungen im Bereich Dateinamen sind:

- Umbenennung der Konfigurations-Dateien:
 - cms.ini → fs-server.conf
 - content.ini → fs-database.conf
 - logging.ini → fs-logging.conf
 - license.ini → fs-license.conf
 - wrapper.conf → fs-wrapper.conf
 - jetty.xml → fs-webapp.xml
- Umbenennung von Programmdateien:
 - CMSServer.jar → fs-server.jar
 - CMSClient.jar → fs-client.jar
 - CMSAccess.jar → fs-access.jar
 - OR.jar → fs-or.jar
- Umbenennung von Log-Dateien:
 - CMS_Server.log → fs-server.log
 - Wrapper.log → fs-wrapper.log
 - Deployment.*.log → fs-schedule.*.log



4.1.2 Webanwendungen

Im Bereich der zu FirstSpirit gehörigen Web-Applikationen haben sich folgenden Veränderungen und Neuerungen ergeben:

- Startseite und JavaClient (Dynamische Vorschau): fs4root
- Staging (lokales Vorschausystem, Statische Vorschau): fs4staging
- WebEdit und WebEdit-Vorschau: fs4webedit
- Server-Monitoring: fs4webmon
- Vorschau (nicht WebEdit): fs4preview

4.1.3 Verzeichnisstrukturen

Die Umorganisation der Verzeichnisstrukturen:

- Systembereiche, die bei der Server-Aktualisierung komplett überschrieben werden:
 - ~fs4\bin Start-Umgebung inkl. benötigter System-Binaries
 - ~fs4\server Java-Umgebung inkl. aller Bibliotheken
 - ~fs4\web FirstSpirit Web-Applikationen
- Konfigurations-Bereiche, die vom Benutzer für die Systemkonfiguration verändert werden können:
 - ~fs4\conf FirstSpirit Konfigurations-Dateien (siehe Kapitel 4.3)
- Datenbereiche:
 - ~fs4\data alle Projektdaten inkl. Module usw.
- Temporäre Dateibereich & Exporte:
 - ~fs4\backup Projektexporte für Datensicherung
 - ~fs4\export Projektexporte für interaktive Export/Import-Vorgänge
 - ~fs4\log Log-Dateien
 - ~fs4\work Temporäre FirstSpirit Dateien
 - ~fs4\web\fs4staging\projects Staging mit Web-Apps
 - ~fs4\web\fs4preview\preview_cache Vorschau-Cache
 - ~fs4\web\fs4webedit\preview_cache Vorschau-Cache WEBedit



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 33).

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 5.3 Seite 176). 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 413 und Kapitel 9 Seite 430). 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 9.5 Seite 468).



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 35).



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.

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)



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 437). 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 35)
- Server (Kapitel 4.3.1.2 Seite 38)
- Server-Monitoring (Kapitel 4.3.1.3 Seite 40)
- Thread Pool (Kapitel 4.3.1.4 Seite 41)
- Thread Queues (Kapitel 4.3.1.5 Seite 42)
- JAAS (Kapitel 4.3.1.6 Seite 44)
- Web Applications (Kapitel 4.3.1.7 Seite 45)
- Path (Kapitel 4.3.1.8 Seite 48)
- Preview (Kapitel 4.3.1.9 Seite 49)
- Mail (Kapitel 4.3.1.10 Seite 51)
- LDAP (Kapitel 4.3.1.11 Seite 52)
- Storage Engine Properties (Kapitel 4.3.1.12 Seite 57)
- CacheManager (Kapitel 4.3.1.13 Seite 57)
- Interne Datenbank (Kapitel 4.3.1.14 Seite 58)



- WEEdit configuration (Kapitel 4.3.1.15 Seite 59)
- Miscellaneous (Kapitel 4.3.1.16 Seite 62)
- Webstart configuration (Kapitel 4.3.1.17 Seite 63)

4.3.1.1 Bereich: Communication

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

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

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

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

URL=http://fs4server.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 den automatisch ermittelten bzw. durch den Parameter `URL` definierten 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=4200
SOCKET_PORT=4300
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 6.3.5.1 Seite 186), werden gemäß der in Kapitel 7.3.9 Seite 246 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.*

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

`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.

`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.

`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.

`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.

`ALLOWED_ENCRYPTIONS` (optional): Über diesen Parameter kann dem Benutzer vorgegeben werden, welche Verschlüsselung er in seinen Verbindungseinstellungen (siehe Kapitel 6.3.5.1 Seite 186) 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 Web-Anwendung `fs4root` 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.).

`HELP_URL`: Definiert den relativen URL, unter dem die FirstSpirit-Online-Hilfe abgelegt ist (zur Notation `{}` siehe Kapitel 4.2 Seite 32). Beispiel:
`HELP_URL=${WEBAPP_ROOT_NAME}/help`

`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`

`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).

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

`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`).



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=
```

Ab Version 4.2R4 wurden für eine bessere Performance die Standardwerte für die Größe des *Thread Pools* und der *Thread Queues* angepasst.

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 9-13 Seite 474).

`ThreadPool.maxSize`: Definiert die Maximalgröße des limitierten ThreadPools und damit die größtmögliche Anzahl parallel auszuführender Tasks (vgl. Abbildung 9-13 Seite 474). 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 ThreadPool).

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).
# - rejection     rejection strategy for incoming tasks if queue
# capacity is exceeded (REJECT, BLOCK, EXECUTE).
#                 REJECT reject task if queue capacity is exceeded
#(only allowed in BOUNDED queue!).
#                 BLOCK  wait until queue has free capacities.
#                 EXECUTE execute queue's oldest task inline.
#####
ThreadQueue.LOW.maxRunning=2
ThreadQueue.LOW.queueCapacity=128
ThreadQueue.LOW.rejection=BLOCK

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

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

Die Ausführung von Tasks erfolgt über den ExecutionManager, der eine Vielzahl von unterschiedlich klassifizierten Queues verwaltet (siehe Kapitel 9.8 Seite 473). 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 (wovon die unter `maxRunning` definierte Anzahl direkt abgearbeitet wird) und weitere Tasks vorerst zurückgewiesen. Nach Ablauf einer bestimmten Zeitspanne wird dann der erneute Versuch unternommen, diese Tasks in die Queue zu stellen (Standardwert ist 50 Threads).

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 41).

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 76). Eine komfortable Möglichkeit zur Konfiguration dieses Bereichs befindet sich in der Server- und Projektkonfiguration (siehe Kapitel 7.3.13 Seite 250) und über das Server-Monitoring (siehe Kapitel 8.6.1.8 Seite 443).

² Java Authentication and Authorization Service (Informationen: <http://java.sun.com/products/jaas/>)



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_WEBEDIT_URL=/${WEBAPP_WEBEDIT_NAME}
WEBAPP_WEBMON_URL=/${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_WEBEDIT_PATH=${WEB_DIR}/${WEBAPP_WEBEDIT_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 fs4root site
WEBAPP_ROOT_TITLE=FirstSpirit ${HOSTNAME}:${HTTP_PORT}

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

WEBAPP_ROOT_URL: URL zur Webanwendung Startseite (fs4root). Der hier hinterlegte Wert wird für das Mapping innerhalb der internen Servlet Engine benötigt (siehe auch Kapitel 4.3.7.2 Seite 101). 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_WEBEDIT_URL: URL zur Webanwendung WebEdit (fs4webedit) 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 101). 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_WEBEDIT_NAME}`).

WEBAPP_WEBMON_URL: URL zur Webanwendung "Server-Monitoring" (fs4webmon) 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 101). 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 (fs4staging). Der hier hinterlegte Wert wird für das Mapping innerhalb der internen Servlet Engine benötigt (siehe auch Kapitel 4.3.7.2 Seite 101). 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 (fs4preview). Der hier hinterlegte Wert wird für das Mapping innerhalb der internen Servlet Engine benötigt (siehe auch Kapitel 4.3.7.2 Seite 101). 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 (fs4root, fs4preview, fs4staging, fs4webedit, fs4webmon) (zur Notation `${}` siehe Kapitel 4.2 Seite 32).

WEBAPP_ROOT_PATH: Pfadangabe zum Verzeichnis im Dateisystem in dem der FirstSpirit-Server die Webanwendung FirstSpirit Startseite (fs4root) ablegt (zur Notation `${}` siehe Kapitel 4.2 Seite 32).

WEBAPP_WEBEDIT_PATH: Pfadangabe zum Verzeichnis im Dateisystem in dem der FirstSpirit-Server die Webanwendung WebEdit (fs4webedit) ablegt (zur Notation `${}` siehe Kapitel 4.2 Seite 32).



`WEBAPP_WEBMON_PATH`: Pfadangabe zum Verzeichnis im Dateisystem in dem der FirstSpirit-Server die Webanwendung Server Monitoring (fs4webmon) ablegt (zur Notation `${}` siehe Kapitel 4.2 Seite 32).

`WEBAPP_STAGING_PATH`: Pfadangabe zum Verzeichnis im Dateisystem in dem der FirstSpirit-Server die Webanwendung FirstSpirit Staging (fs4staging) ablegt. In diesem Verzeichnis legt der FirstSpirit-Server auch alle generierten Projektdateien ab (zur Notation `${}` siehe Kapitel 4.2 Seite 32).

`WEBAPP_PREVIEW_PATH`: Pfadangabe zum Verzeichnis im Dateisystem in dem der FirstSpirit-Server die Webanwendung Preview (fs4preview) ablegt. In diesem Verzeichnis legt der der FirstSpirit-Server auch die generierten Vorschauseiten ab. Der Parameter kann auch über die Server- und Projektkonfiguration definiert werden (siehe Kapitel 7.3.3 Seite 233).

`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 35), 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: Path

```
#####  
# paths  
#####  
BACKUP_PATH=${cmsroot}/backup  
BATCHPATH=${cmsroot}/data/scripts  
  
# base path for "rollout client applications"  
CLIENTAPP_PATH=${cmsroot}/data/clientapp
```

BACKUP_PATH: Pfadangabe zum Backup-Verzeichnis, das Verzeichnis im Dateisystem, in dem der FirstSpirit-Server die Backups ablegen soll.

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

CLIENTAPP_PATH: Der Parameter wird ausschließlich für den Ausrollprozess nativer Clientapplikationen (ab FirstSpirit Version 4.2) benötigt (siehe Kapitel 4.9.1 Seite 165). Standardwert: ~FS42\data\clientapps.

CLIENT_HOME_DIR: (ab FirstSpirit Version 4.2R2) Pfadangabe zum Verzeichnis im Dateisystem, in das Client-Applikationen abgelegt werden sollen (siehe Kapitel 4.9.2 Seite 165). 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. **Ab 4.2R4** wird für jede FirstSpirit-Version ein eigenes Verzeichnis verwendet. Der Verzeichnisname enthält dabei die jeweilige FirstSpirit-Major-, -Minor- und -Release-Versionsnummer, z. B. `\.firstspirit_4.2R4`.



Das bis 4.2R2 einschließlich verwendete Verzeichnis `\.firstspirit` wird dabei nicht vom System gelöscht, sondern muss bei Bedarf (z. B. um Speicherplatz freizugeben) manuell entfernt werden.



Standardwert **bis 4.2R2 einschließlich:**

```
CLIENT_HOME_DIR=~/.firstspirit
```

ab 4.2R4:

```
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`).

4.3.1.9 Bereich: Preview

```
#####  
# preview  
#####  
preview.cacheMediaFiles=  
preview.internalDelivery=html,htm,txt,xml,pdf,jsp,shtml,ini  
preview.cacheTimeout=120
```

`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.)

Bis FirstSpirit Version 4.1 ist der Standardwert `html,htm,txt,xml,pdf,jsp,shtml,ini`. **Ab FirstSpirit Version 4.2** ist der 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.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://fs4.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 (Standardwert ist 120 Sekunden).

Neue Parameter ab FirstSpirit Version 4.2:

`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.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.

4.3.1.10 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.default-recipient`: Angabe einer Standard-Email-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 ein Standardwert verwendet.

`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_ADDRESS` in Kap. 4.3.1.10 Seite 51) geschickt.

4.3.1.11 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 76). 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!

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 220).

`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 `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 `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 ab FirstSpirit Version 4.2R2 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 103.



4.3.1.12 Bereich: Storage Engine Properties

```
#####  
# storage engine properties  
#####  
repository.je.checkpointer.wakeupInterval=1000000  
repository.je.maxMemoryPercent=25
```

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.



Bei Änderungen an der Datei `fs-server.conf` ist ein Neustart des Servers erforderlich (vgl. Kapitel 4.2 Seite 32).

4.3.1.13 Bereich: CacheManager

```
#####  
# cache manager  
#####  
CACHE_PERCENT=40
```

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 40 gesetzt.

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



4.3.1.14 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 75).

Ein Beispiel zur Konfiguration der Derby-Datenbank für eine Verwendung von externen Prozessen aus (z. B. Web-Anwendung mit FirstSpirit-Modul Integration im externen Application-Server) findet sich in Kapitel 4.8.7.6 Seite 162.



4.3.1.15 Bereich: WEBedit configuration

```
#####
# WEBedit configuration
#####
webedit.globalTheme=default
webedit.errorPage=ErrorPage.jsp
webedit.noLogin=0
webedit.webSSO=0

# Configuration switches
webedit.showReferenceName=0
webedit.sitestoreWorkflow=1
webedit.serviceUrl=mailto:please_set_email_url_in_fs-server.conf
webedit.plainPassword=0
webedit.additionalLogoutUrl=

# TemplatePreview
webedit.preview.noPreviewUrl=themes/preview.gif

# WebeditRefresh
webedit.refresh.checkInterval=60
webedit.refresh.minReload=90
webedit.refresh.maxReload=300
webedit.refresh.reloadDelta=10

webedit.sessionCache.lru_size=0
webedit.globalCache.lru_size=5120
web.sessionCache.lru_size=0

# Content expiration:
webedit.expirationMode=default
```

`webedit.globalTheme`: Der Parameter dient zur Angabe eines Basis-Themes, das verwendet wird, wenn noch kein Projekt ausgewählt wurde (z. B. Login, Projektauswahl). Das Theme für ein Projekt wird über die FirstSpirit Projektkonfiguration definiert und ausgewertet, sobald der Benutzer das entsprechende Projekt ausgewählt hat (siehe Kapitel 7.4.14 Seite 311).

`webedit.errorPage`: Über den Parameter kann eine Seite (.jsp) definiert werden, die angezeigt wird, wenn ein Fehler (Benutzer- oder Systemfehler) beim Arbeiten mit WebEdit aufgetreten ist.

`webedit.noLogin`: Über den Parameter kann das Anzeigen der Anmeldeseite unterbunden werden (Wert=0), wenn beispielsweise der Anmeldevorgang automatisch über das SAP-Portal erfolgt.

`webedit.webSSO`: Über den Parameter kann definiert werden, ob der Anmeldevorgang für WebEdit über SSO (Single Sign-on) möglich ist



(Wert=1) oder nicht (Wert=0) (Standardwert 0).

`webedit.showReferenceName`: Über den Parameter kann definiert werden, ob Objekte, beispielsweise Seiten, in WebEdit mit dem eindeutigen Referenznamen (Wert=1) oder mit der sprachabhängigen Bezeichnung (Wert=0) angezeigt werden. Werden die sprachabhängigen Bezeichnungen genutzt, muss innerhalb der Projektkonfiguration außerdem die Checkbox "Anzeigename sprachabhängig" aktiviert werden (siehe Kapitel 7.4.2 Seite 280).

`webedit.sitestoreWorkflow`: Über den Parameter kann definiert werden, ob ein Arbeitsablauf, der über die WebEdit-Leiste gestartet wird, auf einer Seitenreferenz der Struktur-Verwaltung (Wert=1) oder einer Seite der Inhalte-Verwaltung (Wert=0) ausgeführt wird (siehe Kapitel 5.2.7.1 Seite 175).

`webedit.serviceUrl`: Über den Parameter kann eine E-Mail-Adresse für die Fehlerseite definiert werden, an die beim Auftreten eines Fehlers im WebClient eine Benachrichtigung mit Fehlerreport gesendet wird (vgl. `webedit.errorPage`).

Der Standardwert `mailto:please_set_email_url_in_fs-server.conf` muss beim Einsatz von WebEdit entsprechend konfiguriert werden.

`webedit.plainPassword`: Über diesen Parameter kann definiert werden, ob das Passwort innerhalb der Anmeldeseite im Klartext gesendet wird (Wert=1) oder nicht (Wert=0) (siehe Kapitel 5.3.3 Seite 177).

`webedit.additionalLogoutUrl`: Hier kann ein zusätzlicher URL (ohne URL-Parameter) für den Abmeldevorgang in WebEdit angegeben werden.

`webedit.preview.noPreviewUrl`: In den Dialogen, die über die WebEdit-Quick-Bar beim Anlegen von Seiten und Absätzen geöffnet werden, sind Vorschaubilder hinterlegt, die in den Seiten- und Absatzvorlagen definiert werden können. Wurde für eine Vorschau kein Bild in der Vorlagen-Verwaltung hinterlegt, so wird das im Parameter `webedit.preview.noPreviewUrl` definierte Bild angezeigt. Die Angabe ist relativ zum Application-Root von WebEdit.

`webedit.sessionCache.lru_size`: Der Parameter bestimmt die Größe des LRU⁴-Caches für jede einzelne WebEdit-Sitzung. Der LRU-Cache ist der Cache für alle FirstSpirit-Objekte, die innerhalb des Projekts

⁴ LRU: Least recently used



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 WebEdit-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 WebEdit-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.

`webedit.expirationMode`: Der Parameter bestimmt den Cache-Modus. Der Wert "expires" wirkt sich momentan ausschließlich auf die Aktualisierung der Thumbnail-Übersichten in WebEdit aus. Möglich sind die Werte:

- Wert "none":
Keine "pragma"-, "Cache-Control"- und "Expires"-Header
- Wert "default" (Standardeinstellung):
"pragma"="no-cache", "Cache-Control"="no-cache", "Expires"="0":
Kein "Pragma" und "Cache-Control"-Header, "Expires" enthält einen RFC 1123-konformen Datumswert. Der Wert berechnet sich aus dem Zeitpunkt der Auslieferung plus dem angegebenen Wert in Stunden (GMT, also für unseren Breitengrad CEST-2 bzw. CET-1).

⁵ DTO: Data Transfer Object



4.3.1.16 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
```

Die folgenden Parameter sind ab FirstSpirit Version 4.2R4 gültig:

`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 226, 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.

4.3.1.17 Bereich: Webstart configuration

Zum Starten der FirstSpirit Server- und Projektkonfiguration und des FirstSpirit-JavaClients wird das Sun Java Runtime Environment (JRE) ab Version 1.5.0 benötigt (empfohlen wird die Version 1.6.0.), darin ist Java Webstart enthalten. (Das JRE wird i.d.R. automatisch bei der Installation von JDK 1.5.0. bzw. 1.6.0. installiert.) Die Konfiguration erfolgt über JNLP-Dateien:

- global (serverweit) über die FirstSpirit Server- und Projektkonfiguration (siehe Kapitel 7.3.8 Seite 245) oder über Parameter in der Konfigurationsdatei `fs-server.conf` (siehe unten)
- benutzerspezifisch über die Verbindungseinstellungen auf der FirstSpirit-Startseite (siehe Kapitel 6.3.5.1 Seite 186).

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 7.3.9 Seite 246).

```
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 6.3.5.1 Seite 186).





Für den Parameter `webstart.client.encryption` kann keine TLS-Verschlüsselung (Parameterwert 1) verwendet werden, wenn Websphere als Application-Server für die Web-Anwendung `fs4root` 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 6.3.5.1 Seite 186).



Für den Parameter `webstart.admin.encryption` kann keine TLS-Verschlüsselung (Parameterwert 1) verwendet werden, wenn Websphere als Application-Server für die Web-Anwendung `fs4root` 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.

4.3.2 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 Stopp 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 441).



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 12.2 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/fc-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 Sun 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 in Version 1.5.0 und 1.6.0



einen vergleichbaren Garbage Collector zur Vermeidung langer GC-Pausen (-Xgcpolicy:optavgpause).

Seit FirstSpirit Version 4.2 werden die im folgenden Abschnitt genannten Parameter 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.

Bei einer Aktualisierung von einer früheren Version als FirstSpirit 4.2 sollten die Parameter vollständig durch die hier genannten ersetzt 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 Sun

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

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

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

Sofern nur die 32bit-Java-VM verwendet werden soll, wird der Parameter "-d64" entfernt und ein Leereintrag wie folgt vorgenommen, um die fortlaufende Nummerierung der Einträge "wrapper.java.additional" zu gewährleisten:

```
wrapper.java.additional.5=
```



Zur Konfiguration des CMS-GC dienen nachfolgende Zeilen:

```
wrapper.java.additional.6=-Xshare:off
wrapper.java.additional.7=-Xmn250M
wrapper.java.additional.8=-XX:PermSize=200M
wrapper.java.additional.9=-XX:MaxPermSize=300M
wrapper.java.additional.10=-XX:+DisableExplicitGC
wrapper.java.additional.11=-XX:SoftRefLRUPolicyMSPerMB=20
wrapper.java.additional.12=-XX:+UseParNewGC
wrapper.java.additional.13=-XX:+UseConcMarkSweepGC
wrapper.java.additional.14=-XX:+CMSIncrementalMode
wrapper.java.additional.15=-XX:+CMSParallelRemarkEnabled
wrapper.java.additional.16=-XX:+CMSClassUnloadingEnabled
wrapper.java.additional.17=-XX:SurvivorRatio=1
wrapper.java.additional.18=-XX:TargetSurvivorRatio=80
wrapper.java.additional.19=-XX:InitialTenuringThreshold=15
wrapper.java.additional.20=-XX:-UseLargePages
wrapper.java.additional.21=-Djava.rmi.dgc.leaseValue=3600000
wrapper.java.additional.22=-Dsun.rmi.dgc.server.gcInterval=3600000
wrapper.java.additional.23=-Dsun.rmi.dgc.client.gcInterval=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.

Zumindest in der Test- und Entwicklungsphase sollten die Parameter für das Protokollieren der Garbage-Collector-Aufrufe aktiviert werden:

```
wrapper.java.additional.24=-verbose:gc
wrapper.java.additional.25=-XX:+PrintGCTimeStamps
wrapper.java.additional.26=-XX:+PrintGCDateStamps
wrapper.java.additional.27=-XX:+PrintGCDetails
wrapper.java.additional.28=-XX:+PrintGCApplicationStoppedTime
```

Die Protokollierung des Garbage Collectors und vor allem die Protokollierung der durch den Garbage Collector bei Überlastung verursachten Pausen des FirstSpirit-Servers erfolgt damit in die Datei `log/fs-wrapper.log`. Diese Datei wird automatisch bei Erreichen einer in `conf/fs-wrapper.conf` definierten Größe archiviert und nach mehreren Archivierungsschritten, ebenfalls in `conf/fs-wrapper.conf` einzustellen, gelöscht.



Folgender Parameter kann für Testsysteme ergänzt werden, um eine übersichtlichere Protokollierung des Garbage Collectors in eine eigene Datei (`log/fs-gc.log`) zu aktivieren. Diese Konfiguration ist für einen Produktionsbetrieb nicht geeignet, da die Log-Datei während der Laufzeit des FirstSpirit-Servers ständig wächst und bei einem Neustart des FirstSpirit-Servers wieder gelöscht wird.

```
wrapper.java.additional.29=-Xloggc:log/fs-gc.log
```

Weiterführende Informationen zu den einzelnen Parametern sind unter <http://java.sun.com/javase/technologies/hotspot/vmoptions.jsp> 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 höher CPU-Auslastung und geringerer Datendurchsatz.

`gencon`: Verbessertes nebenläufiger GC mit zusätzlicher Aufteilung des Heaps in mehrere Generationen. Ist erst mit Java 1.6.0 verfügbar.

Folgende Parameter aktivieren außerdem das Protokollieren der Garbage-Collector-Aufrufe in die Datei `log/fc-gc.log`, was im Test- und Entwicklungsbetrieb sinnvoll ist:

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

Alle weiteren Parameter im Bereich "wrapper.java.additional", außer denen 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/jdk1.6.0/bin/java` oder `c:\JDK1.6.0\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 32Bit-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.



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 32Bit-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 32Bit-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 fortlaufende, eindeutige Nummerierung (X) enthalten.

`wrapper.java.additional.X = *.jmxremote.*`: Konfiguration des JMX-Connectors. JMX dient zur Abfrage des Systemstatus und liefert aktuelle Systeminformationen des FirstSpirit-Servers und Java-Systems. Als Client kann z. B. `jconsole` aus dem JDK 1.6 oder 1.5 benutzt werden oder System-Monitore die das JMX-Protokoll unterstützen. Weiterführende Konfigurationshinweise finden Sie unter <http://java.sun.com/j2se/1.5.0/docs/guide/management/agent.html> Folgender Einträge aktivieren den JMX-Connector auf Port 9000 **ohne** Authentifizierung:

```
wrapper.java.additional.30=-Dcom.sun.management.jmxremote
wrapper.java.additional.31=-Dcom.sun.management.jmxremote.ssl=false
wrapper.java.additional.32=-Dcom.sun.management.jmxremote.authenticate=false
wrapper.java.additional.33=-Dcom.sun.management.jmxremote.port=9000
```



`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 "fs4 console" gestartet wurde oder unter Windows über "Startmenü -> FirstSpirit4 -> 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.

Die Log-Ausgaben des FirstSpirit-Servers wurden mit **FirstSpirit Version 4.2** erweitert. Der FirstSpirit-Server loggt einmalig beim Starten und zyklisch jede Stunde die VM-StartTime und die VM-Uptime:



| Suchergebnis (2) | |
|------------------|---|
| INFO | [-] 11.05.2009 16:08:33 (de.espirit.firstspirit.server.CMS\$Server): Uptime 25.245.018, StartTime 1.242.025.668.098 (11.05.2009 09:07:48) |
| INFO | [-] 11.05.2009 15:08:32 (de.espirit.firstspirit.server.CMS\$Server): Uptime 21.644.501, StartTime 1.242.025.668.098 (11.05.2009 09:07:48) |

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

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 4.0 (bzw. 4.1) -> 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 fortlaufender 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
```



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 7.3.6 Seite 238). 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 133. 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 149 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 7.3.13 Seite 250) oder über das Server-Monitoring vorgenommen werden (siehe Kapitel 8.6.1.8 Seite 443). 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 12.3 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. Eine weitere Möglichkeit zur Aktualisierung bietet die JMX-Konsole. Über den "AuthenticationManager" kann hier die Operation `reloadConfiguration` ausgeführt werden, die ebenfalls für eine Aktualisierung auf dem Server sorgt (siehe Kapitel 9.3 Seite 464).



*Für den Ausnahmefall, dass **nicht** LDAP, Kerberos oder NTLM zur Authentifizierung verwendet werden, werden in Kennworten bei der Anmeldung nur ASCII-Zeichen geprüft. Siehe auch Kapitel 7.2.4.1 Seite 219.*

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

⁶ Java Authentication and Authorization Service: <http://java.sun.com/j2se/1.5.0/docs/guide/security/jgss/tutorials/>



4.3.4.1 Passwortprüfung gegen die FirstSpirit-Benutzerdatenbank

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

Es wird die FirstSpirit-interne Benutzerdatenbank verwendet.

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 `LdapLoginModule` erfolgt der Verweis auf eine in `fs-server.conf` definierte LDAP-Section, in der die Zugangsdaten zum LDAP-Server eingetragen sind, siehe Kapitel 4.3.1.11. 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 `LdapLoginModule` 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 Web-Browser 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 Web-Browser den Berechtigungsnachweis automatisch an FirstSpirit übermittelt. Als Web-Browser wird für dieses Anmeldeverfahren derzeit nur Internet-Explorer 6 und 7 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 Web-Browser zu aktivieren, da NTLM einen nicht ganz standardkonformen HTTP-Header ("WWW-Authenticate:



Negotiate") verwendet, den einige ältere Web-Browser als Fehler interpretieren. Um NTLM bei allen Web-Browsern zu verwenden ".*" eintragen.

Standardwert: ".*MSIE.*"

Das Modul unterstützt NETBIOS und Active Directory Domains. "Browser-Domain" ist die vom Web-Browser 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.



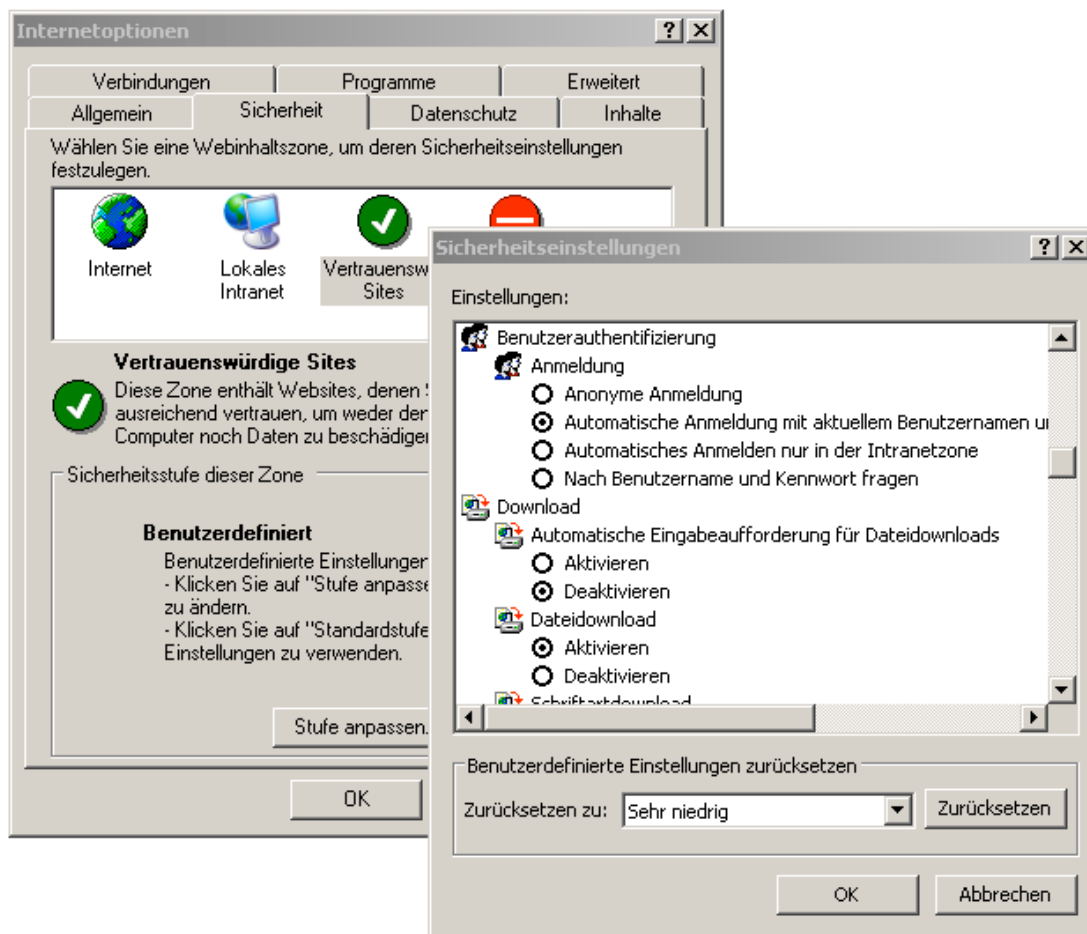


Abbildung 4-2: Sicherheitseinstellungen anpassen

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":



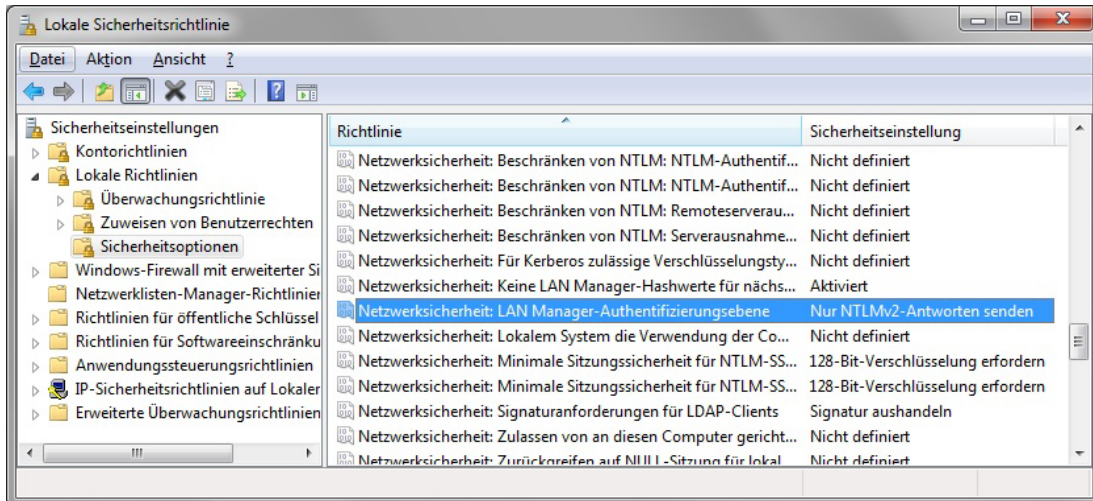


Abbildung 4-3: Netzwerksicherheit: LAN-Manager-Authentifizierungsebene

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):

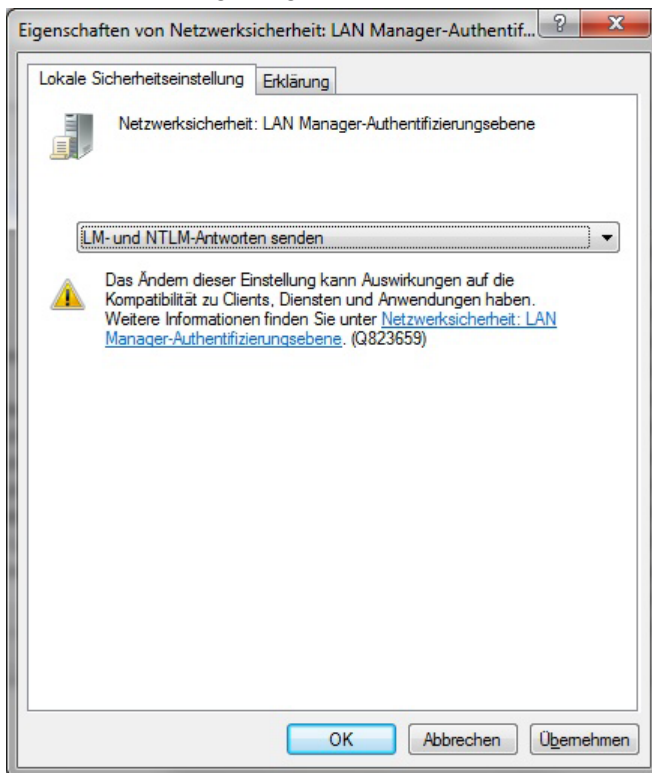


Abbildung 4-4: Standardeinstellung in Windows XP

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





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

4.3.4.5 Kerberos-Ticket (Integrierte Windows-Anmeldung) (ab V4.2R2)

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

Zur Anmeldung wird ein Kerberos-Ticket akzeptiert, das vom Web-Browser 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 Web-Browser zu aktivieren, da Kerberos einen nicht ganz standardkonformen HTTP-Header ("WWW-Authenticate: Negotiate") verwendet, den einige ältere Web-Browser als Fehler interpretieren. Um Kerberos bei allen Web-Browsern 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/fs4host.mydomain.net@MYDOMAIN.NET"
  keyTab="/opt/firstspirit4/conf/fs4host-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 fs4host-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/fs4host.mydomain.net@MYDOMAIN.NET \  
+rndpass -mapuser fs4host-HTTP \  
-crypto RC4-HMAC-NT -ptype KRB5_NT_PRINCIPAL \  
-out fs4host-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/fs4host.mydomain.net@MYDOMAIN.NET \  
+rndpass -mapuser fs4host-HTTP \  
-crypto AES256-SHA1 -ptype KRB5_NT_PRINCIPAL \  
-out fs4host-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 fs4host-HTTP
```

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

```
setspn -d HTTP/fs4host.mydomain fs4host-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/firstspirit4/conf/fs4host-HTTP.keytab`.

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

```
kinit -V -k -t fs4host-HTTP.keytab HTTP/fs4host.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-fs4host-HTTP-rc4.keytab
rkt krb5-fs4host-HTTP-aes256.keytab
wkt /opt/firstspirit4/conf/fs4host-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/firstspirit4/conf/fs-jaas.conf
```

Falls der interne Jetty Application-Server eingesetzt wird, muss in der Datei `/opt/firstspirit4/conf/fs-webapp.xml`



im Abschnitt `org.mortbay.jetty.nio.SelectChannelConnector` folgender Parameter hinzugefügt werden, da der HTTP-Request-Header bei Kerberos eine Größe von mehr als 4kByte erreichen kann:

```
<Set name='headerBufferSize'>10000</Set>
```

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/firstspirit4/log/fs-server.log` und `/opt/firstspirit4/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/firstspirit4/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 Web-Browser 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.





Sicherheitshinweis: Das *KerberosLoginModule* sollte aus Sicherheitsgründen in Produktivsystemen 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 Webstart gestartet wird. Für den JavaClient über Java Webstart ist das *NTLMLoginModule* (siehe Kapitel 4.3.4.4 Seite 78) 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 Anmeldenname verwendet, so dass die Zuordnung von Benutzerkonten zu Projekten auch in Projektexporten gewährleistet ist.





Ab FirstSpirit-Version 4.1 kann das automatische Anlegen von Benutzerkonten 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.webnonssso=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.webnonssso`
- Access-API: `JAAS.system`

4.3.4.9 Konfigurationsbeispiele

Standardkonfiguration:

In Verbindung mit der in Kapitel 12.3 aufgeführten Standardkonfiguration der Datei `fs-jaas.conf` ergibt sich z. B. für den JavaClient folgendes Anmeldeverfahren:

1. Beim Aufruf der FirstSpirit-Startseite über den Web-Browser 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 Web-Browser später weitergereicht wird.
2. Beim Starten des JavaClient über Webstart wird der zuvor erstellte Berechtigungsnachweis vom Web-Browser 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:dcl.e-spirit.de,dc2.e-spirit.de;:dcl.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/fs4.e-spirit.de@E-SPIRIT.DE"
//  keyTab="/opt/firstspirit4/conf/krb5-fs4-HTTP.keytab"
//  useKeyTab="true"
//  storeKey="true"
//  isInitiator="false"
//  doNotPrompt="true"
//  debug="true";
//};
```



Ausschnitt aus der Datei `/opt/firstspirit4/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 438). 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
#Tue May 16 08:11:53 CEST 2006
license.MAXPROJECTS=5
license.EXPDATE=15.01.2007
license.FEATURES=content,dataXML,generateComment,International,pac
kagepool,PageSpecificHeaders
license.ARCHIVE=1
license.VERSION=4
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. Dabei werden die deaktivierten Projekte ebenfalls 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 4.3.1.10 Seite 51) 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 "FirstSpirit Archivfunktion").

`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`: Ist hier der Wert "1" gesetzt, ist die Funktionalität "Dokumentengruppe" aktiviert, die es ermöglicht, Seitenreferenzen in



der Struktur-Verwaltung zu einer Gruppe zusammenzufassen und so ein Ergebnisdokument, beispielsweise eine PDF-Datei, zu erzeugen (weiterführende Dokumentation siehe "FirstSpirit Handbuch für Entwickler – Grundlagen").

`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 "WebEdit" aktiviert, die es ermöglicht, redaktionelle Inhalte direkt innerhalb der Vorschauseite im Browserfenster zu bearbeiten (weiterführende Dokumentation siehe "WebEdit Admin- und Entwicklerhandbuch").

`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 ab FirstSpirit Version 4.2R2 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 280), 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 AppCenter-Anwendungen, 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 *FirstSpirit Release Notes Version 4.2R4*, Kapitel 6.5.2 "license.APPTAB_SLOTS" Seite 145.)



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 439). 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 439).

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.mortbay=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:/FIRSTspirit4/log/err.log`

Beispiel:

```
# file
log4j.rootCategory=ERROR, file
log4j.appender.file=org.apache.log4j.RollingFileAppender
log4j.appender.file.File=D:/FIRSTspirit4/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.



4.3.7 Konfiguration des Webservers (fs-webapp.xml)

Für die Vorschauerstellung, die Server- und Projektkonfiguration und das Arbeiten mit WebEdit 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 106.

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 442). 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 12.4.



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 444).

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

- Konnektoren

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



- Webanwendungen
- Protokollierung

Weitere Konfigurationsmöglichkeiten 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 Web-Browser weiterhin gültig sind, nachdem die Verbindung automatisch erneut aufgebaut wird. Zur Konfiguration der Zeitüberschreitung für die FirstSpirit-Sitzung siehe Kapitel 4.3.1.15.

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_WEBEDIT_PATH" /></Arg>
  <Arg><SystemProperty name="WEBAPP_WEBEDIT_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 LDAP-Benutzer ist oder nicht. Im FirstSpirit-Server lassen sich verschiedene LDAP-Konfigurationen (sogenannte Sections) anlegen (siehe Kapitel 4.3.4 Seite 76) und konfigurieren (siehe Kapitel 4.3.1.11 Seite 52). Ein LDAP-Benutzer wird dabei mit genau einer Section assoziiert (siehe 7.2.4.2 Seite 219). 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 4.3.1.11 Seite 52.
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 298). Login und Passwort des externen LDAP-Benutzers werden nach dem erstmaligen Einloggen automatisch nach FirstSpirit übernommen. In früheren FirstSpirit-Versionen war es aus diesem Grund möglich, sich nach einer erfolgreichen ersten 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 4.3.1.11 Seite 52).

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 4.3.1.11 Seite 52):

- Benutzername: Name des FirstSpirit Benutzers.
- EMail: 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
```

¹² <http://www.openssl.org/>



Zusätzlich muß 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 der 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.

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-Web-Anwendungen 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.

Falls in FirstSpirit-Projekten Web-Anwendungen nach dem Standard Java EE 5 eingesetzt werden sollen, muss außerdem der Jetty-Webserver durch eine andere Servlet-Engine ersetzt werden, beispielsweise Tomcat 6, wie in Kapitel 4.5.2 beschrieben. Jetty bietet in der von FirstSpirit verwendeten Version 6.1 nur J2EE 1.4 mit der Ergänzung JDK 1.6.

Falls als Protokoll zwischen externem Webserver und Servlet-Engine AJP verwendet werden soll, muss ebenfalls eine andere Servlet-Engine verwendet werden, da der Jetty-Webserver nicht vollständig kompatibel zu AJP 1.3 ist. Dazu siehe Kapitel 4.5.2.



Eine weitere Konfigurationsmöglichkeit ist die Lastverteilung auf mehrere Servlet-Engines (siehe Kapitel 4.5.4 Seite 121).



4.5.1 Apache HTTP-Server mit Servlet-Engine Jetty

Der Apache HTTP-Server in Version 2.2 wird in dieser Konfiguration in Verbindung mit der in FirstSpirit integrierten Servlet-Engine des Jetty-Webservers 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 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-Web-Anwendungen über `mod_proxy_http` an Jetty weitergeleitet werden. Von den unter Jetty installierten FirstSpirit-Web-Anwendungen 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 `firstspirit4/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://fs4.yourdomain.net`.

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



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 fs4.yourdomain.net           # EXAMPLE
ServerAlias fs4                         # EXAMPLE

LogLevel warn
CustomLog /var/log/apache2/fs4.access.log combined # EXAMPLE
ErrorLog /var/log/apache2/fs4.error.log      # EXAMPLE

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

DocumentRoot /opt/firstspirit4/web        # EXAMPLE

DirectoryIndex index.html index.jsp index.php

<Location />
  order allow,deny
  allow from all
</Location>

# 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} !^/fs4preview/preview_cache
RewriteCond %{REQUEST_URI} !^/fs4webedit/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://fs4server: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 `firstspirit4/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 wird in dieser Konfiguration in Verbindung mit der Servlet-Engine Tomcat 6 verwendet. Als Protokoll zwischen Apache und Tomcat wird AJP eingesetzt.

Gegenüber der in FirstSpirit integrierten Servlet-Engine Jetty bietet Tomcat 6 den Standard Java EE 5 für Web-Anwendungen.



Zurzeit existieren als AJP-Connector für Apache `mod_jk` und `mod_proxy_ajp`¹⁴. 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 4.5.4 Seite 121).

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-Web-Anwendungen über `mod_proxy_ajp` an Tomcat weitergeleitet werden. Von den unter Tomcat installierten FirstSpirit-Web-Anwendungen wird dann bei Bedarf automatisch zusätzlich eine interne HTTP-Verbindung zur Auslieferung spezieller Dateitypen wie PHP oder ASP aufgebaut.

Bevor Apache und Tomcat konfiguriert werden, ist die FirstSpirit-Konfiguration zu ändern. In der Datei `firstspirit4/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 Apache, beispielsweise `http://fs4.yourdomain.net`, verwendet wird:

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

Anschließend den FirstSpirit-Server herunterfahren, das Verzeichnis `firstspirit4/web/fs4root` entfernen und den FirstSpirit-Server wieder starten.

Nun erfolgt 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.

¹⁴ http://httpd.apache.org/docs/2.2/mod/mod_proxy.html



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_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 fs4.yourdomain.net           # EXAMPLE
ServerAlias fs4                         # EXAMPLE

LogLevel warn
CustomLog /var/log/apache2/fs4.access.log combined # EXAMPLE
ErrorLog /var/log/apache2/fs4.error.log      # EXAMPLE

ServerSignature off
UseCanonicalName off
AddDefaultCharset off
ProxyRequests off
RewriteEngine on

DocumentRoot /opt/firstspirit4/web        # EXAMPLE

DirectoryIndex index.html index.jsp index.php

<Location />
    order allow,deny
    allow from all
</Location>

# 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 # EXAMPLE
</Proxy>

# forward requests for FirstSpirit-Webapps to Servlet-Engine
RewriteCond %{REQUEST_URI} !^/balancer-manager
RewriteCond %{REQUEST_URI} !^/server-status
RewriteCond %{REQUEST_URI} !^/fs4preview/preview_cache
RewriteCond %{REQUEST_URI} !^/fs4webedit/preview_cache
RewriteCond %{REQUEST_URI} !^/fs4staging.*php$
RewriteRule ^/(.*) balancer://fsajp/$1 [proxy,last]

</VirtualHost>
```

Die Tomcat-Konfiguration wird im nächsten Kapitel (4.5.3) beschrieben.



4.5.3 Externe Servlet-Engine (Tomcat)

Der in FirstSpirit 4 integrierte Webserver Jetty 6.1 bietet nur J2EE 1.4 mit Ergänzung JDK und ist nicht vollständig kompatibel zu AJP. Falls in FirstSpirit-Projekten Web-Anwendungen nach Java EE 5 verwendet werden sollen oder AJP zur Anbindung eines HTTP-Servers (z. B. Apache), bietet sich an, Tomcat 6¹⁵ zu verwenden, dessen Konfiguration für FirstSpirit im folgenden Abschnitt beschrieben wird.

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 `firstspirit4/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://fs4.yourdomain.net:8080`, verwendet wird:

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

Anschließend den FirstSpirit-Server herunterfahren, das Verzeichnis `firstspirit4/web/fs4root` entfernen und den FirstSpirit-Server wieder starten.

Die Installation des Tomcat erfolgt in diesem Beispiel nach `/opt/firstspirit4/tomcat`. Falls die Installation in ein anderes Verzeichnis erfolgt, muss im nachfolgenden Konfigurationsbeispiel an allen Stellen `"${catalina.home}/.."` durch `"/opt/firstspirit4/"` 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:

¹⁵ <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.



```
<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>
```



Um das Auflisten der Verzeichnisse für den Bereich `/fs4staging` zu aktivieren, 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:

```
<?xml version='1.0' encoding='utf-8'?>
<tomcat-users>
  <role rolename="manager"/>
  <user username="Admin" password="tomcat-password" roles="manager"/>
</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`¹⁶ durch folgenden Inhalt ersetzen, um das sofortige Prüfen auf Dateiänderungen (HTML und JSP) zu aktivieren und um die TCP-Verbindung der FirstSpirit-Webapplikationen zum FirstSpirit-Server zu konfigurieren:

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

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

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

  <Parameter name="firstspirit.host" value="fs4server.mydomain.net"
  override="false" />
  <Parameter name="firstspirit.port" value="1088" override="false" />

</Context>
```

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

Bis FirstSpirit-Version 4.2 Release 0 einschließlich:

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

Ab FirstSpirit-Version 4.2 Release 2 muss dieser Eintrag wie folgt lauten:

```
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**):

Bis FirstSpirit-Version 4.2 Release 0 einschließlich:

```
common.loader=${catalina.home}/lib,${catalina.home}/lib/*.jar,
d:/Programme/FirstSpirit4/server/lib/fs-server.jar,
d:/Programme/FirstSpirit4/shared/lib/*.jar
```

¹⁶ 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.



Ab FirstSpirit-Version 4.2 Release 2 muss dieser Eintrag wie folgt lauten:

```
common.loader=${catalina.home}/lib,${catalina.home}/lib/*.jar,  
d:/Programme/FirstSpirit4/data/fslib/fs-webrt.jar,  
d:/Programme/FirstSpirit4/shared/lib/*.jar
```

Die Datei `tomcat/lib/log4j.properties` mit folgendem Inhalt erstellen, um das Logging der FirstSpirit-Web-Anwendungen 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
```

Ab FirstSpirit-Version 4.2 Release 2: 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 von Tomcat statt Jetty wird an allen Stellen in der Server- und Projektkonfiguration (in den Server-Eigenschaften vgl. Kapitel 7.3.15 und 7.3.16 ab Seite 258, in den Projekt-Eigenschaften Kapitel 7.4.18 Seite 319) 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 4.6.2 (Seite 127) beschriebenen Anforderungen an die Konfiguration eines Application-Servers. Bei Tomcat wird zur Konfiguration der Java-VM die Shell-Umgebungsvariable `CATALINA_OPTS` verwendet, die in der Datei `.bashrc` (oder vergleichbarer) definiert wird. Zusätzlich sollte die Umgebungsvariable `CATALINA_PID` definiert werden, also der Pfad einer PID-Datei angegeben werden, z. B. `$HOME/.catalina.pid`, so dass Tomcat definiert über folgenden Aufruf heruntergefahren werden kann, ohne dass ein



Prozess "hängenbleibt":

```
catalina.sh stop -force
```



4.5.4 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-Web-Anwendungen ihren Zustand nicht persistent serialisieren können.

Die statischen Dateien der FirstSpirit-Web-Anwendungen, die von FirstSpirit automatisch bei jeder Aktualisierung nach `/opt/firstspirit4/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
  BalancerMember ajp://tomcat-host2:8009 loadfactor=100 route=node2
  BalancerMember ajp://tomcat-host3:8009 loadfactor=100 route=node3
  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 Route-Namen der Servlet-Engine ersetzen.

Außerdem alle Vorkommen von `${catalina.home}/../`, wo auf die FirstSpirit-Verzeichnisse `web` und `server/lib`, **ab FirstSpirit-Version 4.2 Release 2** zusätzlich `data/fslib`, in `conf/server.xml` und `conf/catalina.properties` verwiesen wird, durch `/import/firstspirit4/` ersetzen.

Das Verzeichnis `/opt/FirstSpirit4` des FirstSpirit-Servers über NFS read-only allen Servlet-Engines als lokalen Mountpoint `/import/firstspirit4` bereitstellen.

Die Preview-Cache-Verzeichnisse müssen in dieser Konfiguration ausserhalb des nur lesbaren Dateisystems `/import/firstspirit4` liegen, da sie von den Web-Anwendungen `fs4preview` und `fs4webedit` beschrieben werden. Dazu werden beide Preview-Cache-Verzeichnisse über Symlinks auf jeweils lokale Dateisystem umgeleitet. Auf dem FirstSpirit-Servers dazu Folgendes aufrufen:

```
mkdir -p /opt/firstspirit4/previewcache/fs4preview
mkdir -p /opt/firstspirit4/previewcache/fs4webedit
rm -rf /opt/firstspirit4/web/fs4preview/preview_cache
rm -rf /opt/firstspirit4/web/fs4webedit/preview_cache
ln -s /opt/firstspirit4/previewcache/fs4preview \
    /opt/firstspirit4/web/fs4preview/preview_cache
ln -s /opt/firstspirit4/previewcache/fs4webedit \
    /opt/firstspirit4/web/fs4webedit/preview_cache
```

Auf jeder Tomcat-Instanz werden die lokalen Cache-Verzeichnisse erstellt:

```
mkdir -p /opt/firstspirit4/previewcache/fs4preview
mkdir -p /opt/firstspirit4/previewcache/fs4webedit
```

Andere Verzeichnisse unterhalb von `/opt/firstspirit4` werden auf Seite der Tomcat-Instanz nicht benötigt, da sie bereits im Lesezugriff über NFS auf `/import/firstspirit4` erreichbar sind.

Zusätzlich muss in der Konfigurationsdatei `/opt/firstspirit4/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://fs4.yourdomain.net/balancer-manager/` die Erreichbarkeit der Servlet-Engines prüfen.

Die FirstSpirit-Startseite kann nun beispielsweise über `http://fs4.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/firstspirit4/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 7.3.15 Seite 258) können nun beliebige Webserver-Steuerungen hinzugefügt und konfiguriert werden. Zunächst muss der gewünschte externe Webserver hinzugefügt werden (siehe



7.3.15.3 Seite 263). 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 4 auf Seite 268 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 4 (Seite 268 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 4 Seite 268).

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 `firstspirit4/server/lib/fs-server.jar` (**ab FirstSpirit-Version 4.2 Release 2** `firstspirit4/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 Web-Anwendungen unter `WEB-INF/lib` nicht geladen werden können. Die Datei `fs-server.jar` (**ab FirstSpirit-Version 4.2 Release 2** `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-server.jar` (**ab FirstSpirit-Version 4.2 Release 2** `fs-webrt.jar`) angegeben, zum Beispiel `/opt/firstspirit4/server/lib/fs-server.jar` (**ab FirstSpirit-Version 4.2 Release 2** `/opt/firstspirit4/data/fslib/fs-webrt.jar`). Falls der



FirstSpirit-Server auf einem anderen Server betrieben wird als Websphere, empfiehlt es sich, das Verzeichnis `/opt/firstspirit4/server` (**ab FirstSpirit-Version 4.2 Release 2** `opt/firstspirit4/data/fslib`) über NFS an Websphere weiterzugeben, um die Datei `fs-server.jar` (**ab FirstSpirit-Version 4.2 Release 2** `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 (`fs4preview`), WebEdit (`fs4webedit`) oder Staging (`fs4staging`) 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`

In der Datei `fs-server.conf` muss der Parameter `preview.cacheFileWithTimestamp=*` gesetzt werden (für weitere Informationen zu diesem Parameter siehe Kapitel 4.3.1.9 Seite 49).



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 Sun 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!

4.6.1.1 Logging-Konfiguration für FirstSpirit-Web-Anwendungen unter Websphere

Zunächst im Websphere-Interface unter "Server > Anwendungsserver > Servername



> Prozessdefinition > Java Virtual Machine > Benutzerdefinierte Merkmale"
folgenden Parameter hinzufügen, um Fehler in der Log-Konfiguration anzuzeigen:

```
Name: org.apache.commons.logging.diagnostics.dest  
Wert: STDOUT
```

Diese Einstellung sollte nur temporär für den nächsten Start der Web-Anwendungen während dieser Konfigurationsschritte aktiv sein, da ansonsten zu viele Daten nach `SystemOut.log` geschrieben werden. Nachdem das Logging erfolgreich aktiviert wurde, sollte dieser Parameter wieder entfernt werden.

Die folgende Konfiguration wird dauerhaft beibehalten:

Die Datei `commons-logging.jar` in Version 1.1x von http://commons.apache.org/downloads/download_logging.cgi herunterladen, als lokale Datei für den Websphere-Server erreichbar speichern und über die Websphere-Adminoberfläche unter "Umgebung > Gemeinsame Bibliotheken" mit Namen "FirstSpirit Commons-Logging" hinzufügen.

Nun entweder unter "Anwendungsserver > Servername > Klassenlader > Classloader_ > Bibliotheksreferenz" über einen neuen Klassenlader (Parent-First) global eine Referenz auf die gemeinsame Bibliothek "FirstSpirit Commons-Logging" hinzufügen
oder die Referenz später für jede Web-Anwendung einzeln hinzufügen, nachdem die Web-Anwendungen installiert wurden.

Auf dem FirstSpirit-Server unterhalb von `firstspirit4/web/fs4preview`, `firstspirit4/web/fs4root` und `firstspirit4/web/fs4webmon` folgende Dateien mit dem jeweils angegebenen Inhalt hinzufügen:

Datei `META-INF/services/org.apache.commons.logging.LogFactory`:

```
org.apache.commons.logging.impl.Log4jFactory
```

Datei `WEB-INF/classes/log4j.properties`:

```
log4j.rootCategory=INFO, stdout  
log4j.appender.stdout=de.espirit.firstspirit.server.logging.FSAppender  
log4j.appender.stdout.consoleLogging=true  
log4j.appender.stdout.plainLogging=false  
log4j.appender.stdout.datedLogging=false
```

Der Log-Level kann auf Testsystemen von INFO auf DEBUG umgestellt werden.

Die Datei `log4j-1.2.14.jar` in Version 1.2.x von <http://logging.apache.org/log4j/1.2/download.html> herunterladen und nach WEB-



INF/lib/ kopieren.

Das Logging erfolgt mit diesen Einstellungen in die Datei `SystemOut.log` der Websphere-Instanz. Diese Datei wird in den Standardeinstellungen des Websphere-Servers automatisch bei Erreichen eines bestimmten Limits rotiert.

Log-Meldungen der FirstSpirit-Web-Anwendungen sind mit dem Paketnamen `de.espirit.*` gekennzeichnet.

Die Web-Anwendungen sind nun konfiguriert und können im nächsten Schritt im Websphere-Server installiert werden. Nach dem Installieren der Web-Anwendungen muss der Websphere-Server neu gestartet werden, um das Logging zu aktivieren!



Zur Konfiguration der Java-VM des Websphere-Servers bitte das nachfolgende Kapitel (4.6.2) beachten.

4.6.2 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 beschriebenen. Folgende Parameter dienen als Anhaltspunkt zur Konfiguration, falls die Java-VM von Sun 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` dient 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://java.sun.com/j2se/1.5.0/docs/guide/management/jconsole.html>) zur interaktiven Abfrage verwendet werden. Es ist Bestandteil jedes JDK (nicht JRE) von Sun.

Im Testbetrieb sollten folgende zusätzliche Parameter ergänzt werden, um die Auslastung des Garbage Collectors überwachen zu können. Falls diese Überwachung im Produktionsbetrieb gewünscht ist, muss statt des Dateinamens bei `xloggc` ein Verfahren eingesetzt werden, die Ausgabe zu archivieren, um nicht eine ständig wachsende Datei zu erhalten:

```
-XX:+PrintGCTimeStamps \  
-XX:+PrintGCDetails \  
-XX:+PrintGCApplicationStoppedTime \  
-Xloggc:$HOME/tomcat/logs/tomcat-gc.log
```

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 131). Falls ein eigenes Testzertifikat mit anderem Hostnamen erstellt werden soll, kann dies über folgenden Aufruf geschehen:

```
keytool -genkeypair -alias fs4.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, 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 131).

Um das selbstsignierte Test-Zertifikat auf Seiten des FirstSpirit-JavaClients verwenden zu können, falls dieser nicht über Java-Webstart aufgerufen wird oder das Zertifikat von Webstart 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`¹⁷ 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 fs4.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. `fs4.yourdomain.net`.

Im nächsten Schritt muss eine Zertifizierungsanforderung erzeugt werden:

```
keytool -certreq -keystore conf/fs-keystore.jks -storepass changeit -alias fs4.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 fs4.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

¹⁷ <http://www.openssl.org/>



wird von keytool die Fehlermeldung "Failed to establish chain from reply" 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:



```
<Call name="addConnector">
  <Arg>
    <New class="org.mortbay.jetty.security.SslSelectChannelConnector">
      <Set name="port">8443</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>
      <Set name="keystore"><SystemProperty name="cmsroot" />/conf/fs-
keystore.jks</Set>
      <Set name="password">changeit</Set>
      <Set name="keyPassword">changeit</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 die Zertifikatserstellung über `openssl` gewählt wurde, muß folgende Option hinzugefügt werden

```
<Set name="KeystoreType">pkcs12</Set>
```

und der Name des Zertifikatsspeichers bei `keystore` von `fs-keystore.jks` in `fs-keystore.p12` geändert werden.

Nach einem Neustart des FirstSpirit-Servers ist die Startseite nun zusätzlich über <https://fs4.yourdomain.net:8443> neben <http://fs4.yourdomain.net:8000> verfügbar.

Weitere Informationen zur HTTPS-Konfiguration des von FirstSpirit verwendeten Webservers Jetty finden Sie unter:

<http://docs.codehaus.org/display/JETTY/How+to+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 werden folgende Ziele erreicht:

1. Es 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.
2. Die Unterstützung neuer Datenbanksysteme kann auf einfache Art und Weise durch die Implementierung der Datenbankabstraktionsschicht realisiert werden.
3. In der Regel kann ein Projektexport bzw. -import von einem Datenbanksystem auf ein anderes Datenbanksystem erfolgen.

Die Minimalvoraussetzung, um eine Datenbank-Abstraktionsschicht zu realisieren, ist ein JDBC-Treiber Version 2.0, der auf dem Betriebssystem des Servers lauffähig ist.

4.8.1 Ablage der JDBC-Treiberdateien

Es bestehen unterschiedliche Möglichkeiten, die JDBC-Treiberdateien in den FirstSpirit-Server zu integrieren:

1. Über das Verzeichnis `shared/lib`:
Die entsprechenden Treiberdateien müssen in das Verzeichnis `firstspirit4/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.
2. Als FirstSpirit-Modul: **ab FirstSpirit Version 4.2R4 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 XML-Datei zur Namensdefinition in einer FSM-Datei (ZIP-Archiv) zusammengefasst. In der Layer-Konfiguration wird über den zusätzlichen Parameter `module` (siehe Kapitel 4.8.4.2 Seite 150) dann auf dieses FirstSpirit-Modul verwiesen.



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



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

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 134)
- `module.xml`-Datei (siehe Kapitel 4.8.2.2 Seite 135)

optional:

- `web.xml`-Datei (siehe Kapitel 4.8.2.2 Seite 135)

Diese Dateien werden in einer FSM-Datei (ZIP-Archiv) zu einem FirstSpirit-Modul zusammengefasst (siehe Kapitel 4.8.2.3 Seite 139).

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 156 sowie die anderen Unterkapitel in Kapitel 4.

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 8.2**-Datenbank dar:

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

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

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

  </components>

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

Das Modul ist zweigeteilt: ein Teil definiert die Ressourcen für den FirstSpirit-Server, die so genannte "Server-Komponente" (innerhalb des `<public>`-Tags), der andere



für Web-Anwendungen, die so genannte "Web-Anwendungs-Komponente" (innerhalb des `<web-app>`-Tags). Dieser JDBC-Treiber kann somit im FirstSpirit-Server **und** in Web-Anwendungen genutzt werden. Wird der Treiber nur für den Server benötigt, kann die `<web-app>`-Definition entfallen.



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

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

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

Der für die Server-Komponente vergebene Name wird in der Modul-Übersicht des FirstSpirit-Servers verwendet (siehe Abbildung 4-7) und muss auch in der Datenbank-Layer-Konfiguration angegeben werden (siehe Kapitel 4.8.3.2 Seite 141). Der für die Web-Anwendungs-Komponente vergebene Name wird auch in den Projekt-Eigenschaften, Bereich "Web-Komponenten" verwendet (siehe Abbildung 4-11). Für eine bessere Übersicht wurden im Beispiel `_Server` bzw. `_WebApp` an den Namen für die Server- bzw. die Web-Anwendungs-Komponente angehängt.

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

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

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

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

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



verwendet werden:

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

`postgresql` (PostgreSQL)

`oracle` (Oracle)

`mssql` (Microsoft SQL Server)

`mysql` (MySQL)

`db2` (IBM DB2)

`derby` (Apache Derby)

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

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

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

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

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

für PostgreSQL oder

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

für Oracle.

(Siehe auch Parameter `layerclass` in Kapitel 4.8.4.1 Seite 149 sowie die Kapitel mit den datenbanktyp-spezifischen Beispielkonfigurationen in Kapitel 4.8.7 Seite 158.)

`<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 149 sowie die Kapitel mit den datenbanktyp-spezifischen Beispielkonfigurationen



in Kapitel 4.8.7 Seite 158.)

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

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

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

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

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

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

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

  </components>

  <configuration>
    <layerclass>de.espirit.or.impl.derby.DerbyLayer</layerclass>
```



```
<DRIVER>org.apache.derby.jdbc.ClientDriver</DRIVER>
</configuration>
</module>
```

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

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

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

Wird der Server-Komponententeil benötigt, also interner Jetty- und externer Tomcat-Web-Server gleichzeitig eingesetzt, ist der Datenbank-Layer anzupassen (siehe Kapitel 4.8.3.4.1 Seite 144).



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 ("Server-Komponente") verwendet, müssen die Dateien in folgender Verzeichnisstruktur abgelegt werden (vgl. Kapitel 4.8.2.1 Seite 134 und Kapitel 4.8.2.2 Seite 135):

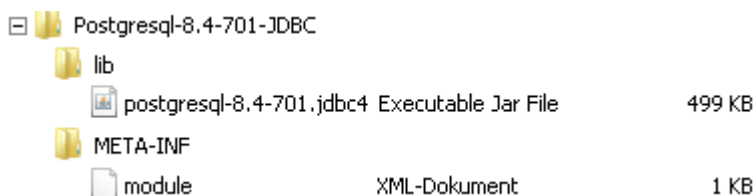
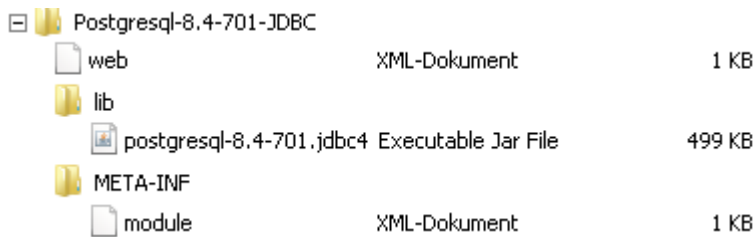


Abbildung 4-5: Verzeichnisstruktur Server-Komponente

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





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

Abbildung 4-6: Verzeichnisstruktur Server- und Web-Anwendungs-Komponente

Um nun ein gültiges FirstSpirit-Modul zu erhalten, muss aus den Inhalten des übergeordneten Ordners ("Postgresql-8.4-701-JDBC") eine ZIP-Datei erstellt werden. Der übergeordnete Ordner darf nicht in der ZIP-Datei enthalten sein. Diese ZIP-Datei muss dann in *.fsm umbenannt werden. Würde im Beispiel von Abbildung 4-6 der Ordnername als Dateiname übernommen, sollte die Modul-Datei schließlich `Postgresql-8.4-701-JDBC.fsm` heißen.

Alternativ kann die Modul-Datei über folgenden Befehl erstellt werden:

```
jar cvf Postgresql-8.4-701-JDBC.fsm -C Postgresql-8.4-701-JDBC .
```

Das Programm "jar" ist Bestandteil jedes JDK und je nach Betriebssystem und Installation beispielsweise unter `c:\programme\jdk1.6.0\bin\jar.exe` oder `/opt/jdk1.6.0/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 134 beschrieben erstellt, muss es auf dem FirstSpirit-Server installiert werden. Dies wird über die Anwendung zur Server- und Projektkonfiguration vorgenommen.

Dazu wird in den Server-Eigenschaften im Bereich "Module" der Button "Installieren" angeklickt. Im folgenden Dialog kann die Treiber-Modul-Datei aus dem lokalen Dateiverzeichnis ausgewählt und hochgeladen werden. Die erfolgreich installierte Datei wird anschließend in der Übersicht als Modul mit Namen (hier: `PostgreSQL_JDBC_Driver_8_4-701`, siehe Kapitel 4.8.2.2 Seite 135, Tag `<name>`), Version (Tag `<version>`) und der enthaltenen Komponente (Tag `<components>`) angezeigt:



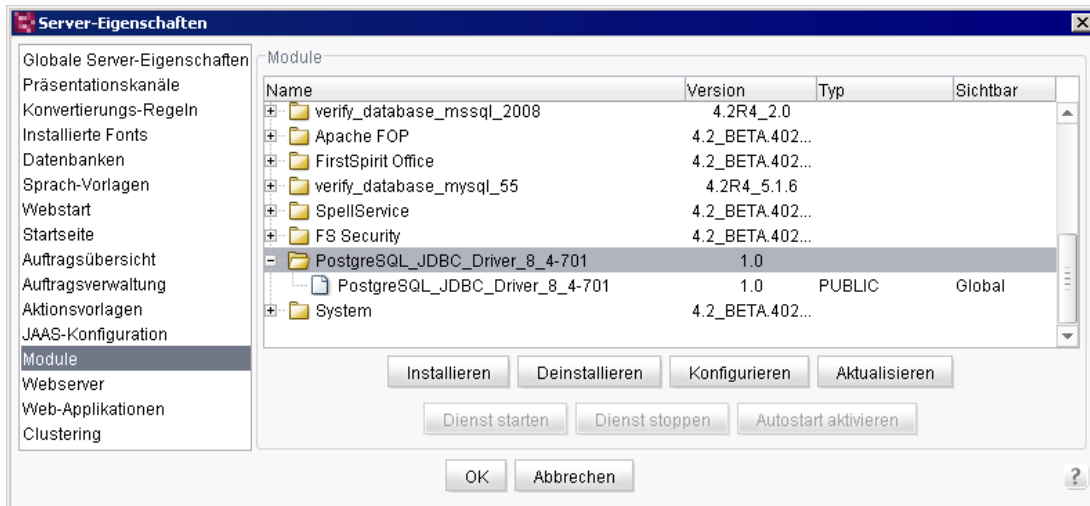


Abbildung 4-7: Server-Eigenschaften – JDBC-Treiber als Modul

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

Hier braucht keine weitere Konfiguration vorgenommen werden.

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:



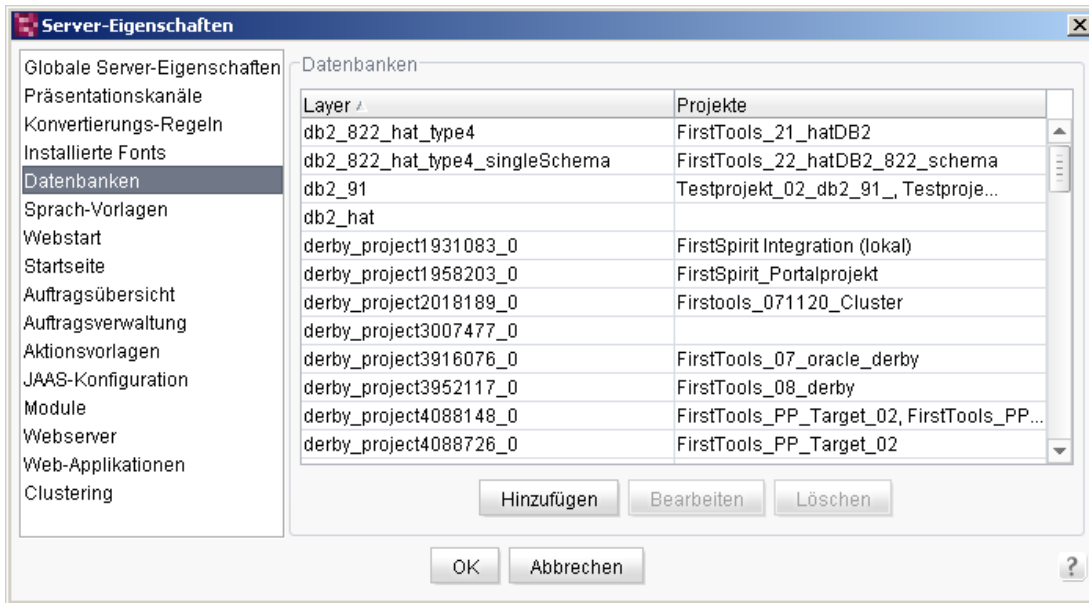


Abbildung 4-8: Server-Eigenschaften – Datenbanken

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=PostgreSQL_JDBC_Driver_8_4-701
```



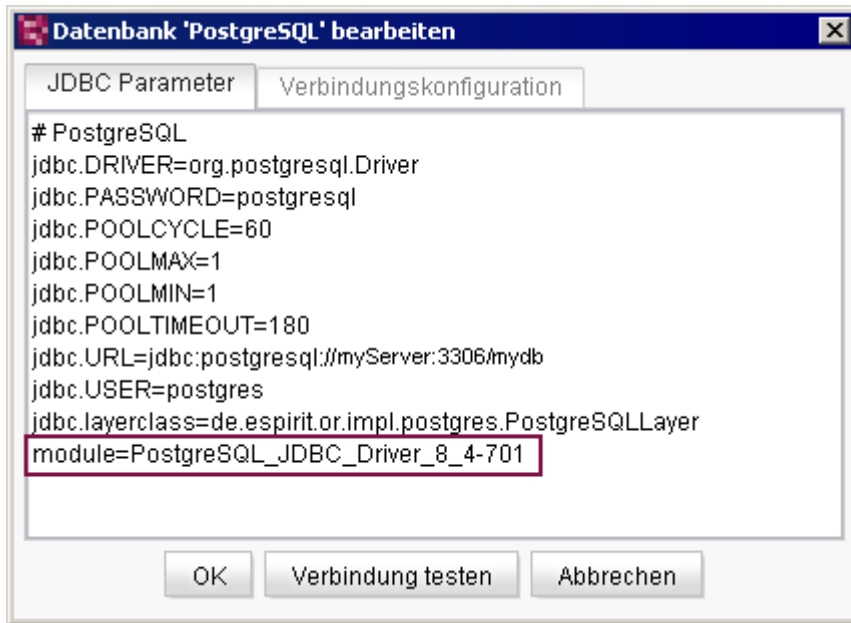


Abbildung 4-10: Datenbank-Konfiguration mit `module`-Parameter

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

4.8.3.3 Bei Verwendung in Web-Anwendungen

Soll die Datenbank in einer Web-Anwendung verwendet werden, muss das Modul in der gewünschten Web-Komponente hinzugefügt werden. Dies wird in den Projekteigenschaften im Bereich "Web-Komponenten" vorgenommen. Mit einem Klick auf "Hinzufügen" öffnet sich eine Liste, aus der die auf dem Server zur Verfügung stehenden Modul-Komponenten ausgewählt werden können. Hier muss die Komponente des JDBC-Treiber-Moduls ausgewählt werden. Der Name ergibt sich dabei aus dem in der `<web-app>`-Definition über `<name>` definierten Wert (siehe Kapitel 4.8.2.2 Seite 135):



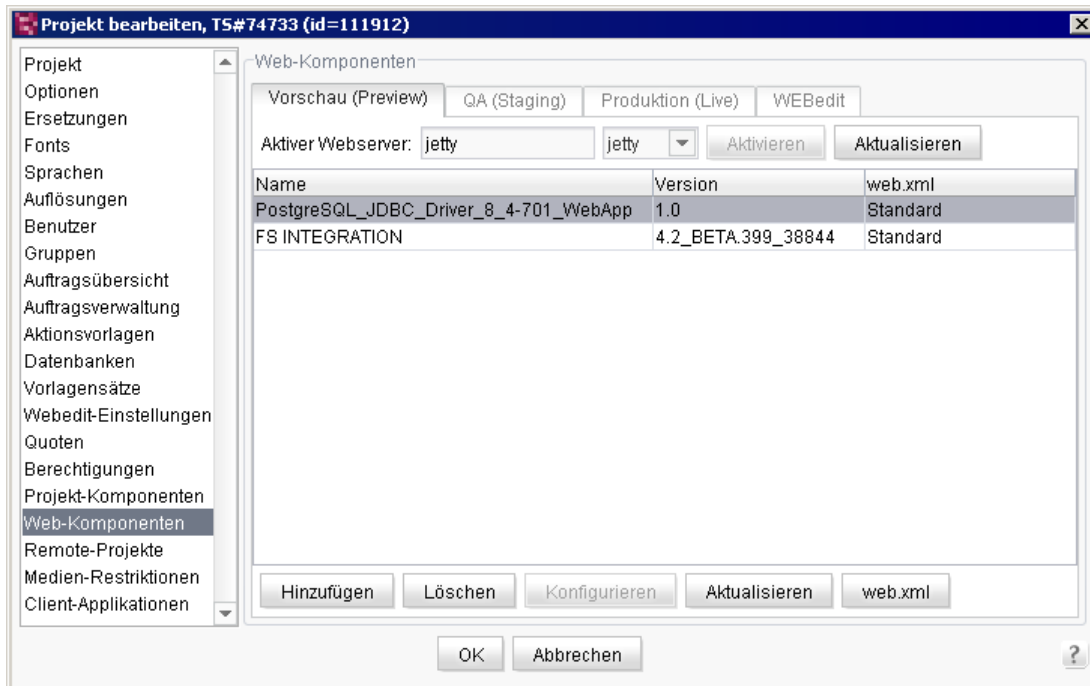


Abbildung 4-11: Projekt-Eigenschaften – JDBC-Treiber als Web-Komponente

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

4.8.3.4 Bei Verwendung der Derby-Datenbank in Web-Anwendungen

Wird eine Derby-Datenbank in Web-Anwendungen (z. B. im FirstSpirit-Modul Integration) verwendet, muss das JDBC-Treiber-Modul ebenfalls, wie in Kapitel 4.8.3.3 Seite 143 beschrieben, der Web-Anwendung hinzugefügt werden.



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

4.8.3.4.1 Beispiel: Modul "FirstSpirit Integration"

Bei der Verwendung des Moduls FirstSpirit Integration mit einem Tomcat-Web-Server muss für jedes Schema die Verbindungskonfiguration angepasst werden. Der Zugriff kann in diesem Fall nur über den TCP-Port der Derby-Datenbank erfolgen. Hierzu muss der Parameter `internalDB.port` in der Konfigurationsdatei `fs-server.conf` angegeben werden.



In der Datenbank-Layer-Konfiguration (siehe Kapitel 4.8.3.2 Seite 141) sind zusätzlich folgende Parameter für jedes Schema anzupassen:

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

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

dann

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

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

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



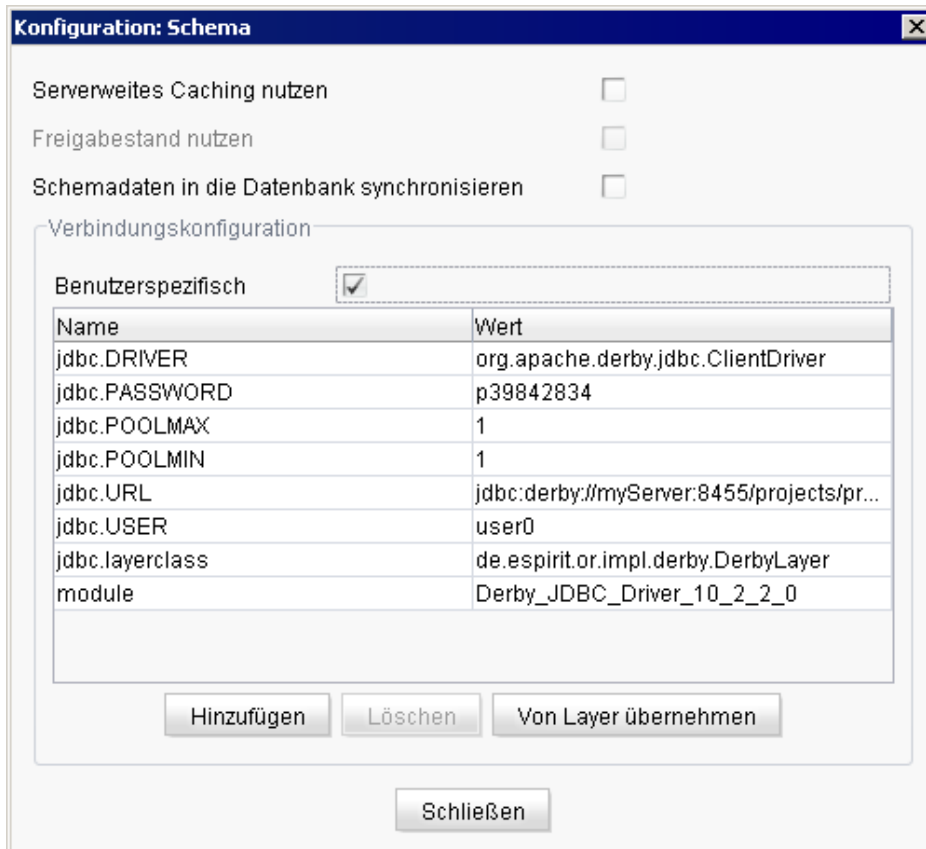


Abbildung 4-12: FirstSpirit Integration – Datenbank-Konfiguration

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

Wird neben dem externen Tomcat- auch der interne Jetty-Web-Server verwendet, so ist auch der Datenbank-Layer anzupassen (siehe Kapitel 4.8.3.2 Seite 141).

4.8.3.4.2 Individuelle Modul-Implementierung

Die Anpassungen, die für das Beispiel "FirstSpirit Integration" (siehe Kapitel 4.8.3.2 Seite 141) 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 Datenbankanbindung des FirstSpirit-Servers erfolgt über die Server- und Projektkonfiguration (siehe Kapitel 7.3.6 Seite 238) oder direkt über den Eintrag "DATABASES" in der Konfigurationsdatei `fs-database.conf` (siehe Kapitel 4.3.3 Seite 75). Empfohlen wird das Bearbeiten der Datenbankanbindung über die Server- und Projektkonfiguration (siehe Kapitel 7.3.6 Seite 238), da hier alle Änderungen automatisch in die `fs-database.conf` geschrieben und auf dem Server aktualisiert werden. Außerdem ist es möglich, die konfigurierte Datenbankanbindung zu testen (siehe Abbildung 7-41).

Die FirstSpirit Datenbankanbindung 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`)
- Datenbank-Konfiguration über die Server- und Projektkonfiguration (empfohlen) oder direkt über die Datei `firstspirit4/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 149.

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 150.



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 133). 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. **Ab FirstSpirit Version 4.2R4** wird die Einbindung von JDBC-Treiberdateien als FirstSpirit-Modul empfohlen (siehe auch Kapitel 4.8.2 Seite 134).

```
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 158).

`<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 Datenbankbindung kommt (siehe Kapitel 7.3.6 Seite 238).

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 Standard-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
```

Für Standard-Layer in PostgreSQL-Datenbanken muss für diesen Parameter immer `public` angegeben werden!

`<database>.jdbc.POOLMAX`: FirstSpirit nutzt für jede Verwendung, z. B. einen Schema-Knoten in einem Projekt, eine deployte Web-Applikation (z. B. FirstSpirit Integration) 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 `ORMapper` innerhalb eines `Web-Containers` oder innerhalb eines `Applikation-Servers`, kann eine `Datenbankverbindung` über einen `DataSource` geholt werden. Damit nutzt der `ORMapper` 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
```

`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 134.

```
database.module=PostgreSQL_JDBC_Driver_8_4-701
```

4.8.4.3 Beschreibung der Oracle-spezifischen Parameter

`<database>.jdbc.oracle.TABLESPACES`: Beim Anlegen einer Tabelle unter `Oracle`, kann mit diesem Parameter der für diese Tabelle benutzte `Tabellenbereich` angegeben werden. Wird dieser optionale Parameter nicht angegeben, wird standardmäßig der Wert `"USERS"` gesetzt.

```
Database.jdbc.oracle.TABLESPACES=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 158.

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.



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 speziell ab Version 4.2R4 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
```

4.8.6.5 Microsoft SQL-Server 2000

Unicode: Keine Unterstützung von UNICODE. Obwohl der SQL-Server Unicode unterstützt, ist es nicht möglich, durch die standardmäßige JDBC-ODBC-Bridge Unicode Zeichen in der Datenbank abzuspeichern. Dieses Problem kann eventuell durch den Einsatz einer anderen Bridge oder eines nativen JDBC-Treibers umgangen werden.

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 154.

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:

```
jdbc.DRIVER=COM.ibm.db2.jdbc.net.DB2Driver
jdbc.layerclass=de.espirit.or.impl.db2.DB2Layer
jdbc.URL=jdbc:db2://myServer:db2javaPort/myDB
jdbc.USER=myUser
jdbc.PASSWORD=myPass
jdbc.SCHEMA=myDB
```

Treiber: db2java.zip (muss exakt zum verwendeten DB2-Server passen)

Port: DB2-Java-Connector (wird bereitgestellt über: `db2jstr` PORTNUMMER).

Ab FirstSpirit Version 4.2 muss zur Konfiguration einer IBM DB2 Datenbank 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
```

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:

- vor DB 9.5: db2jcc.jar, db2jcc_license_cu.jar
- seit DB 9.5: db2jcc4.jar, db2jcc_license_cu.jar

Die JAR-Dateien liegen auf dem DB2-Server im Verzeichnis `db2inst1/sqllib/java`. Um Inkompatibilitäten zu vermeiden, sollte immer die vom DB2-Server mitgelieferte Treiberversion verwendet werden.

Port: DB2-Connector (db2jstr 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 202) und schreibender Zugriff auf die Datenbank für dieses Projekt gesetzt (siehe Kapitel 7.4.12 Seite 308). Nachträglich kann eine Derby-Datenbank in den Server-Eigenschaften angelegt (siehe Kapitel 7.3.6 Seite 238) und in den Projekt-Eigenschaften dem Projekt hinzugefügt werden (siehe Kapitel 7.4.12 Seite 308).

Um die Derby-Datenbank von externen Prozessen aus zu benutzen, z. B. Web-Anwendung mit FirstSpirit-Modul Integration im externen Application-Server, muss zunächst der JDBC-Konnektor für Netzverbindungen aktiviert werden (siehe Kapitel 4.3.1.14 Seite 58). Als Verbindungsparameter in der Web-Anwendung 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 Web-Anwendung die Zeilen `jdbc.URL` und `jdbc.DRIVER` durch folgende ersetzen, dabei statt "fs4server" den Hostnamen des FirstSpirit-Servers eintragen und die Projekt-ID 12345 durch die tatsächliche ersetzen:



```
jdbc.URL=jdbc:derby://fs4server:1527/projects/project_12345/derby
jdbc.DRIVER=org.apache.derby.jdbc.ClientDriver
```

Der JDBC-Treiber zur Einbindung in die Web-Anwendung 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 Logdatei `firstspirit4/log/fs-database.log` abgelesen werden. Mit FirstSpirit Version 4.2R4 wurde die Derby-Version geändert auf 10.5.3.0.

Um die JDBC-Konfiguration für die einzelnen Web-Anwendungen 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 `firstspirit4/shared/lib` kopiert werden.

Falls in der externen Web-Anwendung 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://fs4server:1527/projects/project_12345/derby;
DATABASENAME=P{SCHEMA-ID}_{PROJECT-ID}
```

Ab FirstSpirit Version 4.2R4 wird empfohlen, JDBC-Treiber statt in `firstspirit4/shared/lib` und manuell in `WEB-INF/lib` als FirstSpirit-Modul einzubinden, so dass sie automatisch in alle FirstSpirit-Web-Anwendungen 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 134.

4.8.8 Vorgehen bei der Anbindung externer Datenbanken

1. Server- und Projektkonfiguration: In den **Server-Eigenschaften** (Menüeintrag "Server" / "Eigenschaften" / "Datenbanken") wird eine neue Datenbank-Anbindung konfiguriert (siehe Kapitel 7.3.6 Seite 238). 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 308) (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 (ab V4.2)

4.9.1 Ausrollprozess (Server)

Im Rahmen der Funktionalität "Integrierte Vorschau", die mit FirstSpirit Version 4.2 eingeführt wurde, 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 in FirstSpirit Version 4.2 eine aktuelle Version aller plattformabhängigen, nativen Bestandteile der Browserintegration (Client-Applikationen). Die Client-Applikationen werden beim Serverstart standardmäßig in das Verzeichnis `~FS42\data\clientapps` 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 4.3.1.8 Seite 48).

Jede Client-Applikation liegt anschließend in einem individuellen Verzeichnis unterhalb von `~FS42\data\clientapps` (Standardeinstellung), beispielsweise die Mozilla-Engine in Windows-Betriebssystemen unter:

```
~FS42\data\clientapps\JXBrowser\windows\xulrunner
```

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 `clientapps\JXBrowser\windows` oder `clientapps\JXBrowser\mac`.

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.

4.9.2 Ausrollprozess (Arbeitsplatzrechner)

Die Verzeichnisse, die beim Ausrollen der nativen Systemkomponenten auf dem Server angelegt werden (siehe Kapitel 4.9.1 Seite 165), 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\` bzw. **ab 4.2R4** `~\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\jxbrowser\xulrunner
```

bzw. für FirstSpirit-Version 4.2R4:

```
C:\Dokumente und  
Einstellungen\<USER>\.firstspirit_4.2R4\jxbrowser\xulrunner
```

Ab FirstSpirit Version 4.2R2 kann das Verzeichnis, in das die Client-Applikationen ausgerollt werden, 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 4.3.1.8 Seite 48 bzw. Kapitel 6.3.5.1 Seite 186). 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.



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

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:




```
~FS42\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\<USER>\.firstspirit\jxbrowser\crc.txt
```

bzw. ab 4.2R4 z. B.

```
C:\Dokumente und  
Einstellungen\<USER>\.firstspirit_4.2R4\jxbrowser\crc.txt
```

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.



5 Konfiguration der FirstSpirit Webanwendungen



Die gleichzeitige Anmeldung über einen Web-Browser (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 (fs4root)

Die Default-Startseite des FirstSpirit-Servers befindet sich unter:

```
<cms_basedir>\web\fs4root\index.jsp
```

Zum Aufrufen der Startseite ist ein gültiger Login notwendig (siehe Kapitel 6 Seite 180). Die Startseite kann genutzt werden, um den Anwendern einen einfachen Zugriff auf die FirstSpirit-Clients zu ermöglichen (siehe Kapitel 6.3 Seite 182).

Die FirstSpirit Startseite kann über die FirstSpirit Server- und Projektkonfiguration konfiguriert und angepasst werden kann (siehe Kapitel 7.3.8 Seite 245).

Die Schnellstart-Einträge können ebenfalls über die FirstSpirit Server- und Projektkonfiguration konfiguriert werden (siehe Kapitel 7.3.9 Seite 246).

Außerdem besteht die Möglichkeit, die Verbindungseinstellungen benutzerspezifisch zu konfigurieren (siehe Kapitel 6.3.5.1 Seite 186).

5.2 Konfiguration des WebClients

WebEdit wurde als Ergänzung zum Redaktionssystem FirstSpirit-JavaClient entwickelt. Der WebEdit-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 WebEdit keine Java-Umgebung (JRE) benötigt wird. WebEdit arbeitet aus technischer Sicht rein auf Basis von HTML und JavaScript.





Die Konfiguration von WebEdit erfolgt u.a. über die Konfigurationsdatei `fs-server.conf` im Bereich "Web Applications" (siehe Kapitel 4.3.1.7 Seite 45) und "WEBedit configuration" (siehe Kapitel 4.3.1.15 Seite 59). Vor dem produktiven Einsatz von WebEdit sollte die Konfiguration gemäß dieser Kapitel an die jeweiligen Erfordernisse angepasst werden.

Ab FirstSpirit Version 4.1 steht WebEdit nun auch als projektlokal, konfigurierbare Webanwendung zur Verfügung. Das bedeutet über die Server- und Projektkonfiguration kann für jedes Projekt eine gesonderte WebEdit-Instanz installiert, konfiguriert und aktiviert werden (siehe Kapitel 5.2.2 Seite 171).

Für den Einsatz von WebEdit in FirstSpirit-Projekten müssen bestimmte Voraussetzungen erfüllt sein (siehe Kapitel 5.2.1 Seite 170 und Kapitel 5.2.3 Seite 172).

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 ein externer Application-Server, anstelle des standardmäßig konfigurierten, internen Jetty eingesetzt werden soll (siehe Kapitel 4.6 Seite 123).

5.2.1 Projektvoraussetzungen für den Einsatz von WebEdit

Folgende Randbedingungen müssen gegeben sein, um den WebEdit-Modus in FirstSpirit Projekten einsetzen zu können. Diese Bedingungen sollten vor dem Einsatz geprüft werden:

1. Alle im Projekt benötigten Eingabekomponenten müssen von WebEdit unterstützt werden. Besonderes Augenmerk sollte dabei auf komplexe Eingabekomponenten der Datenquellen-Verwaltung, wie TABLIST oder OBJECTCHOOSEER gelegt werden. Auch kundenspezifische Komponenten müssen genau geprüft werden.
2. Browser-Kompatibilität:
 - Welcher Browser wird eingesetzt?
 - Welche Sicherheits-Einstellungen werden im Unternehmen verwendet? Läuft WebEdit in dieser Konfiguration?
3. Proxy-/Firewall-Konfiguration: Wird zwangsweise ein Proxy oder eine Firewall verwendet? Läuft WebEdit in dieser Konfiguration (speziell Refresh-Problem)?
4. Gesicherter Zugriff: kann der http/https-Port des FirstSpirit-Servers für den



WebEdit-Anwenderkreis gefahrlos (ggf. auch nach außen) freigeschaltet werden?

5. Bildschirmauflösung: mind.: 1024x768 Pixel

5.2.2 WebEdit als projektlokale Anwendung (ab V4.1)

WebEdit steht nun auch als projektlokal, konfigurierbare Webanwendung zur Verfügung. Das bedeutet über die Server- und Projektkonfiguration kann für jedes Projekt eine gesonderte WebEdit-Instanz installiert, konfiguriert und aktiviert werden.

Im Bereich "Web-Komponenten" ist dazu, neben den bisherigen Web-Bereichen "Vorschau", "QA" und "Produktion", der neue Bereich "WebEdit" vorhanden. Über die Registerkarte kann eine WebEdit-Instanz für das gewählte Projekt konfiguriert werden (siehe Kapitel 7.4.18 Seite 319).

Der WebEdit-Instanz können weitere Web-Komponenten hinzugefügt werden (z. B. FirstSpirit Integration, FirstSpirit Personalisation). 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.

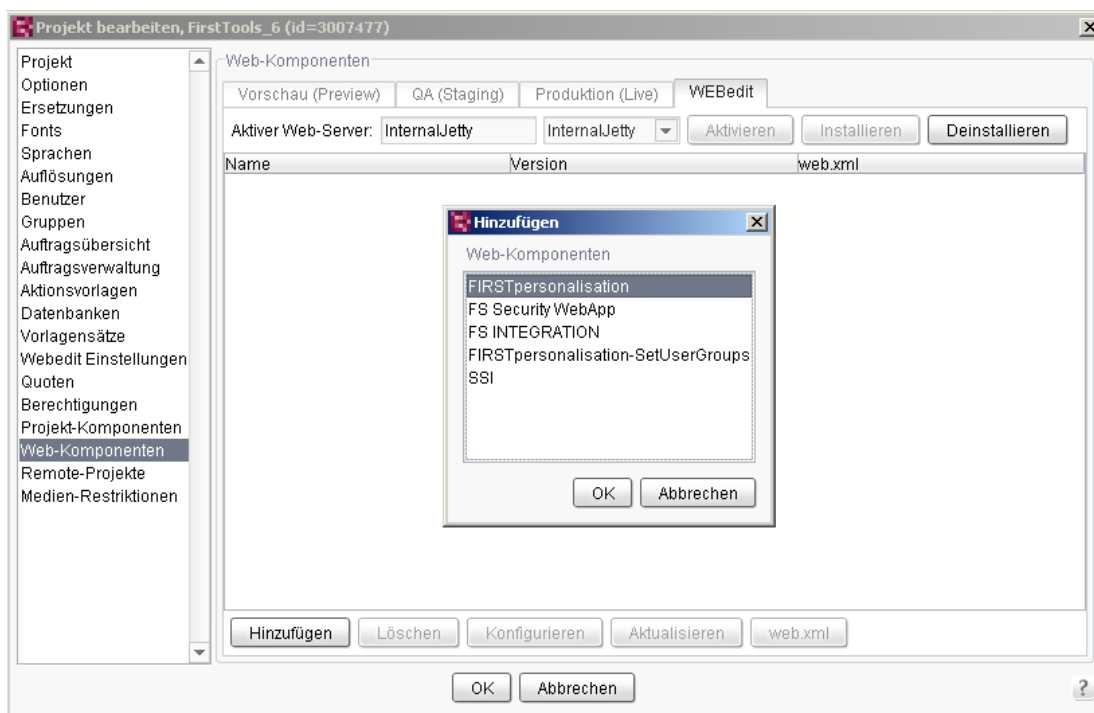


Abbildung 5-1: WebEdit als projektlokale Webanwendung

Die projektlokale Konfiguration ermöglicht es, eine "erweiterte" WebEdit-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 WebEdit mit einer projektspezifischen Personalisierungskonfiguration kombiniert werden.

Bei einer Aktualisierung auf dem Server, beispielsweise mit einer neuen WebEdit-Version, ist das Aktualisieren aller projektllokalen WebEdit-Instanzen möglich.

5.2.3 Browserkonfiguration für den Einsatz von WebEdit

Beim Arbeiten mit dem Web-Browser Firefox können Problemen beim Kopieren, Ausschneiden und Einfügen von Texten in die Eingabekomponente DOM-Editor auftreten. (Beispielsweise können Inhalte nicht mehr gespeichert oder mehrfach eingefügt werden.) Dieses Verhalten ist keine Fehlfunktion von WebEdit, 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.



5.2.4 Aktivierung von Skripten in WebEdit

Skripte können in WebEdit, sowohl in Arbeitsabläufen eingebunden, als auch einzeln ausgeführt werden. Um die Skriptfunktionalitäten einsetzen zu können, sind einige Einstellungen erforderlich:

Zunächst müssen im WebEdit-Verzeichnis (FirstSpirit-Server/www/WEBedit/WEB-INF) in der Datei "fs-webapp.xml.full" die Zeilen:

```
<init-param>
<param-name>host</param-name>
<param-value>localhost</param-value>
</init-param>
<init-param>
<param-name>port</param-name>
<param-value>1088</param-value>
</init-param>
```

an die Server-Konfiguration angepasst werden. In den meisten Fällen reicht es schon aus, den Port aus der Konfigurationsdatei "fs-webapp.xml" zu übernehmen (siehe Kapitel 8.6.1.6 Seite 442). Anschließend sollten die manuellen Änderungen an der "fs-webapp.xml"-Dateien Stück-für-Stück in diese Datei übertragen werden.

Hat man die Änderungen abgeschlossen, so sollte man eine Sicherheitskopie der "fs-webapp.xml" erstellen, die Datei "fs-webapp.xml.full" in "fs-webapp.xml" umbenennen und anschließend den Server neu starten.

Damit sind alle serverseitigen Änderungen abgeschlossen. Um die Änderungen an den WebEdit-Formatvorlagen in dem Projekt – in welchem man Skripte benutzen möchten – zu erhalten, muss über die Server- und Projektkonfiguration in den Projekteigenschaften bei WebEdit-Einstellungen die Checkbox "WebEdit benutzen" deaktiviert werden (siehe Kapitel 7.4.14 Seite 311).

Die Abfrage, ob die WebEdit-Vorlagen entfernt werden sollen, sollte mit "Ja" beantwortet werden. Anschließend ist die Checkbox bei "WebEdit benutzen" wieder zu aktivieren. Danach stehen auch Skriptfunktionalitäten im Projekt vollständig zur Verfügung.

Weitere Informationen zur Verwendung von Skripten in WebEdit befinden sich im FirstSpirit Handbuch für Entwickler (Teil 1 – Grundlagen).

5.2.5 Unterbinden von "Directory-Browsing"

"Directory-Browsing" ist bei der von FirstSpirit mitgelieferten Servlet-Engine "Jetty" standardmäßig aktiviert. Soll das "Directory-Browsing", beispielsweise im



Unternehmenseinsatz, unterbunden werden, muss dazu die "webdefault.xml"-Datei angepasst werden.

Die folgenden Zeilen:

```
<init-param>
  <param-name>dirAllowed</param-name>
  <param-value>>true</param-value>
</init-param>
```

sollten ersetzt werden durch:

```
<init-param>
  <param-name>dirAllowed</param-name>
  <param-value>>false</param-value>
</init-param>
```

Um eine eigene Fehlerseite festzulegen, muss außerdem folgende Zeilen, vor dem schließenden "web-app"-Tag, eingefügt werden:

```
<error-page>
  <error-code>403</error-code>
  <location>URL</location>
</error-page>
```

Hierbei ist natürlich anstelle von "URL" die URL zur Fehlerseite anzugeben.



Durch das Löschen einer Startseite in der Struktur-Verwaltung kann es in WebEdit zu "Directory Listings" kommen. Um dieses Verhalten zu vermeiden, sollte bei allen WebEdit-Projekten darauf geachtet werden, dass eine "Ersatzseite" (eventuell eine Fehlerseite) in den Projekteigenschaften definiert wird (siehe Kapitel 7.4.3 Seite 287).

5.2.6 WebEdit Baumdarstellung konfigurieren

WebEdit unterscheidet zwischen dem Referenznamen eines Objekts und dem symbolischen Namen eines Objekts. Jedes Objekt hat immer nur einen Referenznamen, kann aber sprachabhängig mehrere symbolische Namen besitzen.

Beispiel: Die Menüebenen aus der Struktur-Verwaltung werden auf einer Webseite als Navigationseinträge abgebildet. Wird in WebEdit eine neue Menüebene in der Struktur-Verwaltung angelegt, bekommt diese neue Menüebene beispielsweise den Referenznamen "company" zugewiesen. Ferner gibt es zwei Projektsprachen "deutsch" und "englisch", die jeweils die Menünamen "Unternehmen" und



"Company" für die neue Menüebene zugewiesen bekommen. Für die Referenzierung innerhalb des Projekts wird immer der Referenzname "company" eingesetzt. Dieser Referenzname erscheint auch in der Baumansicht der Struktur-Verwaltung. Die Beschriftung der Navigationseinträge wird jedoch sprachabhängig durch die Menünamen "Unternehmen" bzw. "Company" abgebildet.

Da diese Diskrepanz zwischen Menünamen in der Navigation und Referenzname in der Baumansicht beim Redakteur zu Verwirrung führen kann, kann die Darstellung der Baumansicht über die Datei "fs-server.conf" konfiguriert werden. Über den Parameter `showReferenceName` kann die Darstellung der Baumansicht konfiguriert werden (vgl. Kapitel 4.3.1.15 Seite 59):

- Wert "1": Ist der Parameter auf den Wert "1" gesetzt, ist die Darstellung der WebEdit-Baumansicht analog zum FirstSpirit-JavaClient. Das bedeutet: Alle Menüebenen der Struktur-Verwaltung und alle Objekte aus der Medien-Verwaltung werden mit ihren Referenznamen in der Baumdarstellung angezeigt.
- Wert "0": Ist der Parameter auf den Wert "0" gesetzt, werden alle Menüebenen der Struktur-Verwaltung und alle Objekte aus der Medien-Verwaltung mit ihren sprachabhängigen Menünamen bzw. Dateinamen in der Baumdarstellung angezeigt. Werden die sprachabhängigen Bezeichnungen genutzt, muss innerhalb der Projektkonfiguration außerdem die Checkbox "Anzeigename sprachabhängig" aktiviert werden (siehe Kapitel 7.4.2 Seite 280).

5.2.7 Konfiguration von Arbeitsabläufen in WebEdit

5.2.7.1 Arbeitsabläufe auf Seitenreferenzen starten

Bislang war es in der WebEdit-Symboleiste nur möglich, einen Arbeitsablauf auf einer Seite der Inhalte-Verwaltung zu starten. Vielfach ist es jedoch nötig, dass der Arbeitsablauf auf der Seitenreferenz gestartet wird. Die Konfiguration kann über den Parameter `webedit.sitestoreWorkflow` in der Datei "fs-server.conf" angepasst werden (vgl. Kapitel 4.3.1.15 Seite 59):

- Wert "1": Ist der Parameter auf den Wert "1" gesetzt, wird der Arbeitsablauf bei einem Klick auf das Icon "Arbeitsablauf starten" in der WebEdit-Symboleiste auf der Seitenreferenz in der Struktur-Verwaltung gestartet.
- Wert "0": Ist der Parameter auf den Wert "0" gesetzt, wird der Arbeitsablauf bei einem Klick auf das Icon "Arbeitsablauf starten" in der WebEdit-



Symbolleiste auf der Seite in der Inhalte-Verwaltung gestartet.

5.2.7.2 Mail-Benachrichtigung konfigurieren

Es ist möglich, in der Vorlagen-Verwaltung des JavaClients eine Benachrichtigung per E-Mail festzulegen, wenn ein Arbeitsablauf eine bestimmte Aktivität erreicht. Über den Platzhalter %WEBeditURL% kann dabei ein Verweis auf eine Vorschauseite in WebEdit übergeben werden (weiterführende Informationen zur Konfiguration von Arbeitsabläufen siehe *FirstSpirit Dokumentation für Entwickler* Kapitel "Eigenschaften einer Aktivität").

5.3 Single-Sign-On (SSO) konfigurieren

5.3.1 SSO in Verbindung mit dem JavaClient

Wird in einem Projekt WebEdit eingesetzt, kann die Vorschau aus der Redaktionsumgebung JavaClient ebenfalls über WebEdit erzeugt werden. Bislang musste sich ein Benutzer beim Anfordern jeder Vorschau aus dem JavaClient erneut in WebEdit authentifizieren. Durch die Funktionalität "Single-Sign-On" muss der Benutzer sich bei einer Vorschau aus dem JavaClient nun nicht mehr zusätzlich in WebEdit anmelden (Standardmäßig aktiviert).

Dazu muss innerhalb der Projekteinstellungen unter dem Menüeintrag "WebEdit Einstellungen", die Checkbox "WebEdit benutzen" aktiviert und für "WebEdit Vorlagensatz" der Pfad "WebEdit" angegeben werden (siehe Kapitel 7.4.14 Seite 311):

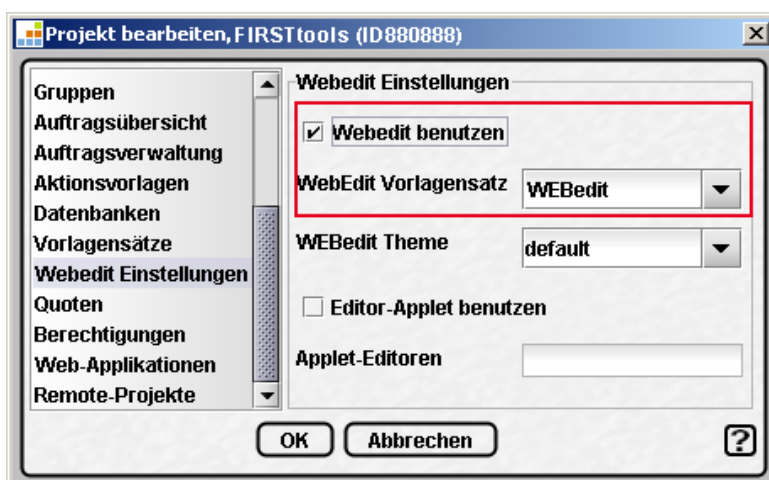


Abbildung 5-2: WebEdit für eine Vorschau aus dem JavaClient konfigurieren





Der Parameter "webedit.webSSO" in der Konfigurationsdatei `fs-server.conf` muss auf den Wert "1" gestellt werden (siehe Kapitel 4.3.1.15 Seite 59).

Neben diesen Konfigurationen sind keine weiteren Konfigurationseinstellungen notwendig. Die Konfigurationsdatei "fs-webapp.xml" und die SSO-Filter in der Datei "web/fs4webedit/WEB_INF/web.xml" sind bereits für die Verwendung von SSO konfiguriert.

Bei einer Vorschau aus dem JavaClient wird nun das SSO-Modul für die WebEdit-Authentifizierung genutzt.



Wird der JavaClient geschlossen, werden auch alle über den JavaClient erzeugten WebEdit-Sitzungen ungültig. Der Benutzer muss sich in diesem Fall erneut anmelden.

5.3.2 WebEdit mit Portal-Authentifizierung

Dieses Verfahren wird benötigt, wenn der Anmeldevorgang eines Benutzers nicht über den WebEdit-Anmelden-Dialog erfolgen soll, sondern beispielsweise über eine Portalumgebung. Der reguläre Anmelden-Dialog muss in diesem Fall deaktiviert werden. Die Konfigurationsdatei `fs-jaas.conf` muss dazu folgendermaßen ergänzt werden (websso-Modul) (vgl. Kapitel 4.3.4 Seite 76):

```
websso {
de.espirit.firstspirit.server.authentication.FSTicketLoginModule sufficient;
de.espirit.firstspirit.server.authentication.SAPLoginModule
de.espirit.firstspirit.server.authentication.FSUserLoginModule optional;
};
```

Die Konfigurationseinstellungen für den Anmeldevorgang in WebEdit müssen anschließend diesem Modul zugeordnet werden (siehe Kapitel 7.3.13 Seite 250).

5.3.3 Anmeldung mit Klartext-Passwort

Aus Sicherheitsgründen wird das Passwort bei der Anmeldung in WebEdit standardmäßig mit JavaScript verschlüsselt, damit es nicht im Klartext übermittelt wird. Diese Konfiguration kann in der Datei `fs-server.conf` über den Parameter `plainPassword` konfiguriert werden. Wird der Parameter auf den Wert "0" gesetzt, wird das Passwort nicht mehr im Klartext übertragen.

```
plainPassword=0
```

Zusätzlich zur Konfiguration der `fs-server.conf` muss die `fs-jaas.conf` angepasst werden (vgl. Kapitel 4.3.4 Seite 76):

```
websso {
de.espirit.firstspirit.server.authentication.FSTicketLoginModule sufficient;
de.espirit.firstspirit.server.authentication.FSUserLoginModule optional
hash="true";
};
```

5.3.4 Einschränkungen für den Einsatz von WebEdit

Eine Auflistung der WebEdit-Einschränkungen in Projekten findet sich in der FirstSpirit Online Dokumentation, Bereich "Vorlagenentwicklung" / "WebEdit" / "Einschränkungen".

Für WebEdit grundsätzlich nicht geplant sind:


- Veränderungen in der Vorlagen-Verwaltung (d.h. alle Arten von Vorlagen, Arbeitsabläufen und die Content-Schemata können über WebEdit grundsätzlich NICHT bearbeitet werden.)
- Unterstützung von Variablen in der Struktur-Verwaltung
- Auflösungen in der Medien-Verwaltung
- Unterstützung von Dokumentengruppen

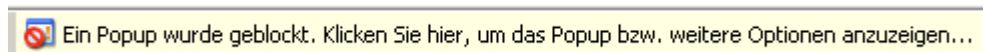
5.4 Deaktivieren von Popup-Blockern

Mit dem Einsatz von Windows XP-SP2 gehen eine Reihe von Änderungen einher, die auch den Einsatz der FirstSpirit Webanwendungen betreffen.

Voraussetzung für den Einsatz der FirstSpirit Webanwendungen ist das Abschalten des Pop-Up-Blockers für die Domain des FirstSpirit-Servers.

Folgende Maßnahmen sind zu ergreifen, damit die Anwendungen wie gewohnt benutzt werden können (Beispiel: Internet Explorer).

Wenn ein Popup geblockt wird, erscheint im Browserfenster ein gelb hinterlegter Hinweis. Angezeigt wird zusätzlich ein Popup-Block-Icon in der unteren Statusleiste des Browsers .



Ein Popup wurde geblockt. Klicken Sie hier, um das Popup bzw. weitere Optionen anzuzeigen...



Mit einem Rechtsklick auf die Meldung, kann aus dem Kontextmenü der Eintrag "Popup von dieser Site immer zulassen" ausgewählt werden.

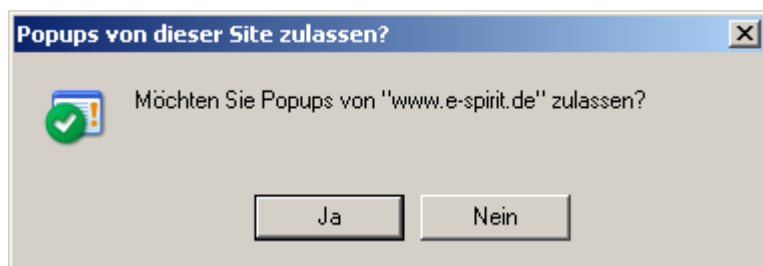



Abbildung 5-3: Popups zulassen

Wird die Abfrage mit "Ja" bestätigt, werden die Einschränkungen für das Arbeiten mit den FirstSpirit Webanwendungen aufgehoben. Der Browser zeigt in der unteren Fensterleiste das Popup-Allowed-Icon an .



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 180) oder durch Angabe von Benutzername und Passwort (Kapitel 6.2 Seite 181). Die Konfiguration des Anmeldevorgangs erfolgt über die Konfigurationsdatei `fs-jaas.conf` (Parameter siehe Kapitel 4.3.4 Seite 76).



Die gleichzeitige Anmeldung über einen Web-Browser (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 JAAS¹⁸ siehe Kapitel 4.3.1.6 Seite 44). Die entsprechende Konfiguration kann über die FirstSpirit Server- und Projektkonfiguration vorgenommen werden (siehe Kapitel 7.3.13 Seite 250).

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 und gleichzeitig der Internet Explorer verwendet wird, wird der Benutzer automatisch unter seinem Windows-Login auf dem FirstSpirit-Server angemeldet und gelangt direkt zur Startseite (Kapitel 6.3 Seite 182). Ist der Benutzer noch nicht unter seinem Windows-Login auf dem Server registriert, dann wird er als externer Benutzer neu angelegt.

¹⁸ Java Authentication and Authorization Service (Informationen: <http://java.sun.com/products/jaas/>)



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.

*FIRSTspirit
your content integration platform

Server Version: 4.0_DEVELOP.12
Server Name: pan

Deutsch

Herzlich Willkommen!

Bitte geben sie folgende Informationen ein:

Benutzer

Passwort

Anmelden

© 2007 The e-Spirit Company GmbH | Alle Rechte vorbehalten | www.FIRSTspirit.de

Abbildung 6-1: Anmeldung

Oben 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 35). 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".

Deutsch

Über diese Auswahlbox oben rechts auf der Seite 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 ist.

Passwort: In diesem Feld wird das Kennwort des Benutzers eingetragen.

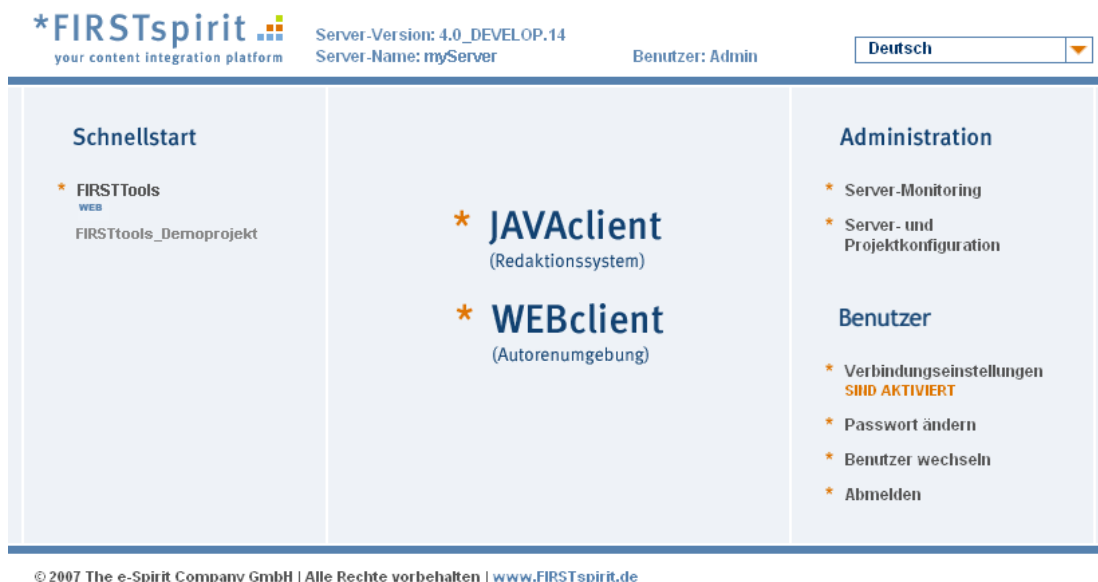
Durch einen Klick auf die Schaltfläche erfolgt die Anmeldung unter dem eingetragenen Benutzernamen.

6.3 FirstSpirit-Startseite

Nach der Anmeldung (automatisch oder manuell) öffnet sich die FirstSpirit-Startseite. Die Startseite ist unterteilt in Bereiche, die abhängig von den Rechten des jeweiligen Benutzers, ein- oder ausgeblendet werden:

- Bereich Schnellstart (siehe Kapitel 6.3.2)
- Bereich Clients (siehe Kapitel 6.3.3)
- Bereich Administration (siehe Kapitel 6.3.4)
- Bereich Benutzer (siehe Kapitel 6.3.5)

Zum Starten der FirstSpirit Server- und Projektkonfiguration und des FirstSpirit-JavaClients wird das Sun Java Runtime Environment (JRE) ab Version 1.5.0. benötigt (empfohlen wird die Version 1.6.0.), darin ist Java Webstart enthalten (Das JRE wird i.d.R. automatisch bei der Installation von JDK 1.5.0. bzw. 1.6.0. installiert).



*FIRSTspirit
your content integration platform

Server-Version: 4.0_DEVELOP.14
Server-Name: myServer

Benutzer: Admin

Deutsch

Schnellstart

- * FIRSTTools
WEB
FIRSTtools_Demoprojekt

* **JAVAclient**
(Redaktionssystem)

* **WEBclient**
(Autorenumgebung)

Administration

- * Server-Monitoring
- * Server- und Projektkonfiguration

Benutzer

- * Verbindungseinstellungen
SIND AKTIVIERT
- * Passwort ändern
- * Benutzer wechseln
- * Abmelden


© 2007 The e-Spirit Company GmbH | Alle Rechte vorbehalten | www.FIRSTspirit.de

Abbildung 6-2: FirstSpirit-Startseite

Oben auf der Seite befinden sich Informationen über den Namen und die Version des FirstSpirit-Servers. Außerdem wird dort angezeigt, welcher Benutzer gerade auf



dem Server angemeldet ist.

 Über diese Auswahlbox oben rechts auf der Seite kann eine Sprache für das weitere Arbeiten mit FirstSpirit festgelegt werden.

(Zur Konfiguration der Webanwendung FirstSpirit Startseite siehe Kapitel 5, Seite 169 ff.)

6.3.1 Überarbeitete Startseite in FirstSpirit Version 4.1

Im Zuge der Einführung des neuen "Look & Feels" wurde auch die FirstSpirit Startseite überarbeitet:



Abbildung 6-3: Neue Startseite

Neben der geänderten Darstellung werden auch einige zusätzliche Informationen auf der Startseite angezeigt:

- Name des FirstSpirit-Servers
- verwendete FirstSpirit-Version
- Name des angemeldeten Benutzers

Über die Sprachauswahl im rechten, oberen Bereich der Seite kann nun auch italienisch für die Spracheinstellung der Menübeschriftungen, Kontextmenüs und Dialoge der FirstSpirit-Anwendungen ausgewählt werden.



6.3.2 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 7.3.9 Seite 246).

6.3.3 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).
- **WebClient (Autorenumgebung):** Mit einem Klick auf diesen Eintrag wird über einen Browser die FirstSpirit Autorenumgebung gestartet. Die FirstSpirit Autorenumgebung bietet einen, gegenüber dem JavaClient, eingeschränkten Funktionsumfang zum redaktionellen Arbeiten mit FirstSpirit (siehe Kapitel 6.4).

6.3.4 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 413).
- **Server- und Projektkonfiguration:** Durch einen Klick auf diesen Eintrag öffnet sich eine Konsole für die FirstSpirit Server- und Projektkonfiguration. Eine ausführliche Beschreibung dazu befindet sich in Kapitel 7 (Seite 197).





*Dieser Bereich ist nur für Server- und Projektadministratoren sichtbar.
(Wird ein Projekt deaktiviert, werden die Einträge für die Projektadministratoren des Projekts ausgeblendet.)*

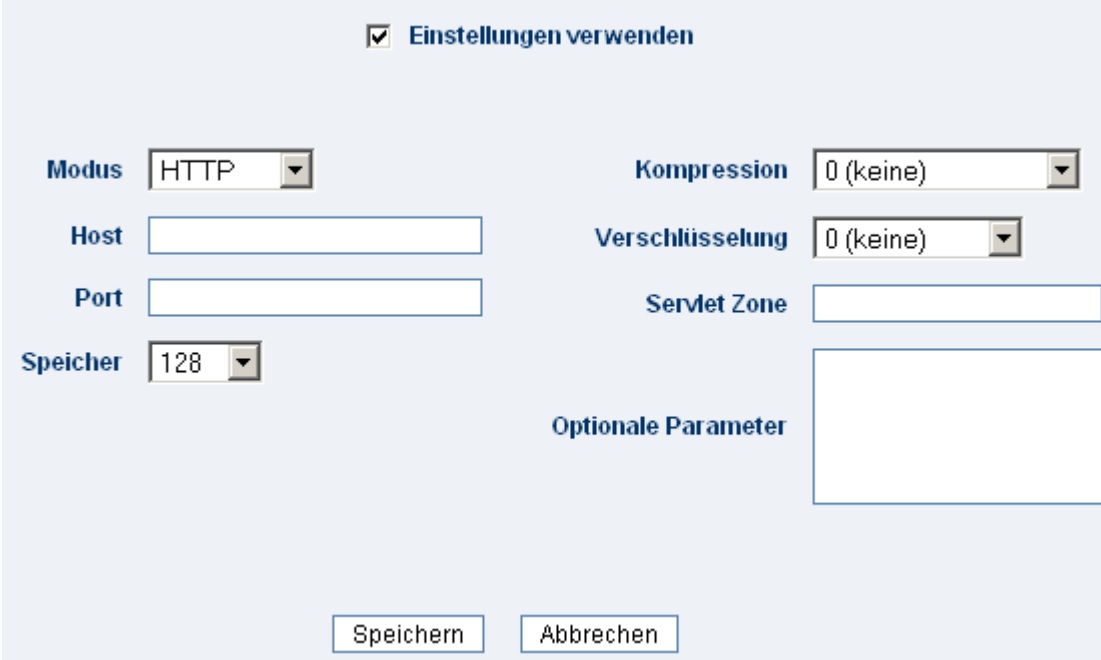
6.3.5 Benutzer

Im unteren Bereich der rechten Fensterseite befindet sich der Bereich für die Benutzereinstellungen des aktuell angemeldeten Benutzers:

- **Verbindungseinstellungen:** Hier können die Verbindungseinstellungen des aktuell angemeldeten Benutzers geändert werden (siehe Kapitel 6.3.5.1 Seite 186).
- **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 6.3.5.3 Seite 190).
- **Passwort ändern:** Hier kann das Passwort des aktuell angemeldeten Benutzers geändert werden (siehe Kapitel 6.3.5.2 Seite 189).
- **Abmelden:** Ein Klick auf diesen Eintrag beendet die aktuelle FirstSpirit Sitzung für den angemeldeten Benutzer (siehe Kapitel 6.3.5.4 Seite 190).



6.3.5.1 Verbindungseinstellungen konfigurieren



Einstellungen verwenden

Modus

Host

Port

Speicher

Kompression

Verschlüsselung

Servlet Zone

Optionale Parameter

Abbildung 6-4: Verbindungseinstellungen konfigurieren

Verbindungseinstellungen des aktuell angemeldeten Benutzers für den Start des JavaClients und der Anwendung zur Server- und Projektkonfiguration. Die hier konfigurierten Werte überschreiben die serverweit gültigen Webstart-Einstellungen für diesen Benutzer (vgl. Kapitel 7.3.8 Seite 245). 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.

Host: Servername oder IP-Adresse des FirstSpirit-Servers, mit dem sich der Client beim Web Start 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. Gültige Werte sind 128m, 256m, 512m oder 1024m.



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.

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.
- TLS¹⁹: 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 45)).

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`: Optionaler Parameter für die Kompression (bei der Kommunikation zwischen FirstSpirit-Clients und -Server).
Mögliche Werte: 0 (keine), 1 (Deflate), 2 (Deflate Speed), 3 (Deflate Best)
Beispiel: `compression=3`
- `encryption`: Optionaler Parameter für die Verschlüsselung (Kommunikation zwischen FirstSpirit-Clients und -Server).
Mögliche Werte: 0 (keine), 1 (TLS), 2 (DH ARC4)
Beispiel: `encryption=3`

¹⁹ Transport Layer Security



- `login`: Optionaler Parameter für die Angabe des Login-Moduls (plain, sso).
Beispiel: `login=plain`
- `autologin`: Optionaler 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`: Optionaler Parameter für die Angabe des Hostnamen (Vorauswahl des Servers).
- `port`: Optionaler Parameter für die Angabe der Portnummer (Ganzzahl).
- `mode`: Optionaler Parameter für die Angabe des Verbindungsmodus (`http`, `socket`)
- `httpproxy`: Optionaler 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`: Optionaler 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`: Optionaler Parameter, um die Auswertung der Java-Webstart Proxy-Konfiguration zu verhindern (vgl. `httpproxy`).
Beispiel: `nohttpproxy=1`
- `usehttps`: Optionaler 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`: Optionaler 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`
- `lookAndFeel`: Optionaler Parameter für die Umstellung des neuen Standard-Look-&-Feels (siehe Kapitel 1.3 Seite 19).
- `inlinebrowser.httpproxy` (ab FirstSpirit Version 4.2R2): Optionaler 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` (ab FirstSpirit Version 4.2R2): Optionaler Parameter zur Angabe eines Verzeichnisses im Dateisystem, in das Client-Applikationen abgelegt werden sollen (siehe Kapitel 4.3.1.8 Seite 48 und Kapitel 4.9.2 Seite 165).

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 folgende Information angezeigt:

Verbindungseinstellungen
SIND AKTIVIERT

6.3.5.2 Passwort ändern

Durch einen Klick auf diesen Eintrag kann ein Benutzer sein Passwort zur Anmeldung auf den FirstSpirit-Server ändern.



Abbildung 6-5: 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. Nicht für externe Benutzer, die durch eine automatische SSO-Anmeldung angelegt wurden.

6.3.5.3 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 181), die nun um die Möglichkeit der Automatischen Anmeldung erweitert ist.

*FIRSTspirit your content integration platform Server Version: 4.0_DEVELOP.8 Server Name: pan Deutsch

Verwenden Sie die Schaltfläche "Automatisches Anmelden", um sich automatisch ohne die Eingabe von Benutzername und Passwort anzumelden.

Automatisches Anmelden

Bitte geben sie folgende Informationen ein:

Benutzer

Passwort

Anmelden

© 2005-2006 The e-Spirit Company GmbH | Alle Rechte vorbehalten | www.FIRSTspirit.de

Abbildung 6-6: 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.

Automatisches Anmelden Durch einen Klick auf diese Schaltfläche erfolgt die Anmeldung unter dem Windows-Login.

6.3.5.4 Abmelden

Durch einen Klick auf diesen Eintrag kann sich der angemeldete Benutzer von dem Server abmelden. Es öffnet sich dann wieder die Anmeldeseite (siehe Kapitel 6.2 ab



Seite 181).

6.4 Starten der Anwendungen

6.4.1 JavaClient

Für den Start des JavaClients wird ein Web-Browser mit "Java Web Start"²⁰ benötigt. Über Java Web Start werden Softwareaktualisierungen der FirstSpirit Produkte automatisch beim Start an die Client-Systeme weitergegeben²¹. 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 182).

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.

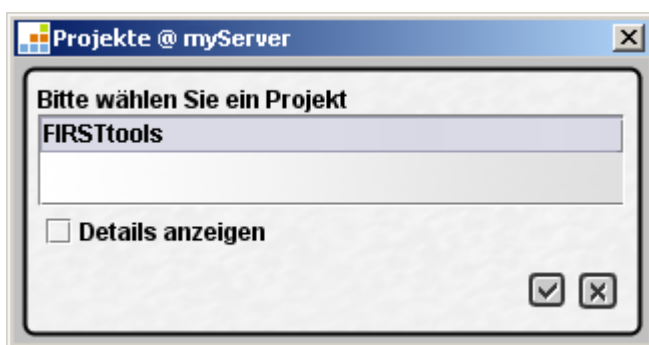


Abbildung 6-7: Projekt auswählen

Mit einem Doppelklick auf den Eintrag oder mit einem Klick auf das Symbol, wird das ausgewählte Projekt geladen.

Sie können nun damit beginnen, sich mit dem FirstSpirit Client vertraut zu machen.

²⁰ Weitere Informationen: <http://java.sun.com/products/javawebstart/>

²¹ Prinzip: http://de.wikipedia.org/wiki/Java_Web_Start



6.4.2 WebClient

Der WebClient benötigt keine Java-Umgebung und kann direkt über den Web-Browser aufgerufen werden. Zum Starten des Clients genügt ein Klick auf den entsprechenden Eintrag auf der FirstSpirit-Startseite (siehe Kapitel 6.3 Seite 182).



Der Verweis wird nur angezeigt, wenn der verwendete Browser unterstützt wird (z. B. Internet Explorer ab Version 6; weitere Informationen siehe "Technisches Datenblatt").

Beim Starten des WebClients öffnet sich nach dem Verbindungsaufbau ein Projektauswahldialog mit einer Liste der zur Verfügung stehenden Projekte für den angemeldeten Benutzer. Initial ist nur das Beispielprojekt vorhanden.

6.4.3 Server-Monitoring

Der WebClient benötigt keine Java-Umgebung und kann direkt über den Web-Browser 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 182).



Der Verweis wird nur angezeigt, wenn der verwendete Browser unterstützt wird (z. B. Internet Explorer ab Version 6; weitere Informationen siehe "Technisches Datenblatt").

6.4.4 Server- und Projektkonfiguration

Für den Start des JavaClients wird ein Web-Browser mit "Java Web Start"²² benötigt. Über Java Web Start werden Softwareaktualisierungen der FirstSpirit Produkte automatisch beim Start an die Client-Systeme weitergegeben²³. Hierfür sind unter anderem die benötigten Berechtigungskonfigurationen (z. B. Dateierstellungsrechte) auf System- oder Benutzer-Ebene vorzunehmen.

²² Weitere Informationen: <http://java.sun.com/products/javawebstart/>

²³ Prinzip: http://de.wikipedia.org/wiki/Java_Web_Start



Zum Starten des Clients genügt ein Klick auf den entsprechenden Eintrag im Bereich Administration auf der FirstSpirit-Startseite (siehe Kapitel 6.3 Seite 182).

6.5 FirstSpirit-Client als JAVA-Anwendung starten



Diese Funktionalität ist erst ab FirstSpirit-Version 4.1 freigegeben. Daher werden Screenshots im neuen Look & Feel "LightGray" dargestellt. Im Look & Feel "Classic" kann die Darstellung geringfügig abweichen.

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 6.3.5.1 ab Seite 186, konfiguriert werden.

6.5.1 Socket-Modus



Abbildung 6-8: 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 eingeloggten 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.

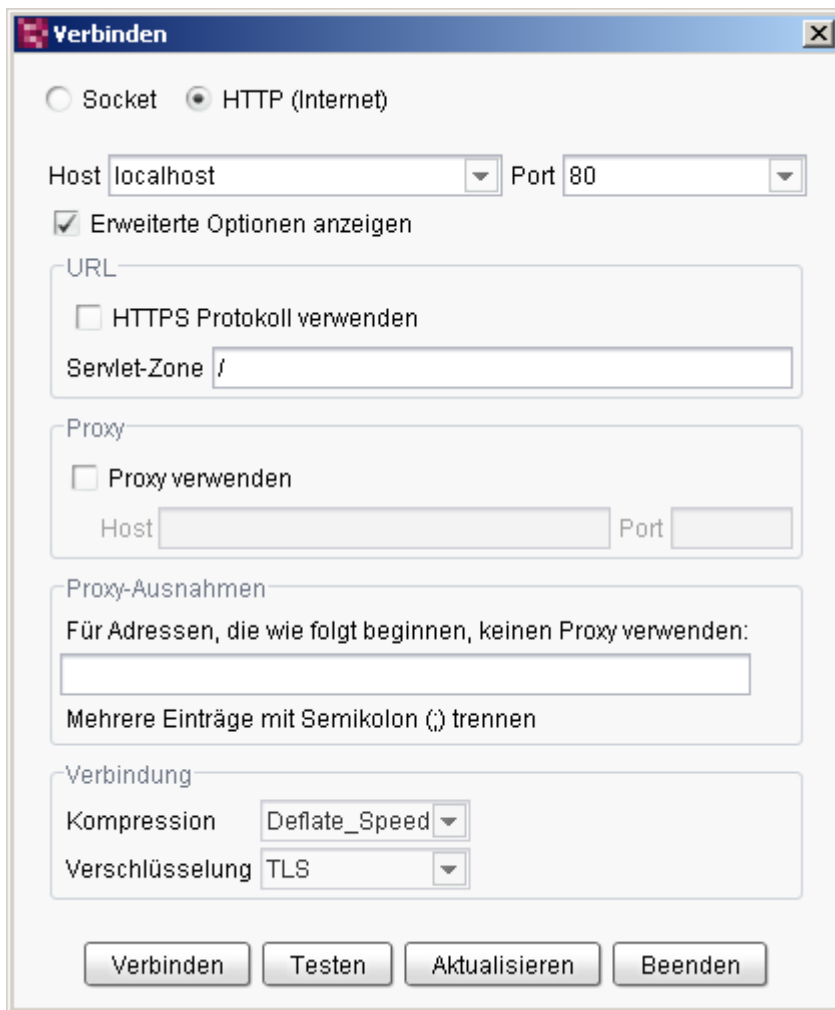
Verschlüsselung: Verschlüsselung für die Kommunikation zwischen FirstSpirit-Clients und -Server für den aktuell eingeloggten Benutzer:

- **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 6.3.5.1 Seite 186.



6.5.2 HTTP (Internet)-Modus

**Abbildung 6-9: 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 6.3.5.1 Seite 186.



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 beispielsweise 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.

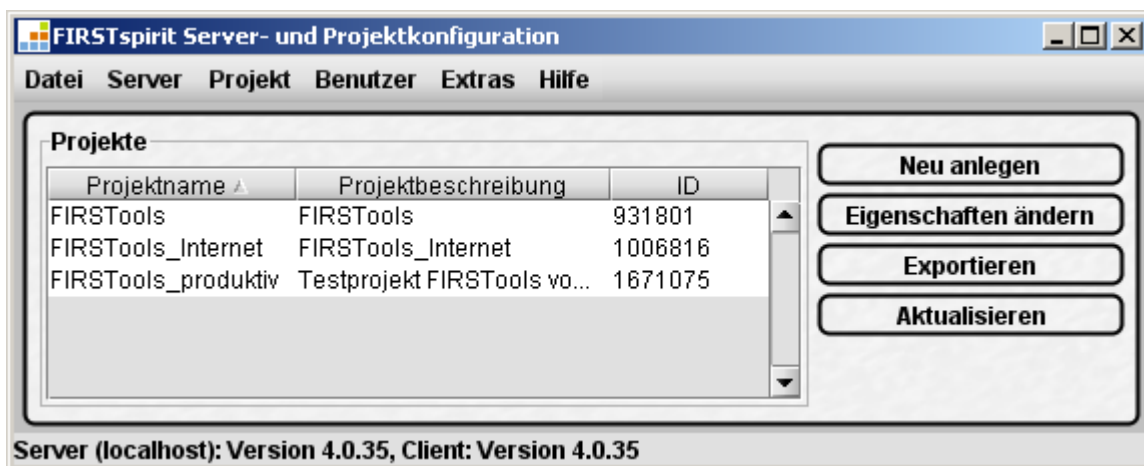


Abbildung 7-1: Server- und Projektkonfiguration

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 / exportieren / löschen
- Benutzer anlegen
- die Einstellungen aller Projekte verändern
- Projektadministratoren definieren
- Editor- und Funktions-Komponenten installieren und deinstallieren
- spezielle Operationen des Serverbetriebs ausführen

7.2 Elemente der Menüleiste

Im Folgenden werden die einzelnen Einträge in der Menüleiste der Server- und Projektkonfiguration beschrieben.

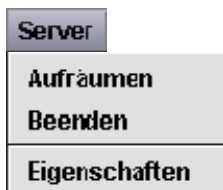
7.2.1 Datei



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:



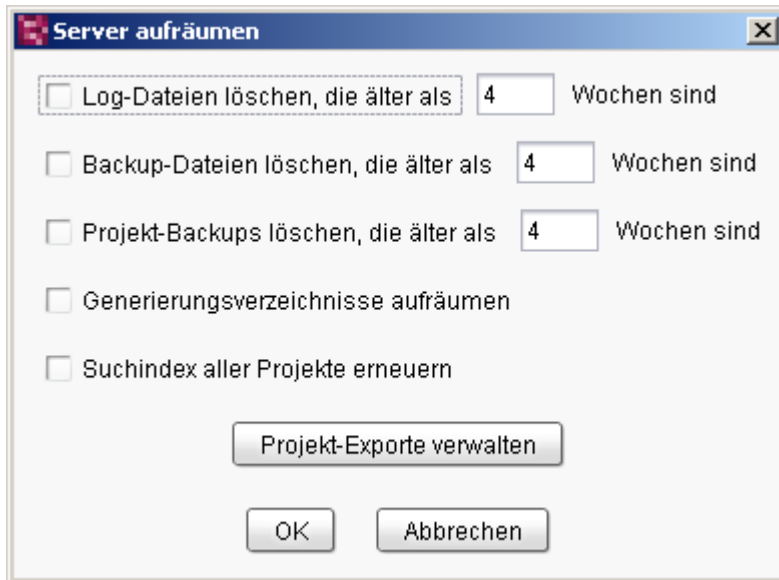


Abbildung 7-2: Server aufräumen

Log-Dateien löschen: Entfernt alle Dateien aus dem Log-Verzeichnis des Servers, die älter als die angegebene Anzahl von Wochen sind.

Backup-Dateien löschen: Beseitigt die Backup-Dateien für Seiten und Medien, die älter als die angegebene Anzahl von Wochen sind. Das Löschen von Backup-Dateien bezieht sich auf alle Projektverzeichnisse des Servers.

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 7.5.9.5 Seite 384). (Dabei ist es möglich, das Verzeichnis auf eine andere Festplatte auszulagern (siehe Kapitel 4.3.1.8 Seite 48).)



Im Export-Verzeichnis für Projekte werden über die Aktivierung der Funktion "Projekt-Backups löschen" keine Projekte 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/fs4staging" des FirstSpirit-Servers.
- Aufräumen der Verzeichnisse "project_projectID_partial_deployment" im Unterverzeichnis "web/fs4staging" des FirstSpirit-Servers.
- Aufräumen der Verzeichnisse, in dem die Previews des Projekts gespeichert wurden (Verzeichnis "project_projectID" im Verzeichnis "preview_cache" des Unterverzeichnisses "web/fs4staging"). Das Vorschau-Verzeichnis sowie die zugehörigen Pfade sind über die Konfigurationsdatei fs-server.conf konfigurierbar (siehe Kapitel 4.3.1.9 Seite 49).

Suchindex aller Projekte erneuern (ab V4.1): Ab FirstSpirit Version 4.1 kann eine Suche (Volltextsuche über den Lucene-Suchindex) über die Auswahldialoge des JavaClients erfolgen. 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 Suchindexes kann abhängig von der Anzahl und Größe der Projekte sehr lange dauern. (Eine Indizierung für einzelne Projekte kann über die BeanShell-Konsole innerhalb des JavaClient ausgeführt werden.)

Unbenutzte Projekt-Verzeichnisse löschen (ab V4.2): 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 ab FirstSpirit Version 4.2 mit dieser Option endgültig gelöscht werden.

Projekt-Exporte verwalten (ab V4.1): Mit einem Klick auf den Button öffnet sich der Dialog "Projekt-Exporte verwalten":



| Projekt | Datum | Größe | Version | Typ | Dateiname |
|-----------------------|-------------------|----------|---------|-----|---|
| fimmo31 | 06.07.07 12:47:07 | 21,47 MB | 3.x | ZIP | 20070706_124707_fimmo31.zip |
| firstimmobilia | 12.07.07 09:36:49 | 0,1 MB | 4.x | TGZ | 20070712_093649_firstimmobilia.tar.gz |
| firstimmobilia demo | 26.06.07 12:21:02 | 21,47 MB | 3.x | ZIP | 20070626_122102_firstimmobilia_demo... |
| firstimmobilia demo | 26.06.07 12:17:36 | 21,47 MB | 3.x | ZIP | 20070626_121736_firstimmobilia_demo... |
| firstimmobilia web... | 12.07.07 09:44:54 | 8,11 MB | 3.x | ZIP | 20070712_094454_firstimmobilia_webe... |
| firstimmobilia2 | 13.07.07 11:19:14 | 0,1 MB | 4.x | TGZ | 20070713_111914_firstimmobilia2.tar.gz |
| firstimmobilia3 | 06.07.07 12:40:24 | 0,1 MB | 4.x | TGZ | 20070706_124024_firstimmobilia3.tar.gz |
| firstimmobilia3 | 06.07.07 12:39:47 | 0,1 MB | 4.x | TGZ | 20070706_123947_firstimmobilia3.tar.gz |
| firstimmobilia5 | 06.07.07 12:43:04 | 0,1 MB | 4.x | TGZ | 20070706_124304_firstimmobilia5.tar.gz |
| firstools | 09.07.07 14:08:25 | 54,79 MB | 4.x | TGZ | 20070709_140825_firstools.tar.gz |
| firstools 070921 | 04.10.07 10:29:49 | 21,79 MB | 4.x | TGZ | 20071004_102949_firstools_070921.tar.gz |
| firstools 070921 | 21.09.07 09:33:09 | 21,86 MB | 4.x | TGZ | 20070921_093309_firstools_070921.tar.gz |
| firstools 071015 | 18.12.07 17:11:04 | 23,61 MB | 4.x | TGZ | 20071218_171104_firstools_071015.tar.gz |
| firstools 071015 | 15.10.07 16:10:12 | 21,86 MB | 4.x | TGZ | 20071015_161012_firstools_071015.tar.gz |
| firstools 071120 | 20.11.07 16:01:35 | 21,86 MB | 4.x | TGZ | 20071120_160135_firstools_071120.tar.gz |
| firstools 071120 2 | 20.11.07 16:45:12 | 21,86 MB | 4.x | TGZ | 20071120_164512_firstools_071120_2.t... |
| firstools 2007060... | 24.07.07 16:27:48 | 54,79 MB | 4.x | TGZ | 20070724_162748_firstools_20070605_... |
| firstools 2007101... | 10.10.07 12:54:28 | 21,86 MB | 4.x | TGZ | 20071010_125428_firstools_20071010_... |
| firstools 2007101... | 10.10.07 12:58:45 | 21,86 MB | 4.x | TGZ | 20071010_125845_firstools_20071010_... |
| firstools internet | 11.09.07 11:22:34 | 50,70 MB | 4.x | TGZ | 20070911_112234_firstools_internet.tar.gz |

Abbildung 7-3: Projekt-Exporte verwalten

Über den Dialog werden tabellarisch alle vorhandenen Projektexporte aufgelistet. Diese können in der Übersicht markiert und bei Bedarf über den Button "Projekt-Export löschen" gelöscht werden:

7.2.2.2 Beenden

Mit dieser Funktion wird der Server herunter gefahren.



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 226.



7.2.3 Projekt



- Projekt neu anlegen (siehe Kapitel 7.2.3.1)
- Projekt importieren (siehe Kapitel 7.2.3.2)
- Projekt exportieren (siehe Kapitel 7.2.3.3)
- Projekt deaktivieren (siehe Kapitel 7.2.3.4)
- Projekt reaktivieren (siehe Kapitel 7.2.3.5)
- Projekt löschen (siehe Kapitel 7.2.3.6)
- Projekt Eigenschaften (siehe Kapitel 7.2.3.7)
- Projekt Archiv (siehe Kapitel 7.2.3.8)

7.2.3.1 Neu anlegen

Mit dieser Funktion lässt sich ein neues Projekt auf dem Server anlegen.

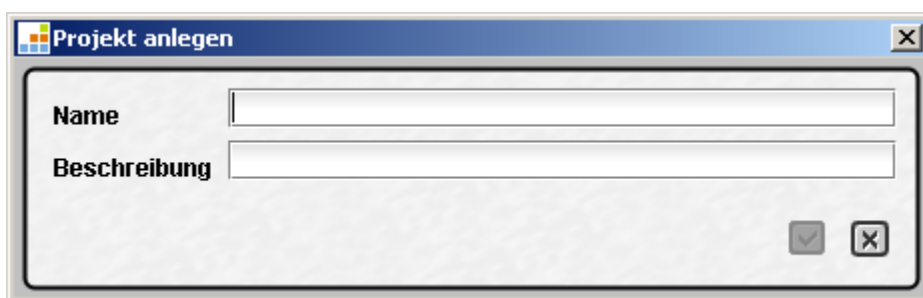


Abbildung 7-4: Projekt – Neu anlegen

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: Durch Aktivieren dieser Checkbox werden die mit FirstSpirit ausgelieferten Standard Formatvorlagen 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, die Arbeitsabläufe:

1. "Aufgabe": Arbeitsablauf zur allgemeinen Bearbeitung von Aufgaben im Projekt.
2. "Freigabe Anfordern": Arbeitsablauf zur Freigabe von Objekten, die im Projekt neu angelegt oder verändert wurden.

WebEdit Formatvorlagen: Durch Aktivieren dieser Checkbox werden die mit FirstSpirit ausgelieferten WebEdit-Formatvorlagen in den Knoten Formatvorlagen (innerhalb der Vorlagen-Verwaltung des Projekts) importiert. Für die Verwendung von WebEdit in Projekten werden diese Formatvorlagen nicht zwingend benötigt. In den meisten Fällen ist die Verwendung von Formatvorlagen jedoch sinnvoll, beispielsweise, wenn ein Projekt mit Frames arbeitet oder wenn bestimmte WebEdit-



Elemente, z. B. "Quick-Edit", verwendet werden sollen.

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 308).



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 aufgenommen und kann über die Schaltfläche "Eigenschaften ändern" (siehe Kapitel 7.3.16 ab Seite 264) ausführlicher bearbeitet werden.

Der Button  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 Button 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):

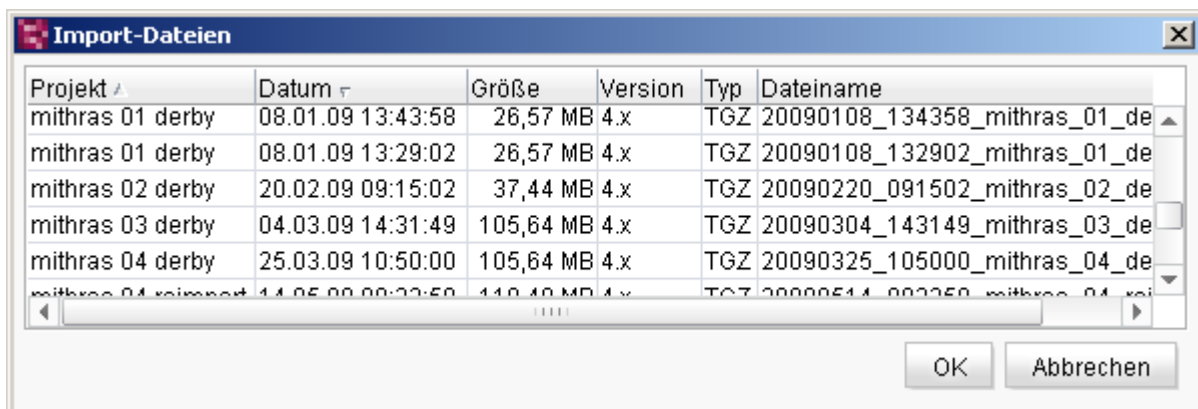


Abbildung 7-7: Projekt Importieren – Liste der Import-Dateien

Projektname: In diesem Feld muss ein eindeutiger Name für das zu importierende Projektes eingetragen werden. Dieser Name wird auf der Projektwurzel des JavaClients angezeigt.

Projektbeschreibung: In diesem Feld kann eine Projektbeschreibung für das zu



importierende Projektes 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:

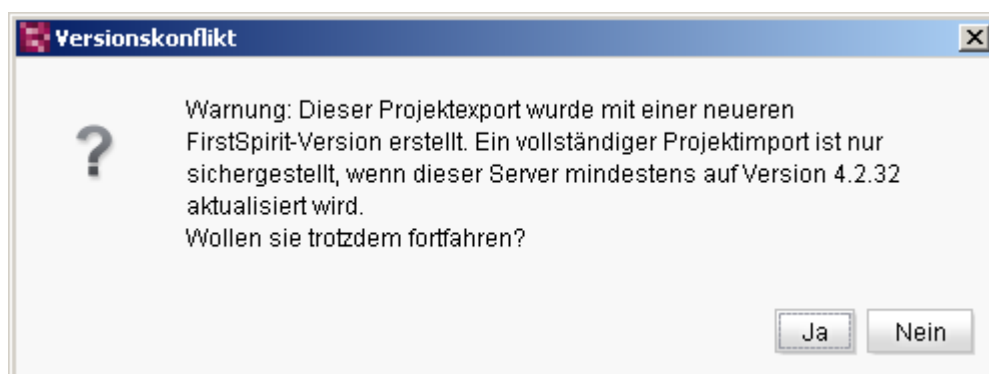


Abbildung 7-8: Versionskonflikt beim Import

Mit "Nein" wird der Import-Vorgang abgebrochen, mit "Ja" wird der Import weiter ausgeführt.





Der Import von Projekten, die mit einem FirstSpirit-Server einer höheren Version erstellt wurden (z. B. Export von einem FirstSpirit-Server der Version 4.2, Import auf einem FirstSpirit-Server der Version 4.1) ist keine zugesicherte Produkteigenschaft! Technisch ist ein solcher Export/Import zwar möglich, kann aber gerade z. B. bei einem Export von 4.2 nach 4.1 aufgrund der zahlreichen neu implementierten Eingabekomponenten, Methoden sowie inkompatiblen Änderungen im Software-Verhalten zu nicht vorhersehbaren Problemen führen.

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 bestehen nun die folgenden Möglichkeiten:

- bis FirstSpirit Version 4.2: allen Schemata eines Projektes einen neuen oder einen bereits auf dem Server vorhandenen Datenbank-Layer zuzuweisen
- seit FirstSpirit Version 4.2: jedem einzelnen Schema eines Projektes einen neuen oder einen bereits auf dem Server vorhandenen Datenbank-Layer zuzuweisen (siehe Abbildung 7-9):

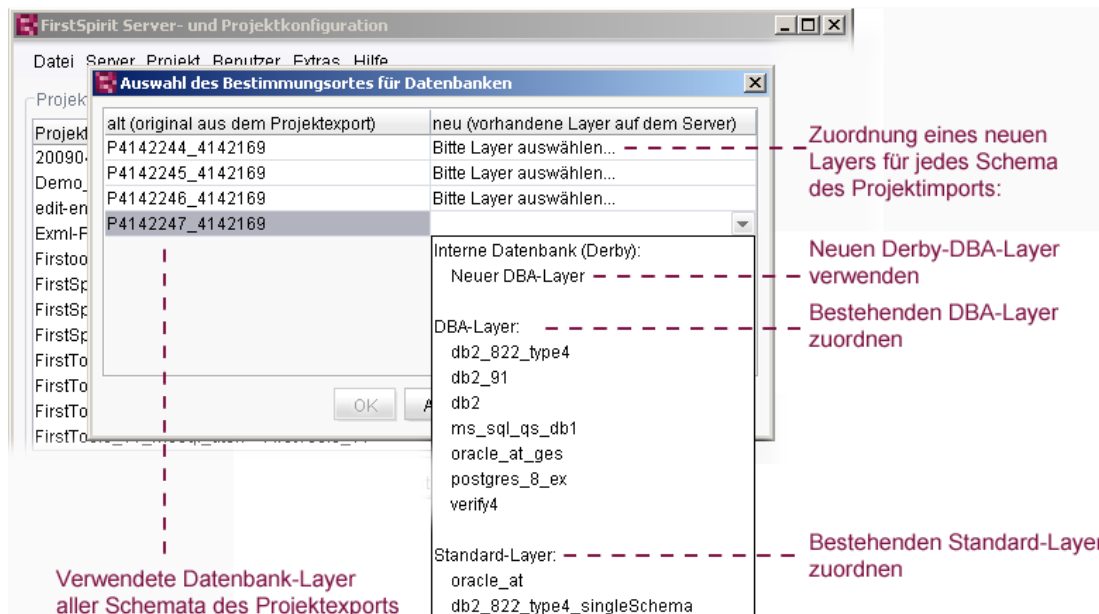


Abbildung 7-9: Auswahl des Bestimmungsortes für Datenbanken (Version 4.2)

Bis FirstSpirit Version 4.2 konnte beim Import eines Projektes nur ein Datenbank-Layer für alle Schemata eines Projektes zugewiesen werden.

Seit FirstSpirit Version 4.2 wird nun jedes Schema der Exportdatei im Dialog "Auswahl des Bestimmungsortes für Datenbanken" angezeigt (linker Bereich). Die neue Zuordnung eines Layers für das importierte Projekt kann über den rechten



Bereich ausgewählt werden. Dabei kann für jedes einzelne Schema entweder:

- ein neuer Derby-DBA-Layer
- ein Standard-Layer oder
- ein DBA-Layer ausgewählt werden.



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 150).



*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"*

Der Importvorgang kann, abhängig von der Projektgröße, einige Minuten dauern. Der Gesamtfortschritt sowie die einzelnen Aktionen werden im Importdialog dargestellt.



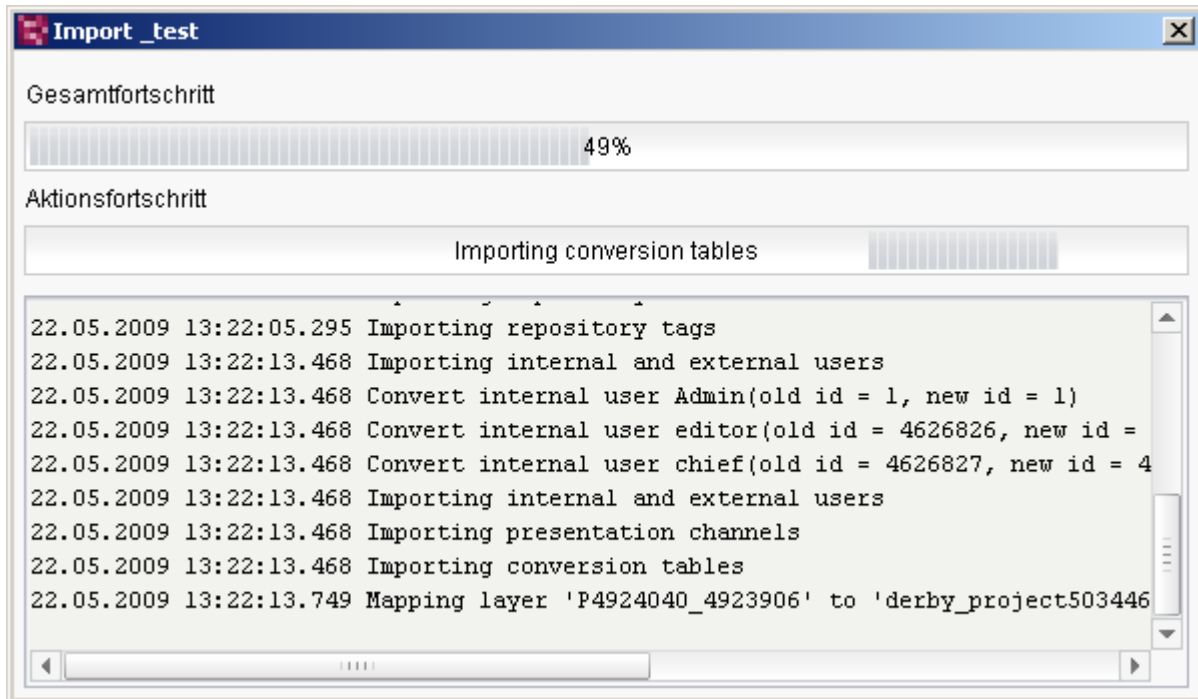


Abbildung 7-10: Projekt Importieren – Fortschrittsanzeige



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, dann wird die Exportdatei auch im Dateisystem des lokalen Rechners gespeichert.

Nach Bestätigung der Einstellungen durch **OK** öffnet sich ein Fenster, in dem noch der Zielort für die Exportdatei ausgewählt werden kann.

Serverdatei nach dem Download löschen: Diese Option lässt sich nur aktivieren, wenn ein lokaler Download der Exportdatei erfolgt. Ist diese Option aktiviert, dann 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 bei besonders großen Projekten erforderlich, um diese



noch auf externe Datenträger kopieren zu können.)

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 9.13 Seite 481).

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.

7.2.3.4 Deaktivieren

Will man ein Projekt vorübergehend aus der Projektliste entfernen (ohne es gleich vollständig zu löschen) kann man es 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 Button 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.

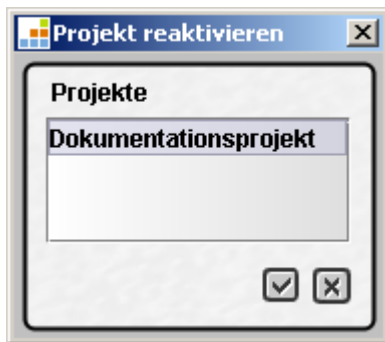


Abbildung 7-13: Projekt – Reaktivieren

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.

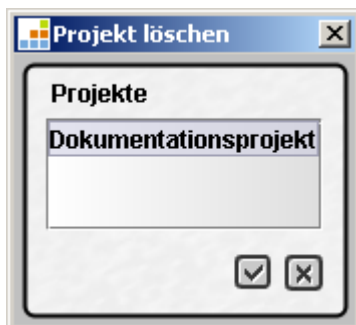


Abbildung 7-14: Projekt – Löschen

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 7-15: Projekt – Eigenschaften

Nach Auswahl des Projektes gelangt man in den Bereich Projekteinstellungen. Näheres dazu siehe in den Projekteinstellungen (Kapitel 7.3.16 ab Seite 264).

7.2.3.8 Archiv (ab V4.1)



Diese Funktionalität ist erst ab FirstSpirit-Version 4.1 freigegeben. Daher werden Screenshots im neuen Look & Feel "LightGray" dargestellt. Im Look & Feel "Classic" kann die Darstellung geringfügig abweichen.



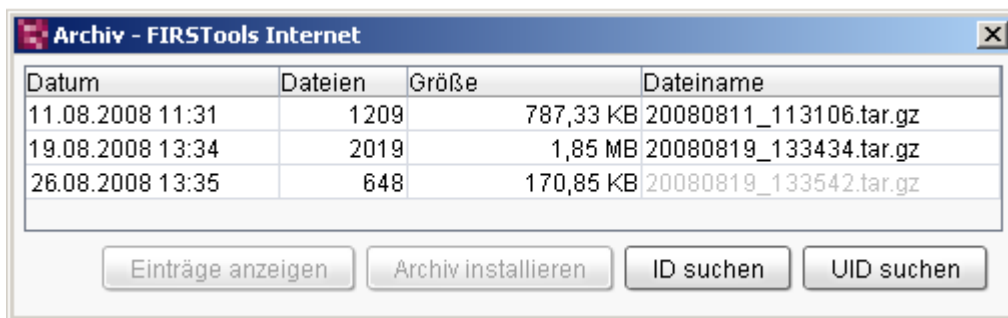
Zur Verwendung dieser Funktion im Vergleich zur Verwendung des Moduls "FirstSpirit EnterpriseBackup" siehe auch FirstSpirit Release Notes 4.2, Kap. "Langzeit-Archivierung und Backup in FirstSpirit".





Abbildung 7-16: Projekt – Archiv

Über die Funktion "Archiv" können Daten aus Archivdateien, die während eines Archivierungsauftrags angelegt wurden (Kapitel 7.5.9.1 Seite 363), angezeigt werden:



| Datum | Dateien | Größe | Dateiname |
|------------------|---------|-----------|------------------------|
| 11.08.2008 11:31 | 1209 | 787,33 KB | 20080811_113106.tar.gz |
| 19.08.2008 13:34 | 2019 | 1,85 MB | 20080819_133434.tar.gz |
| 26.08.2008 13:35 | 648 | 170,85 KB | 20080819_133542.tar.gz |

Below the table are four buttons: 'Einträge anzeigen', 'Archiv installieren', 'ID suchen', and 'UID suchen'.

Abbildung 7-17: Archivliste

Nach der Auswahl eines Projekts aus Abbildung 7-16 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 215)
- Datenbankeinträge (siehe Kapitel 7.2.3.8.2 Seite 216)
- Revisionen (siehe Kapitel 7.2.3.8.3 Seite 217)



7.2.3.8.1 Archivierte Bauelemente (ab V4.1)

| Pfad | ID / UID | Tag | Revision | Archiv | Typ |
|-------------------------|------------------------|-----------------|---------------------|------------------------|-------------|
| / | 305 305 | PROJECTSETTINGS | 15179 ... 16041 (4) | 20080811_113106.tar.gz | OLD_VERSION |
| /layout | 364 stylesheets | MEDIANODE | 14418 | 20080811_113106.tar.gz | OLD_VERSION |
| / | 365 visuals | MEDIANODE | 15901 | 20080811_113106.tar.gz | OLD_VERSION |
| / | 589 div | PAGENODE | 15406, 15845, 15846 | 20080811_113106.tar.gz | OLD_VERSION |
| / | 595 service | PAGENODE | 15485 | 20080811_113106.tar.gz | OLD_VERSION |
| /service/news | 599 newspaper | PAGE | 14614 | 20080811_113106.tar.gz | OLD_VERSION |
| / | 601 Marketing | PAGENODE | 15408, 15699, 15700 | 20080811_113106.tar.gz | OLD_VERSION |
| /service/glossary_1 | 603 glosstest | PAGE | 12569 ... 16054 (4) | 20080811_113106.tar.gz | OLD_VERSION |
| / | 644 644 | GLOBALCONTENT | 11625, 16040 | 20080811_113106.tar.gz | OLD_VERSION |
| /service/news/news... | 646 center | BODY | 14614 | 20080811_113106.tar.gz | OLD_VERSION |
| /service/glossary_1/... | 649 center | BODY | 12569 ... 16054 (4) | 20080811_113106.tar.gz | OLD_VERSION |
| /service/news/news... | 662 left | BODY | 14614 | 20080811_113106.tar.gz | OLD_VERSION |
| /service/glossary_1/... | 663 left | BODY | 12569 ... 16054 (4) | 20080811_113106.tar.gz | OLD_VERSION |
| /service/news/news... | 693 contentquellen... | SECTION | 14614 | 20080811_113106.tar.gz | OLD_VERSION |
| /service/glossary_1/... | 694 Glossar | SECTION | 12569 ... 16054 (4) | 20080811_113106.tar.gz | OLD_VERSION |
| /service/glossary_1/... | 761 newsbox | SECTION | 12569 ... 16054 (4) | 20080811_113106.tar.gz | OLD_VERSION |
| /service/news/news... | 762 Button rechts z... | SECTION | 14614 | 20080811_113106.tar.gz | OLD_VERSION |
| /service/news/news... | 763 Sonderangebot | SECTION | 14614 | 20080811_113106.tar.gz | OLD_VERSION |
| /service/news/news... | 769 Text/Bild | SECTION | 14614 | 20080811_113106.tar.gz | OLD_VERSION |
| / | 783 homepage2col... | PAGE | 15256, 16087, 16088 | 20080811_113106.tar.gz | OLD_VERSION |
| / | 1066 default_1 | MEDIUM | 6435 | 20080811_113106.tar.gz | DELETED |
| / | 1540 homepage_1 | PAGE | 9063 | 20080811_113106.tar.gz | DELETED |

102 Einträge

Abbildung 7-18: 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 (ab V4.1)

| Schema-UID | Schema-ID | Tabelle | FS_ID | gültig von | gültig bis | freigegeben bis | Archiv |
|----------------|-----------|------------------|-------|-------------------|-------------------|-------------------|------------------------|
| usermanagement | 33271 | users | 0 | 12.07.07 23:37:59 | 31.08.07 23:02:09 | - | 20080811_110123.tar.gz |
| usermanagement | 33271 | users | 1 | 12.07.07 23:38:29 | 31.08.07 23:02:09 | - | 20080811_110123.tar.gz |
| usermanagement | 33271 | users | 1 | 31.08.07 23:02:09 | 31.01.08 13:09:24 | 31.01.08 13:09:47 | 20080811_110123.tar.gz |
| usermanagement | 33271 | users | 0 | 31.08.07 23:02:09 | 31.01.08 13:09:31 | 31.01.08 13:09:56 | 20080811_110123.tar.gz |
| usermanagement | 33271 | users | 1 | 31.01.08 13:09:24 | 31.01.08 13:09:47 | - | 20080811_110123.tar.gz |
| usermanagement | 33271 | users | 0 | 31.01.08 13:09:31 | 31.01.08 13:09:56 | - | 20080811_110123.tar.gz |
| usermanagement | 33271 | groups | 2 | 12.07.07 23:39:11 | 31.08.07 23:02:04 | - | 20080811_110123.tar.gz |
| usermanagement | 33271 | groups | 3 | 12.07.07 23:39:42 | 31.08.07 23:02:04 | - | 20080811_110123.tar.gz |
| usermanagement | 33271 | groups | 4 | 12.07.07 23:40:19 | 31.08.07 23:02:04 | - | 20080811_110123.tar.gz |
| products | 33270 | product_cate... | 2 | 05.06.07 10:55:50 | 05.06.07 10:56:42 | - | 20080811_110123.tar.gz |
| products | 33270 | product_cate... | 2 | 05.06.07 10:56:42 | 31.08.07 11:36:03 | - | 20080811_110123.tar.gz |
| products | 33270 | product_cate... | 3 | 05.06.07 10:57:09 | 31.08.07 11:36:03 | - | 20080811_110123.tar.gz |
| products | 33270 | product_cate... | 4 | 05.06.07 10:57:36 | 31.08.07 11:36:03 | - | 20080811_110123.tar.gz |
| products | 33270 | product_appli... | 0 | 05.06.07 10:55:15 | 31.08.07 11:36:07 | - | 20080811_110123.tar.gz |
| products | 33270 | product_appli... | 1 | 05.06.07 10:55:27 | 31.08.07 11:36:07 | - | 20080811_110123.tar.gz |

283 Einträge

Abbildung 7-19: Archivierte Informationen zu Datenbankeinträgen

Wurden durch den Archivierungsauftrag Datenbankeinträge archiviert (siehe Kapitel 7.5.9.1 Seite 363), 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 (ab V4.1)

| Revision | Datum | Bearbeiter | Kommentar | Status | Archiv |
|----------|-------------------|----------------------|------------------------------|-----------|------------------------|
| 15630 | 02.07.08 10:00:18 | Admin, id=1 | edit permissions | COMMITTED | 20080811_113106.tar.gz |
| 15633 | 02.07.08 10:03:02 | Admin, id=1 | create node\$childUid=te... | COMMITTED | 20080811_113106.tar.gz |
| 16225 | 15.07.08 14:38:01 | Admin, id=1 | | COMMITTED | 20080811_113106.tar.gz |
| 16261 | 15.07.08 14:51:43 | Admin, id=1 | save | COMMITTED | 20080811_113106.tar.gz |
| 15846 | 04.07.08 16:36:51 | Admin, id=1 | cancel task creation | COMMITTED | 20080811_113106.tar.gz |
| 15645 | 02.07.08 10:21:36 | test (test), id=6815 | move child\$childUid=test... | COMMITTED | 20080811_113106.tar.gz |
| 15653 | 02.07.08 10:25:59 | test (test), id=6815 | move child\$childUid=test... | COMMITTED | 20080811_113106.tar.gz |
| 15699 | 02.07.08 11:13:10 | Admin, id=1 | unlock | COMMITTED | 20080811_113106.tar.gz |
| 16054 | 14.07.08 15:25:29 | Admin, id=1 | unlock | COMMITTED | 20080811_113106.tar.gz |
| 16088 | 14.07.08 16:49:06 | Admin, id=1 | | COMMITTED | 20080811_113106.tar.gz |
| 15692 | 02.07.08 11:12:15 | Admin, id=1 | create node\$childUid=tex... | COMMITTED | 20080811_113106.tar.gz |
| 16062 | 14.07.08 15:50:39 | Admin, id=1 | delete\$childUid=text_bild | COMMITTED | 20080811_113106.tar.gz |
| 16058 | 14.07.08 15:36:11 | Admin, id=1 | save | COMMITTED | 20080811_113106.tar.gz |
| 16232 | 15.07.08 14:38:02 | Admin, id=1 | @ update workflow states | COMMITTED | 20080811_113106.tar.gz |
| 16236 | 15.07.08 14:38:10 | Admin, id=1 | Start workflow Freigabe A... | COMMITTED | 20080811_113106.tar.gz |
| 16223 | 15.07.08 14:37:27 | Admin, id=1 | Start workflow Freigabe A... | COMMITTED | 20080811_113106.tar.gz |
| 16231 | 15.07.08 14:38:02 | Admin, id=1 | server release | COMMITTED | 20080811_113106.tar.gz |

Abbildung 7-20: Archivierte Revisionen

Wurde im Archivierungsauftrag die Option "Systemdaten" im Bereich Objekte aktiviert (siehe Kapitel 7.5.9.1 Seite 363), 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

: 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 7-17)

1. Installation der Archivdatei vom 19.08.2008, 13:34
2. Installation der Archivdatei vom 11.08.2008, 11:31



Gelöschte Archivdateien können nicht installiert werden.

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.



Abbildung 7-21: Suche nach UID

Bei der Suche werden alle für das Projekt vorhandenen Archive durchsucht. Die Ergebnisse werden in folgendem Fenster aufgelistet:



| Pfad | ID | UID | Tag | Revision | Archiv | Typ |
|------|----|------------|-------------------|---------------------|------------------------------|-------------|
| / | | 15087 test | PAGE | 15698 ... 16236 (4) | 20080811_113106.tar... | OLD_VERSION |
| / | | 15227 test | CONTENTFOLDER | | 15724 20080811_113106.tar... | DELETED |
| / | | 26793 test | SITENODE | 15907 ... 15922 (7) | 20080811_113106.tar... | DELETED |
| /644 | | 26937 test | GLOBALCONTENTNODE | | 15969 20080811_113106.tar... | DELETED |
| / | | 26938 test | GCA_PAGE | 15970 ... 15980 (4) | 20080811_113106.tar... | DELETED |
| / | | 31064 test | MEDIANODE | 16166, 16167 | 20080811_113106.tar... | OLD_VERSION |
| / | | 31069 test | SITENODE | 16275 ... 16281 (5) | 20080811_113106.tar... | OLD_VERSION |

7 Einträge

Abbildung 7-22: Ergebnisliste für UID-Suche nach "test"

7.2.4 Benutzer

Benutzer

- Neu anlegen
- Bearbeiten
- Löschen

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.

Benutzer anlegen

Login:

Kennwort:

Name:

Kürzel:

E-Mail:

Telefon:

Aktiv:

Abbildung 7-23: Benutzer – Neu anlegen

Login: Loginname des neuen Benutzers (Pflichtfeld).

Kennwort: Kennwort des neuen Benutzers (Pflichtfeld).





Für den Ausnahmefall, dass **nicht** LDAP, Kerberos oder NTLM zur Authentifizierung verwendet werden, werden in Kennworten bei der Anmeldung nur ASCII-Zeichen geprüft.

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 7-24).

Aktiv (ab V4.1): 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 220).


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.)

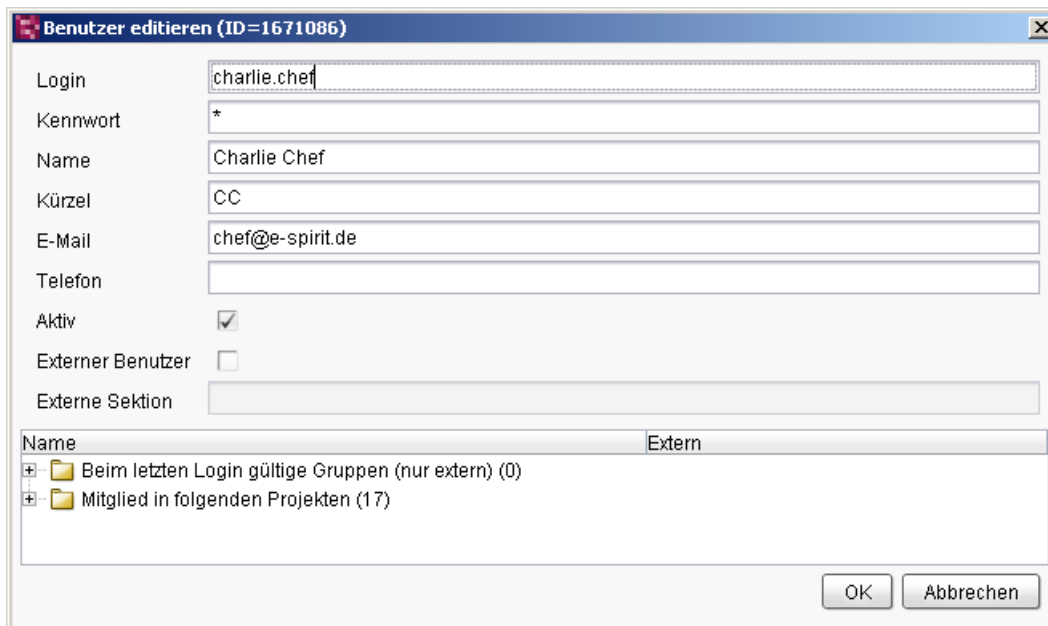
| ID | Name | Login | Kürzel | E-Mail | Aktiv |
|--------|--------------|--------------|--------|--------|-------|
| 898805 | User A | UserA | | | ✓ |
| 898806 | User B | UserB | | | ✓ |
| 33269 | User-1238... | User-1238... | U | | ✓ |
| 33155 | User-13FB... | User-13FB... | U | | ✓ |
| 86383 | User-16A6... | User-16A6... | U | | ✓ |
| 31098 | User-18E5... | User-18E5... | U | | ✓ |
| 708113 | User-1A94... | User-1A94... | U | | ✓ |
| 28095 | User-1BA3... | User-1BA3... | U | | ✓ |
| 30097 | User-1C7B... | User-1C7B... | U | | ✓ |

Abbildung 7-24: 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.



Benutzer editieren (ID=1671086)

Login: charlie.chef

Kennwort: *

Name: Charlie Chef

Kürzel: CC

E-Mail: chef@e-spirit.de

Telefon:

Aktiv:

Externer Benutzer:

Externe Sektion:

Name: Extern

- Beim letzten Login gültige Gruppen (nur extern) (0)
- Mitglied in folgenden Projekten (17)

OK Abbrechen

Abbildung 7-25: 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 (ab V4.1): Ab FirstSpirit Version 4.1 können Benutzer deaktiviert werden, ohne die Benutzer endgültig aus FirstSpirit zu entfernen. Die Benutzer bleiben in diesem Fall 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 298).

Ist die Checkbox *deaktiviert*, wurde der Benutzer manuell angelegt (siehe Abbildung 7-23).

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 Dialog 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 304. Diese Information ist auch über das FirstSpirit™ Server Monitoring verfügbar (siehe Kapitel 8.5 Seite 436).

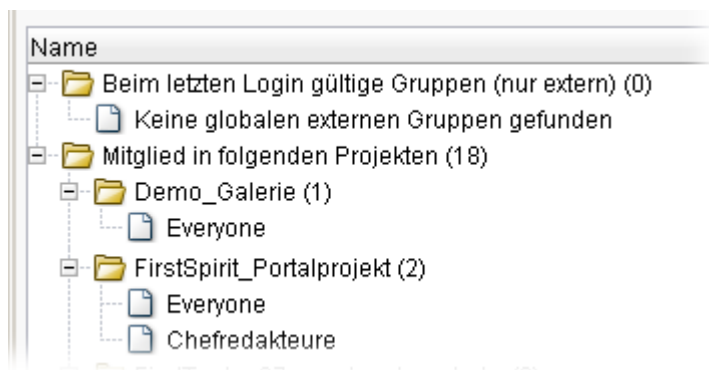


Abbildung 7-26: Gruppenzugehörigkeit eines Benutzers

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.

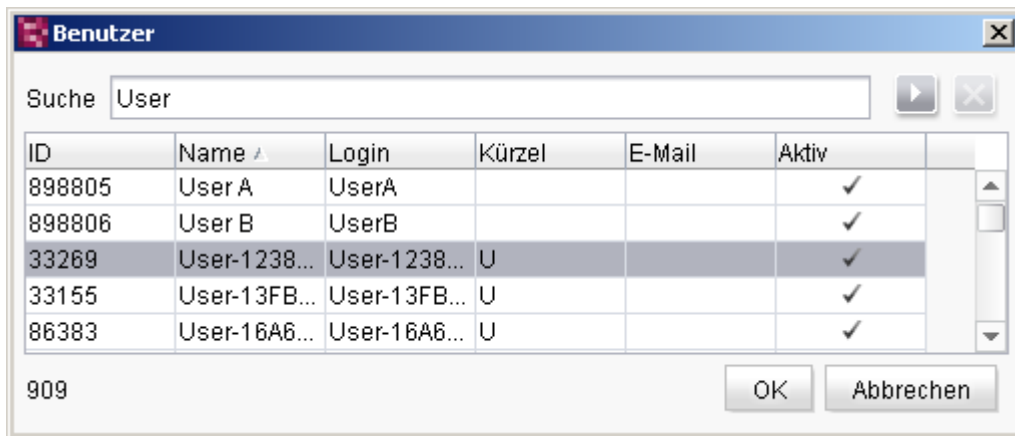



Abbildung 7-27: Benutzer löschen

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 unterscheidet man zwischen:

- 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.
- 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!



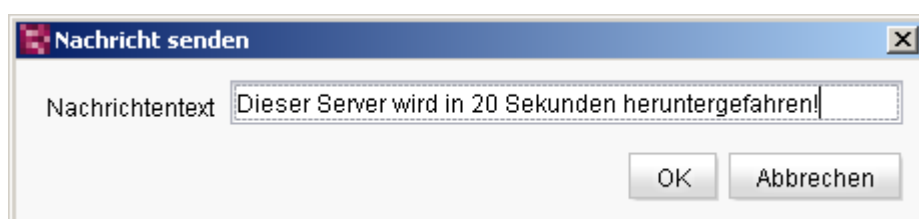
7.2.5 Extras

Extras

Nachricht senden

7.2.5.1 Nachricht senden

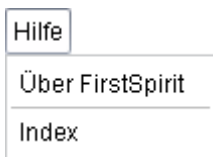
Hier kann ein Nachrichtentext editiert werden, der allen aktiven Clients am Server in Form eines PopUp-Fensters übermittelt wird.



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.

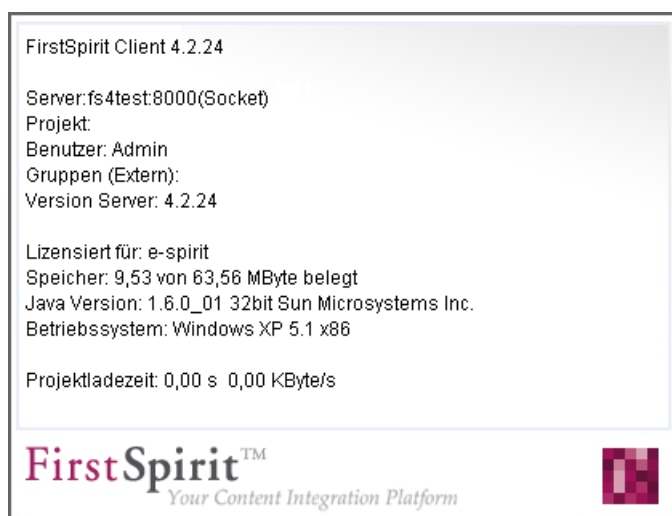


7.2.6 Hilfe



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

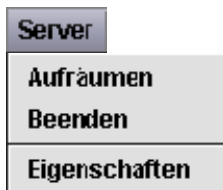


Abbildung 7-28: Menü Server – Servereigenschaften

Über den Menüeintrag Servereigenschaften, können die Eigenschaften des FirstSpirit-Servers konfiguriert werden.

7.3.1 Globale Server-Eigenschaften

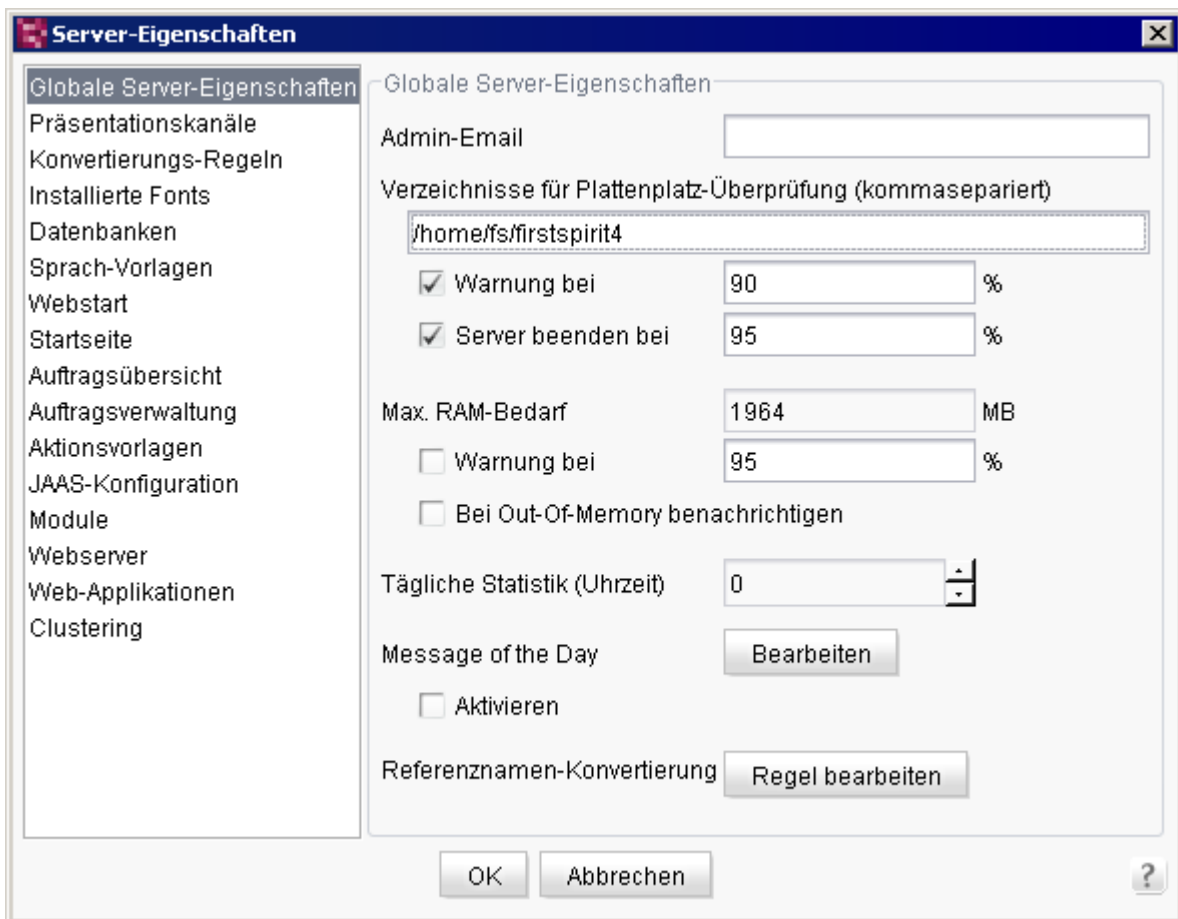


Abbildung 7-29: Server-Eigenschaften – Globale Server-Eigenschaften (ab 4.2R4)

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.

Max. Plattenplatz (bis 4.2R4): In diesem Feld kann der Wert für den maximalen Plattenplatz (in MB) für den Server eingestellt werden. 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 Speicherplatzes überschritten wurde (optional kann eine E-Mail an den Serveradministrator gesendet werden).

Verzeichnisse für Plattenplatz-Überprüfung (kommasepariert) (ab 4.2R4): Mit Version 4.2R4 wurden einige Maßnahmen implementiert, um den bereits verbrauchten bzw. noch freien Speicherplatz der Festplatte besser überwachen und so eventuelle Datenverluste umgehen zu können.

In dieses Feld werden dazu die Verzeichnisse eingetragen, 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 4.3.1.16 Seite 62). 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.



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

Warnung bei (ab 4.2R4): In diesem Feld wird die Prozentzahl angegeben, ab der eine Warnungs-E-Mail an den Serveradministrator versendet werden soll. Betreff und Inhalt der E-Mail: "FIRSTspirit server 'MYSERVER' disk space warning: /home/fs/firstspirit4, 32,01 GB free, 42,52 GB used".

Die Prozentzahl ergibt sich dabei aus dem Verhältnis des im angegebenen Verzeichnis zur Verfügung stehenden und des bereits belegten Speicherplatzes. Standardmäßig ist der Wert auf 90% gesetzt, d. h. der Serveradministrator erhält eine Warnungs-E-Mail, wenn 90% des zur Verfügung stehenden Speicherplatzes belegt sind (siehe auch `fs-server.conf`, Parameter `hdd.limit`, Kapitel 4.3.1.16



Seite 62). Sind mehrere Verzeichnisse angegeben, wird eine E-Mail versendet, wenn in **einem** der Verzeichnisse die Prozentzahl überschritten ist. Der hier gewählte Wert sollte niedriger sein, als der Wert für "Server beenden" (siehe unten). Es wird maximal alle 12 Stunden eine Warnungs-E-Mail versendet.

Über die Checkbox kann der Versand der Warnungs-E-Mails aktiviert oder deaktiviert werden (siehe auch `fs-server.conf`, Parameter `hdd.limit.active`, Kapitel 4.3.1.16 Seite 62). Standardmäßig ist der Versand bei der angegebenen Prozentzahl aktiviert.

Server beenden bei (ab 4.2R4): In diesem Feld wird die Prozentzahl angegeben, bei der der FirstSpirit-Server heruntergefahren und eine E-Mail an den Serveradministrator versendet werden soll. Betreff und Inhalt der E-Mail: "FIRSTspirit server 'MYSERVER' disk space shutdown limit reached: /home/fs/firstspirit4, 463.2 MB free, 5.54 GB used".

Die Prozentzahl ergibt sich dabei aus dem Verhältnis des im angegebenen Verzeichnis zur Verfügung stehenden und des bereits belegten Speicherplatzes. Standardmäßig ist der Wert auf 95% gesetzt, d. h. der Server wird heruntergefahren, wenn 95% des zur Verfügung stehenden Speicherplatzes belegt sind (siehe auch `fs-server.conf`, Parameter `hdd.shutdown`, Kapitel 4.3.1.16 Seite 62). Sind mehrere Verzeichnisse angegeben, wird der Server heruntergefahren, wenn in **einem** der Verzeichnisse die Prozentzahl überschritten ist. Der hier gewählte Wert sollte höher sein, als der Wert für "Warnung bei" (siehe oben).

Über die Checkbox kann das Herunterfahren des Servers aktiviert oder deaktiviert werden (siehe auch `fs-server.conf`, Parameter `hdd.shutdown.active`, Kapitel 4.3.1.16 Seite 62). Standardmäßig ist das Herunterfahren bei der angegebenen Prozentzahl 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: Ab FirstSpirit Version 4.2R4 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:



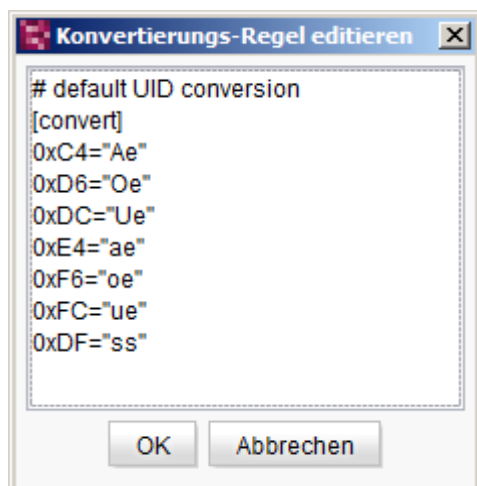


Abbildung 7-30: Referenznamen-Konvertierung

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

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

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



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





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 erst neu gestartet werden.

7.3.2 Präsentationskanäle

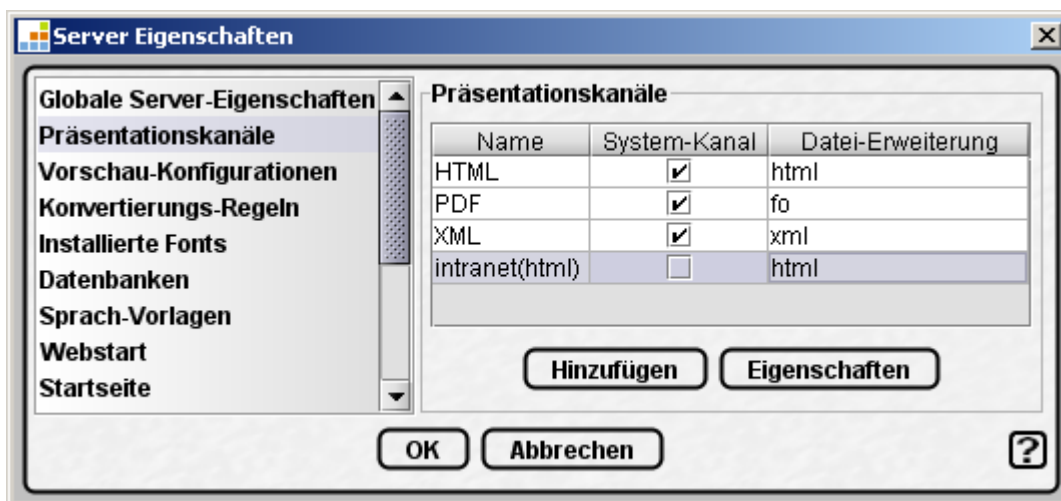


Abbildung 7-31: Server-Eigenschaften – Präsentationskanäle

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 7-32).

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 7-32).



Abbildung 7-32: 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.

Zur Unterstützung von Apache FOP siehe Kapitel 7.8 Seite 406.



7.3.3 Vorschau-Konfigurationen

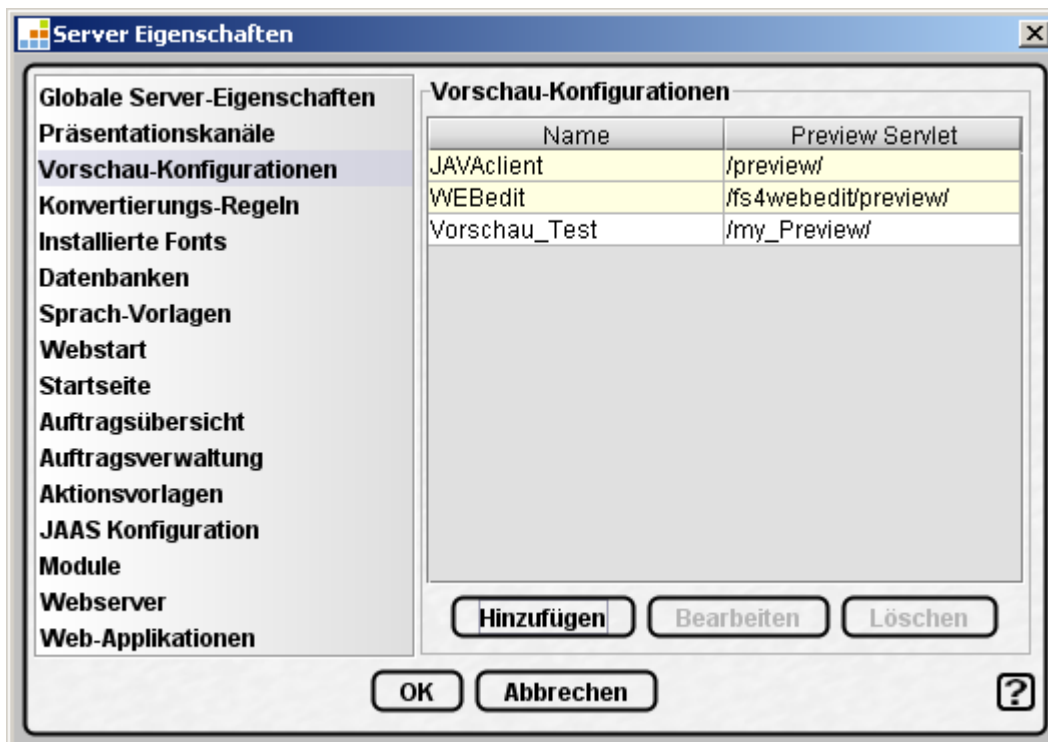


Abbildung 7-33: Server-Eigenschaften Vorschau-Einstellungen

In diesem Bereich können Einstellungen für die Vorschau konfiguriert werden. Standardmäßig sind die Standard-Vorschau-Einstellungen für den JavaClient und den WebClient konfiguriert. Die Standardeinstellungen sind gelb hinterlegt und können nicht bearbeitet oder gelöscht werden.

Name: Eindeutiger Name für die Konfiguration. Unter diesem Namen kann die Konfiguration innerhalb der Projekteigenschaften für die Vorschau eines Projekts konfiguriert werden (siehe Kapitel 7.4.13 Seite 309).

Preview Servlet: Pfadangabe zum Verzeichnis, in dem das Preview Servlet liegt.

Hinzufügen: Mit einem Klick auf den Button kann eine neue Vorschaukonfiguration angelegt werden. Dazu muss ein eindeutiger Name für die neue Konfiguration und der Pfad zum Verzeichnis angelegt werden, indem das Preview Servlet liegt.

Bearbeiten: Mit einem Klick auf den Button kann eine Vorschau-Konfiguration bearbeitet werden. Die Standardeinstellungen sind gelb hinterlegt und können nicht bearbeitet werden.

Löschen: Mit einem Klick auf den Button kann eine Vorschau-Konfiguration



bearbeitet werden. Die Standardeinstellungen sind gelb hinterlegt und können nicht gelöscht werden.

7.3.4 Konvertierungs-Regeln

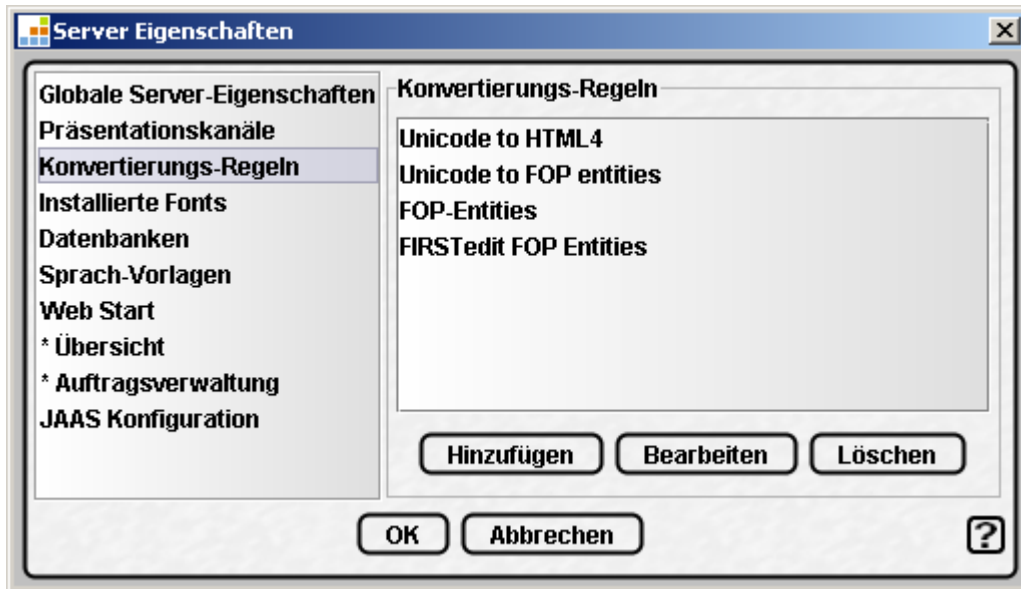


Abbildung 7-34: Server-Eigenschaften – Konvertierungs-Regeln

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. Standard-Konvertierungsregeln sind beispielsweise "Unicode to HTML4" für den HTML-Ausgabekanal und "FOP-Entities" für den PDF-Ausgabekanal. Dabei kann für jeden Ausgabekanal 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.

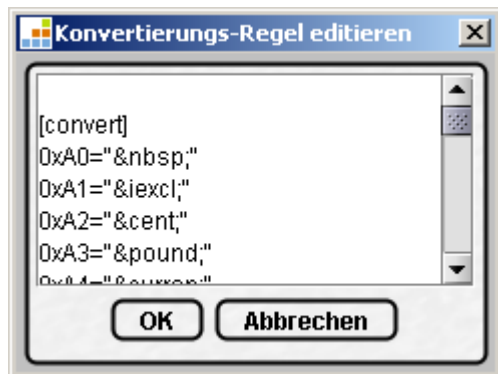


Abbildung 7-35: Server-Eigenschaften – Konvertierungs-Regel editieren

Löschen: Mit einem Klick auf den Button können bestehende Konvertierungsregeln vom Server gelöscht werden. Vor dem Löschen erfolgt eine Sicherheitsabfrage.



7.3.5 Installierte Fonts

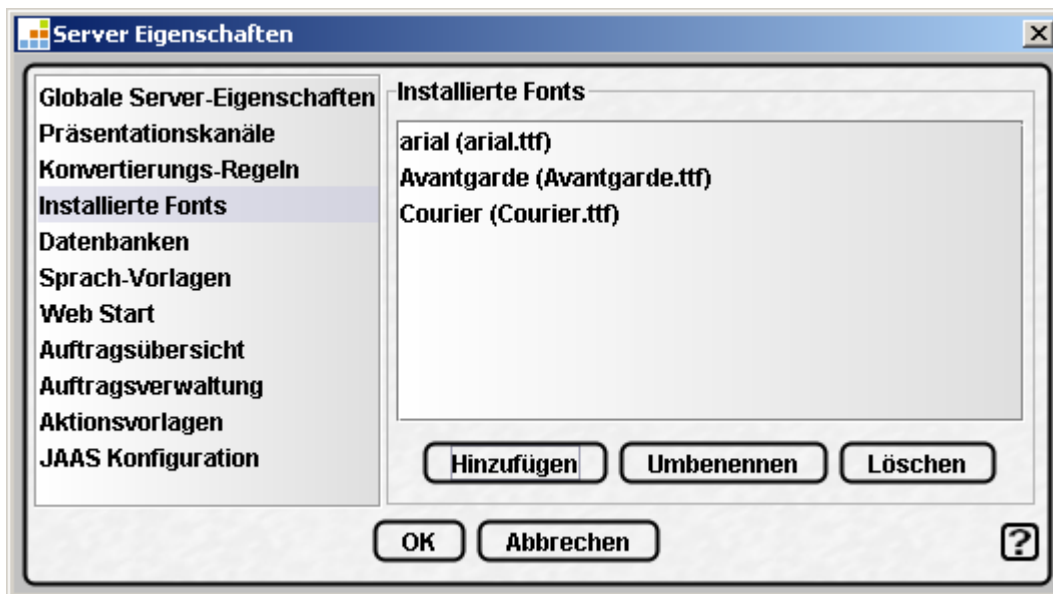


Abbildung 7-36: Server-Eigenschaften – Fonts installieren

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 7-36).

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 7-37: 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.

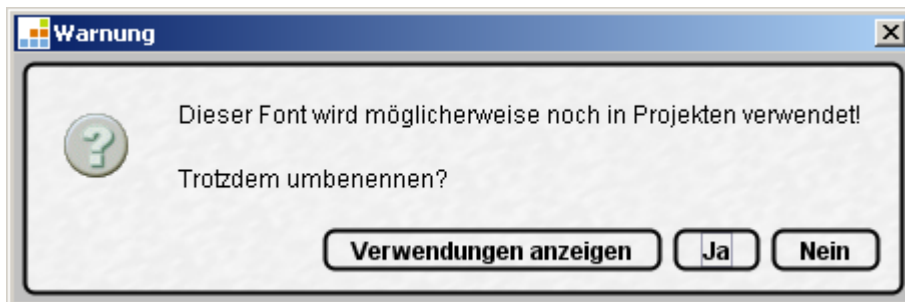


Abbildung 7-38: Warnung vor dem Löschen bzw. Umbenennen von Fonts

OK

Mit einem Klick auf den Button 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", diese Projekte ermittelt werden.

Verwendungen anzeigen

Mit einem Klick auf den Button 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.

Abbrechen

Mit einem Klick auf den Button wird der Dialog geschlossen. Der



betroffene Font wird nicht gelöscht bzw. umbenannt.

7.3.6 Datenbanken

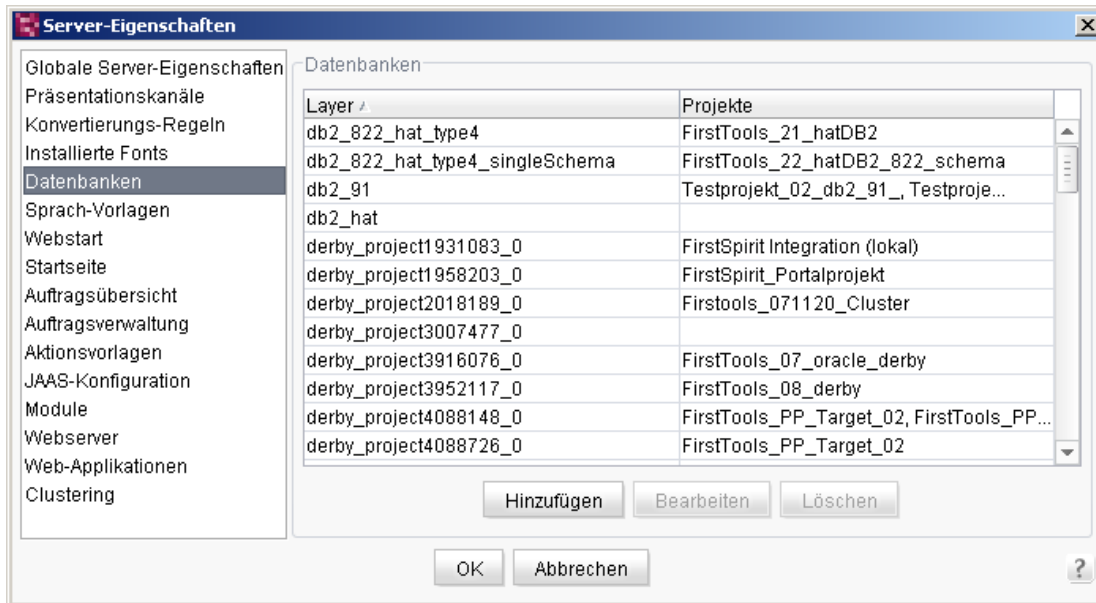


Abbildung 7-39: Server-Eigenschaften – Datenbanken

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 308) 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)").

Hinzufügen

Mit einem Klick auf den Button 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.



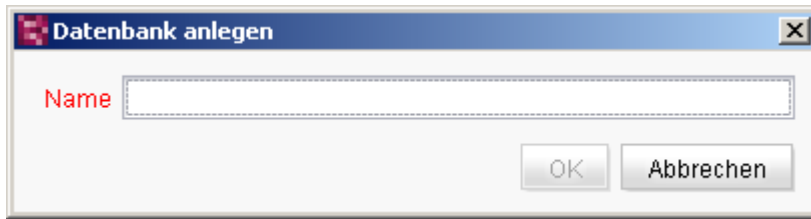


Abbildung 7-40: Anlegen eines neuen Datenbank-Layers

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 7.3.6.1 f.). (Genauere Informationen zur Datenbankanbindung siehe Kapitel 4.3.3 Seite 75 und zur Konfiguration der Datenbank siehe Kapitel 4.8.4 Seite 147.)



FirstSpirit verwendet für den Zugriff auf Datenbanken JDBC²⁴. 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 133).

Bearbeiten

Eine bestehende Konfiguration lässt sich mit einem Klick auf diesen Button bearbeiten. Es öffnet sich ein Fenster mit der bestehenden Konfiguration (siehe Abbildung 7-41). Man unterscheidet:

- Konfiguration JDBC-Parameter (siehe Kapitel 7.3.6.1 Seite 240)
- Konfiguration Verbindung (siehe Kapitel 7.3.6.2 Seite 241)

Löschen

Ein Klick auf den Button löscht 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.)

²⁴ Java Database Connectivity





Bei Datenbanken, die aktuell in einem Projekt verwendet werden, erscheint vor dem Löschen eine Sicherheitsabfrage.

7.3.6.1 JDBC Parameter konfigurieren



Abbildung 7-41: Datenbankanzbindung konfigurieren (JDBC)

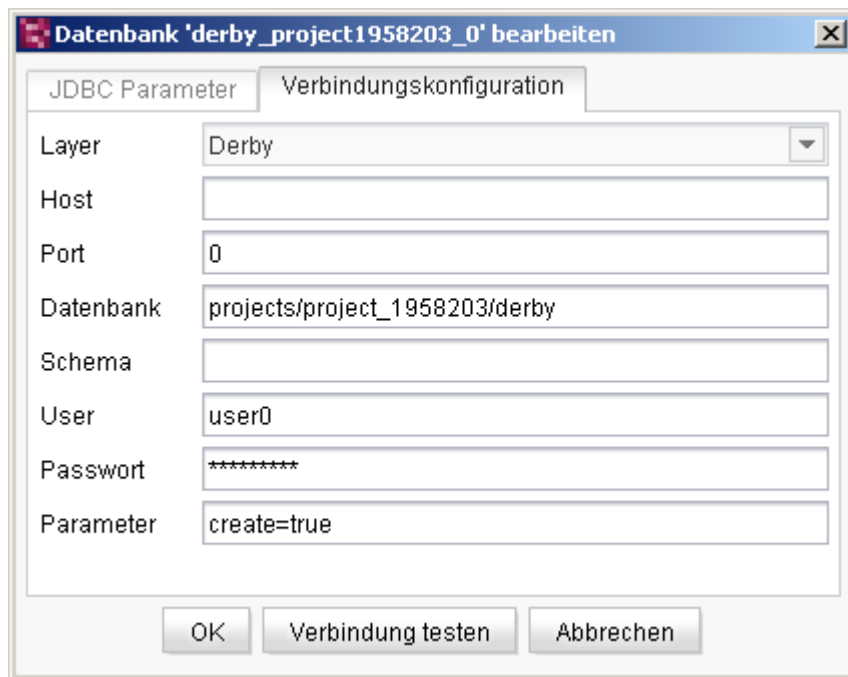
Konfiguration der JDBC-Parameter für die Datenbankanzbindung.

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 147.

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.



7.3.6.2 Verbindungskonfiguration testen

**Abbildung 7-42: 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 149).

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: Man unterscheidet zwischen:

- **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 149).

Passwort: Gültiges Passwort zum Login unter `<database>.jdbc.USER` (vgl. Parameter `jdbc.password` Kapitel 4.8.4.1 Seite 149).

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 147.

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.

7.3.7 Sprach-Vorlagen

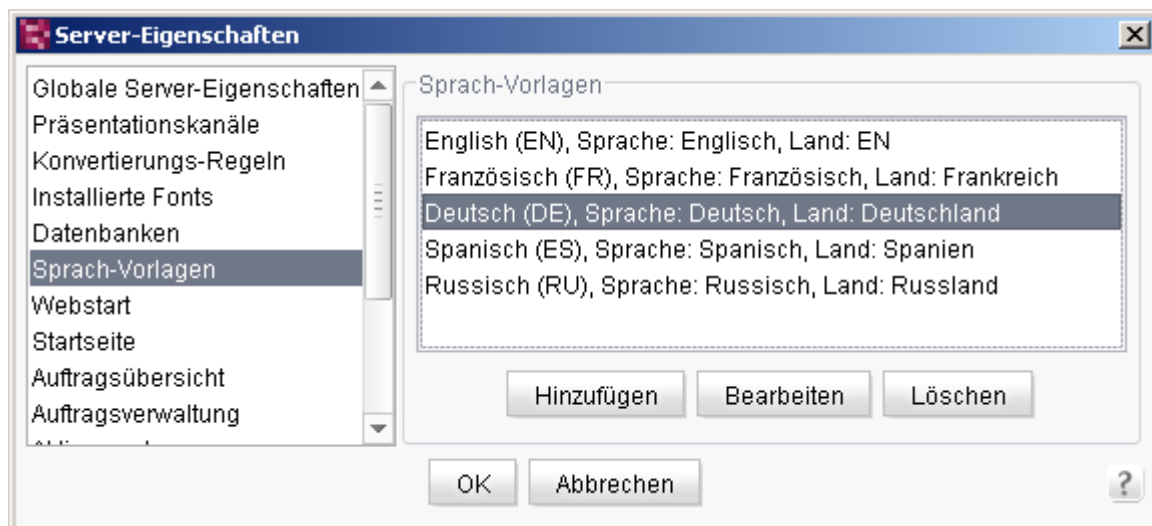


Abbildung 7-43: Server-Eigenschaften – Sprachvorlagen

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 7-44). 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 290). 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 7-44).

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 290), 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.

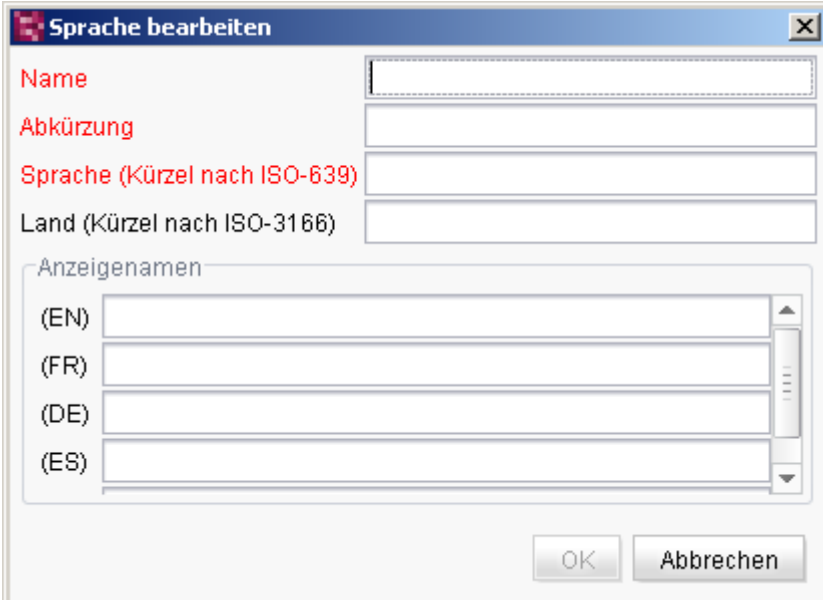


Abbildung 7-44: Server-Eigenschaften – neue Sprache anlegen

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 and 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.

Ab FirstSpirit Version 4.2 können für jede Sprach-Vorlage sprachabhängige Anzeigenamen definiert werden. Im Formularbereich "Anzeigenamen" werden dazu die entsprechenden Eingabefelder angezeigt. Sprachabhängige Anzeigenamen können zu allen Redaktionssprachen eines Projekts definiert werden.



7.3.8 Webstart

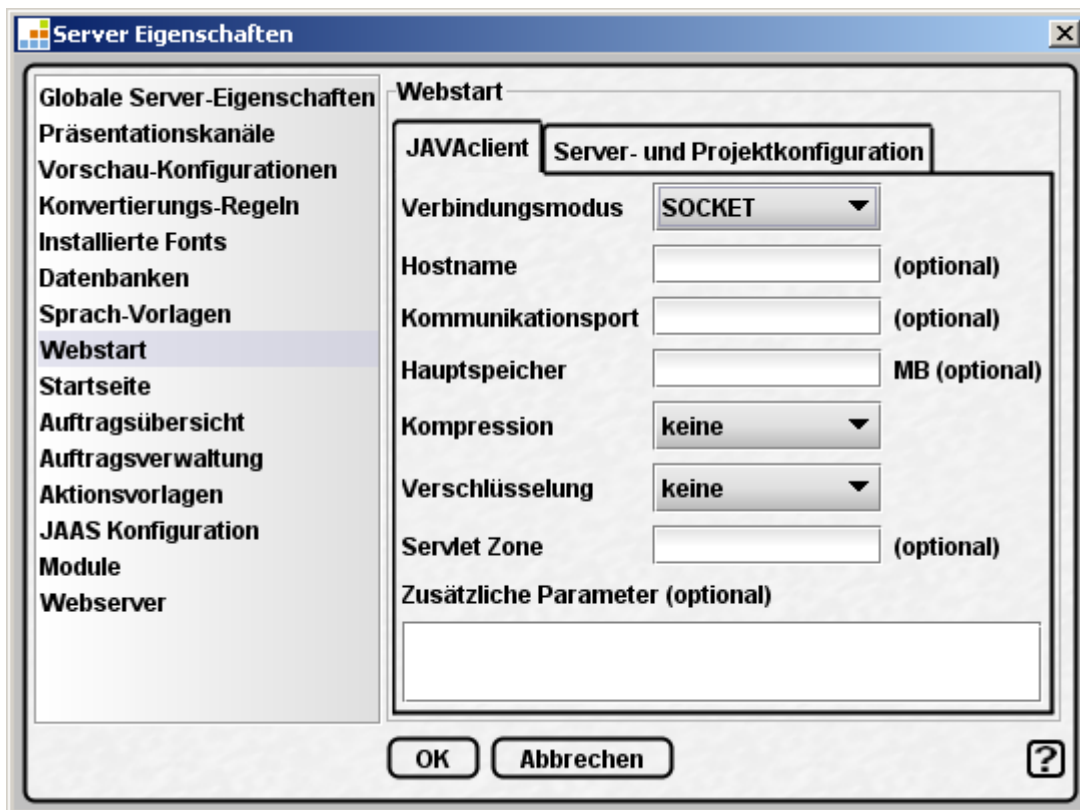


Abbildung 7-45: Server-Eigenschaften – Webstart konfigurieren (Schnellstart)

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 7.3.9 Seite 246).

Die Konfigurationsmöglichkeiten entsprechen der benutzerspezifischen Webstart-Konfiguration (siehe Kapitel 6.3.5.1 Seite 186).

Register Server- und Projektkonfiguration: Konfiguration für die Server- und Projektkonfiguration der Startseite. Die Parameter sind analog zu den Schnellstart-





Einträgen im Register "JavaClient".

7.3.9 Startseite



Abbildung 7-46: Server-Eigenschaften – Startseite konfigurieren

Schnellstart-Projekte: Über das Icon  können aus allen Projekten, die sich auf dem Server befinden, Projekte ausgewählt werden. Diese Projekte werden anschließend in der tabellarischen Übersicht dargestellt und erscheinen auf der FirstSpirit Startseite unterhalb der Schnellstart-Einträge (vgl. Kapitel 6.3.2 Seite184).

Mit einem Klick auf das 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 WebEdit-Konfiguration) referenziert werden.





Abbildung 7-47: 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 entweder mit dem Schriftzug `JAVA` oder mit dem Schriftzug `WEB` versehen. Für Schnellstart-Eintrag, die im JavaClient geöffnet werden, können weitere Verbindungseinstellungen (Einstellungen für Java Webstart) definiert werden. Der Dialog "Webstart Einstellungen" wird mit einem Doppelklick auf den entsprechenden Schnellstart-Eintrag geöffnet (siehe Abbildung 7-48). Die Konfigurationsmöglichkeiten entsprechen der benutzerspezifischen Webstart-Konfiguration (siehe Kapitel 6.3.5.1 Seite 186).



Abbildung 7-48: 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 35). Die dort definierten Parameter können über die Server- und Projektkonfiguration



überschrieben werden und zwar unter dem Menüeintrag "Webstart" einmal für alle Java Webstart-Anwendungen (über die JNLP-Dateien) (siehe Kapitel 7.3.8 Seite 245) 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 6.3.5.1 Seite 186), 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 35) nicht durch die entsprechenden Parameter, die in den Servereigenschaften in den Bereichen "Webstart" und "Startseite" definiert sind, überschrieben.

7.3.10 Auftragsübersicht

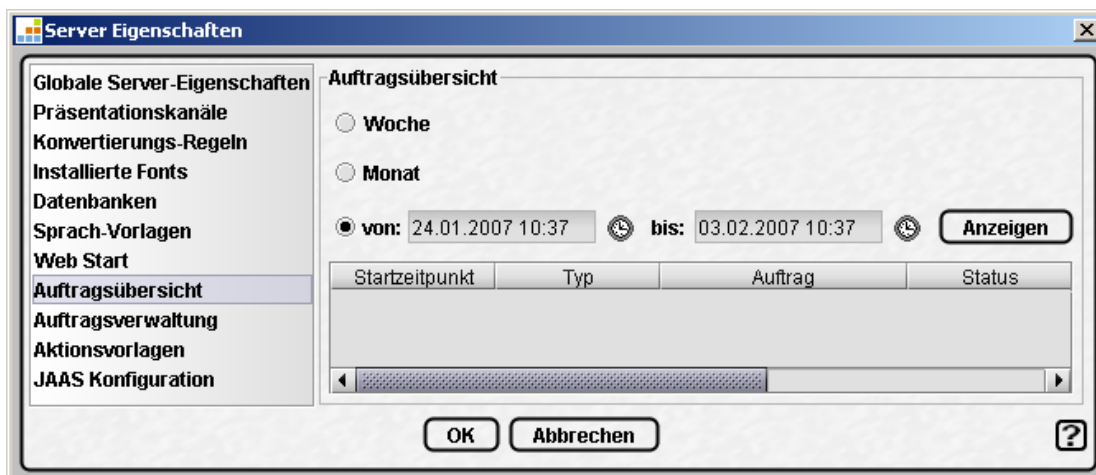


Abbildung 7-49: Server-Eigenschaften – Auftragsübersicht

Zur Auftragsübersicht siehe Kapitel 7.5.1 ab Seite 343.



7.3.11 Auftragsverwaltung



Abbildung 7-50: Server-Eigenschaften – Auftragsverwaltung

Zur Auftragsverwaltung siehe Kapitel 7.5.2 ab Seite 346.

7.3.12 Aktionsvorlagen

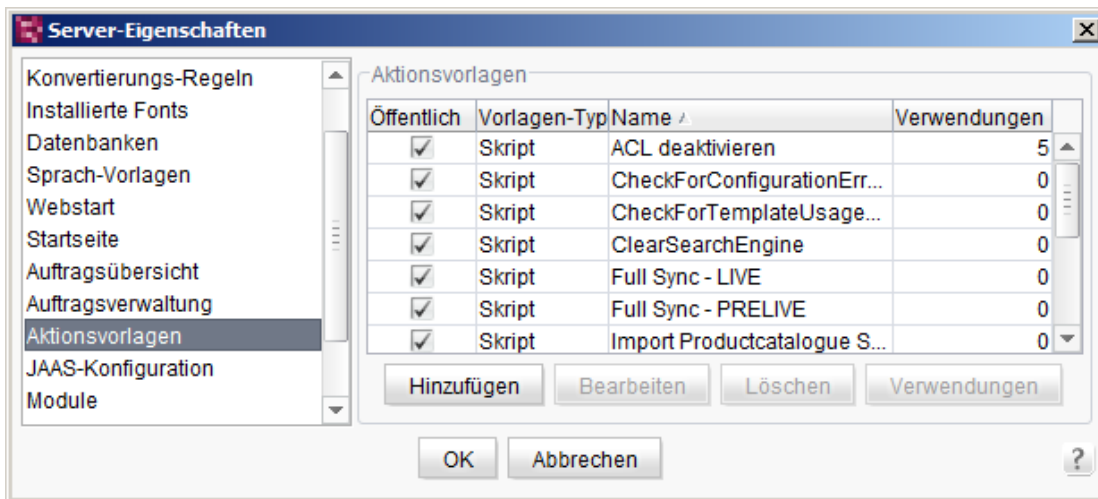
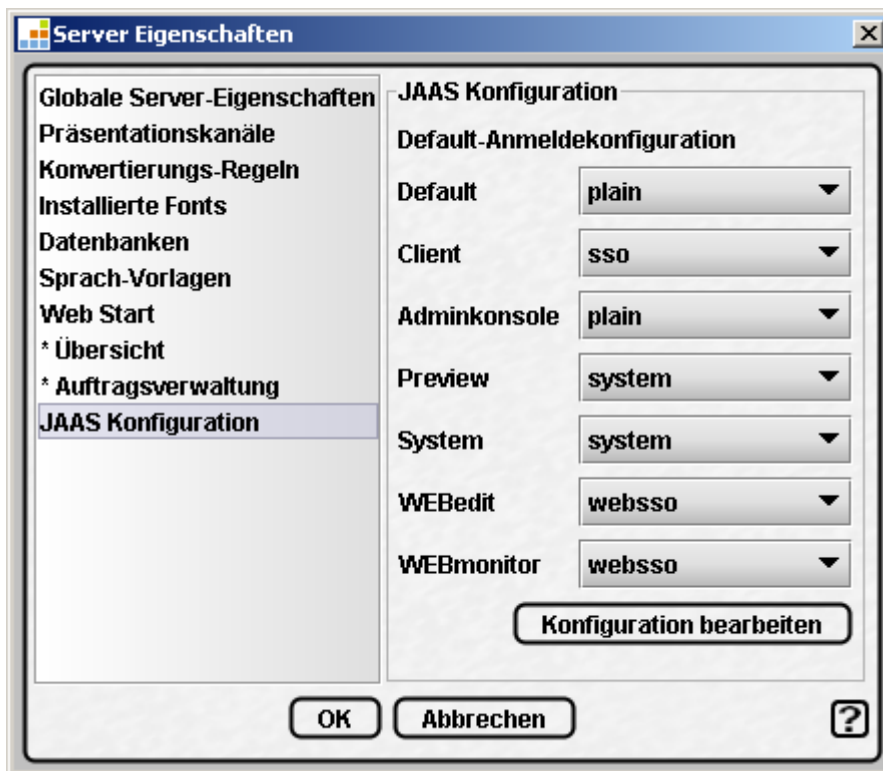


Abbildung 7-51: Server-Eigenschaften – Aktionsverwaltung

Zu Aktionsvorlagen siehe Kapitel 7.5.3 ab Seite 349.



7.3.13 JAAS-Konfiguration

Abbildung 7-52: Server-Eigenschaften – JAAS²⁵-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 76). 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 44).

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

²⁵ Java Authentication and Authorization Service



weitere Module definiert werden. Beim Bestätigen der Eingaben mit "OK" werden die Änderungen in der `fs-jaas.conf` gespeichert und automatisch geladen.

7.3.14 Module

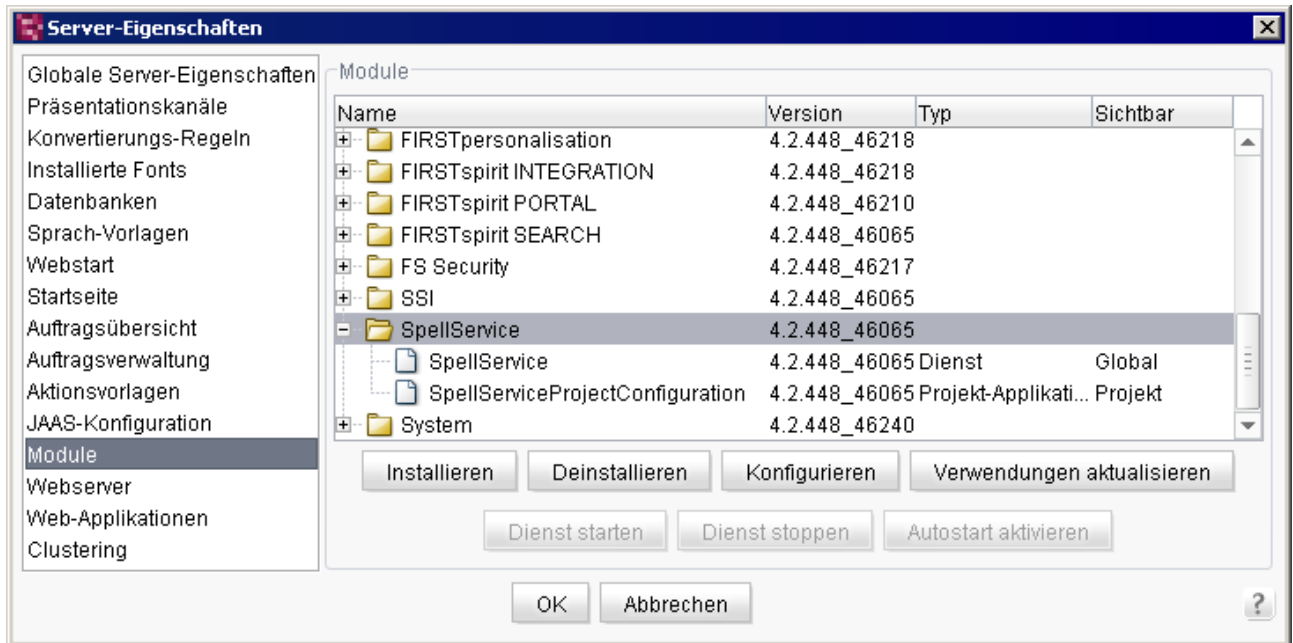




Abbildung 7-53: Server-Eigenschaften – Module

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 -Icon gekennzeichnet. Zu jedem Modul gehören eine oder mehrere Komponenten, die unterhalb des Moduls angezeigt werden. Komponenten werden mit einem -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 7.3.15 Seite 258).

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 319). 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 319). Die Konfigurationsmöglichkeit ist abhängig von der installierten Komponente. Die Komponenten können für die jeweiligen Projekte unterschiedlich konfiguriert werden.

Installieren: Mit einem Klick auf den Button 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 7-53).

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):

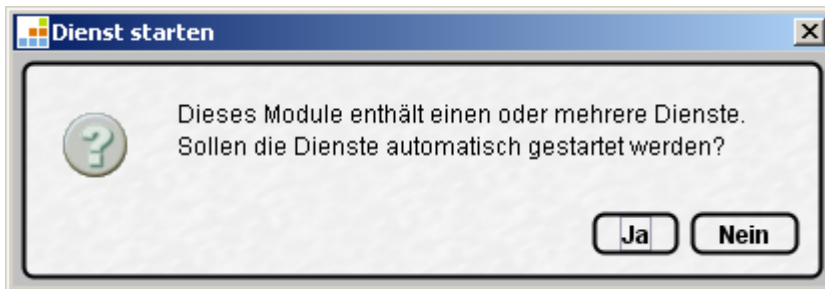


Abbildung 7-54: Autostart für Dienste installieren

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 319).

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"). Ab FirstSpirit Version 4.1 können außerdem Rechte für ein Modul gesetzt werden (siehe Kapitel 7.3.14.1 Seite 256):



Abbildung 7-55: Konfigurationsbeispiel Spell Service



Aktualisieren (bis V4.2R2 einschließlich): Steht eine neuere Version des Moduls zur Verfügung, kann das Modul mit einem Klick auf den Button auf dem FirstSpirit-Server aktualisiert werden. Die ursprüngliche Konfiguration bleibt bei einer Aktualisierung erhalten, es können jedoch Anpassungen der Konfiguration erforderlich werden. Die neue Version des Moduls wird nicht automatisch in die Projekte übernommen. Wurden Module einem Projekt bzw. einem Web-Bereich hinzugefügt, sind bei einer Aktualisierung eventuell Anpassungen innerhalb des Projekts erforderlich. Die Aktualisierungen innerhalb eines Projekts müssen daher manuell vorgenommen werden (siehe Kapitel 7.4.17 Seite 317 und Kapitel 7.4.18 Seite 319). Ab FirstSpirit Version 4.1 können die Verwendungen innerhalb der Projekte über den Button "Verwendungen aktualisieren" vorgenommen werden s.u.). **Ab Version 4.2R4** können Module nur noch installiert werden (Button "Installieren", siehe oben).



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 445.)

Verwendungen aktualisieren (ab V4.1): Das Aktualisieren von Modulen wird ab FirstSpirit Version 4.1 nochmals komfortabler. Für das Aktualisieren eines Moduls, das eine Projekt- oder eine Webanwendung beinhaltet, steht nun 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:

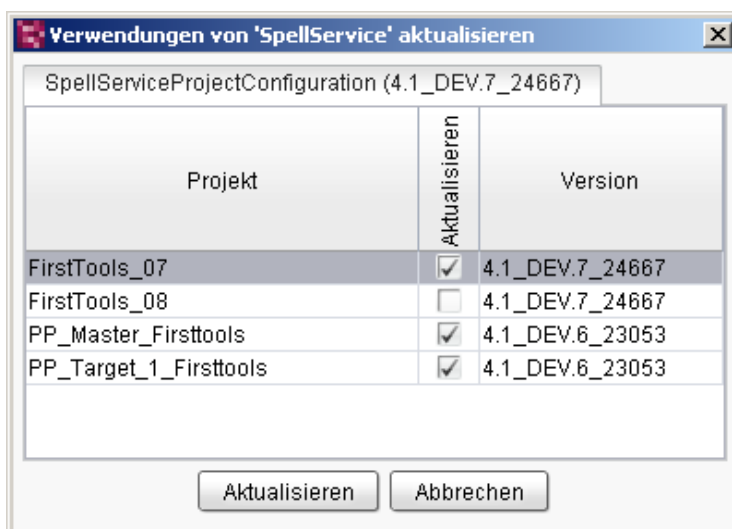


Abbildung 7-56: Alle Verwendungen eines Moduls aktualisieren (ab V4.1)



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 445).



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 445).

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.



7.3.14.1 Vertrauenswürdige Module (ab V4.1)

Die Entwicklung externer Module oder Komponenten für FirstSpirit wird ab Version 4.0 unterstützt²⁶. In **FirstSpirit Version 4.1** wird dieses Konzept um Sicherheitsaspekte erweitert.

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 in Version 4.1 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:

²⁶ Zur Konzeption und Entwicklung siehe "Entwicklerhandbuch für Komponenten" für FirstSpirit Version 4.0



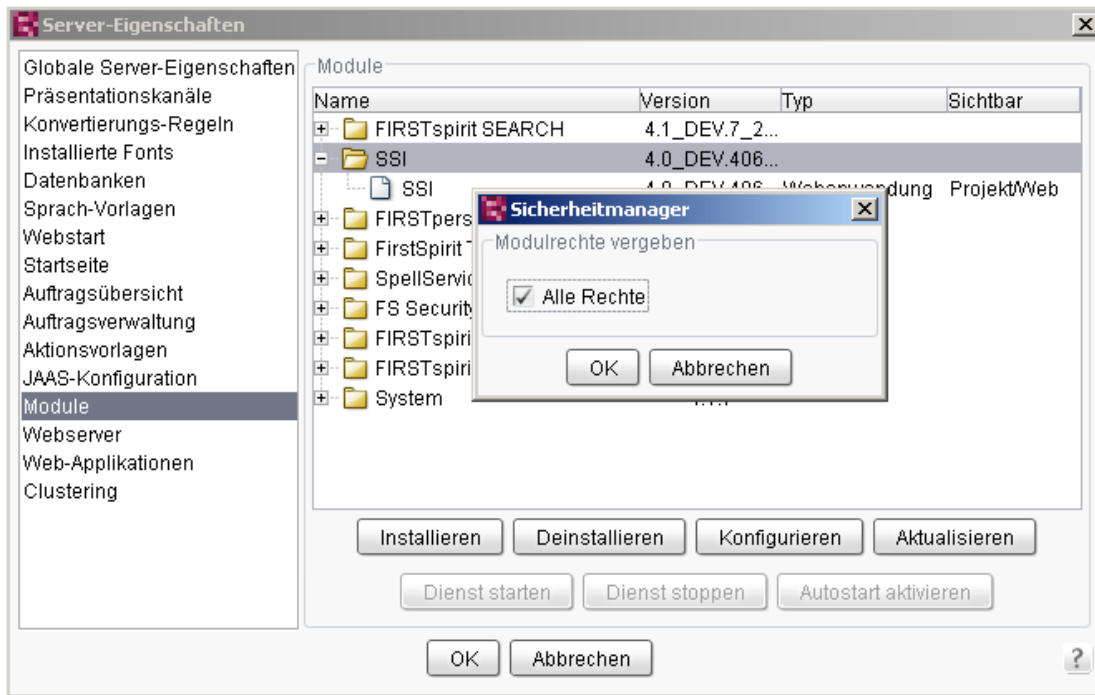


Abbildung 7-57: Sicherheitsmanager – Alle Rechte für dieses Modul vergeben



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".



7.3.15 Webserver

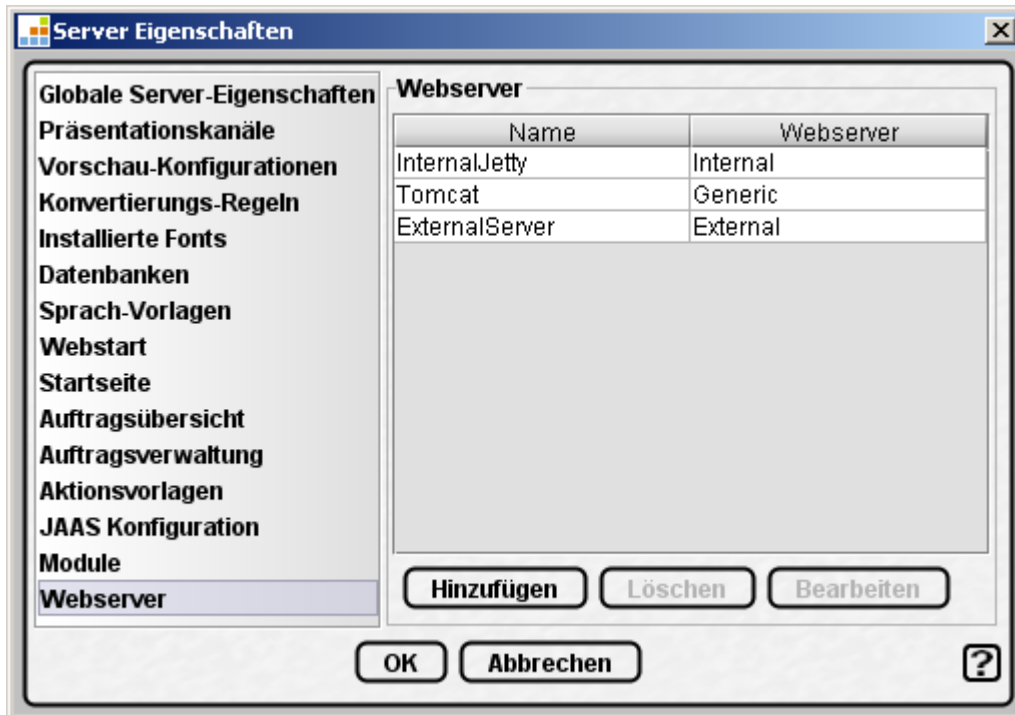


Abbildung 7-58: Server-Eigenschaften – Webserver

In diesem Bereich kann die Webserver-Steuerung für das Installieren und Deinstallieren innerhalb der Projektbereiche eingefügt werden (siehe Kapitel 7.4.17 Seite 317).

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 hinzugefügt und konfiguriert werden.

Name: Eindeutiger Name unter dem die Instanz des Webserver hinzugefügt wurde.



Webserver: Innerhalb des Standard-Moduls "System" stehen drei unterschiedliche Arten von Webservern zur Verfügung (vgl. Dialog "Module" in Kapitel 7.3.14 Seite 251):

- 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.
- 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 7.3.15.2 Seite 260)(weiterführende Informationen zu unterstützten Webservern siehe Technisches Datenblatt für FirstSpirit V4.x).

Hinzufügen

Mit einem Klick auf den Button 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 7.3.15.1 Seite 260)
- Externer Webserver (siehe Kapitel 7.3.15.3 Seite 263)

Löschen

Mit einem Klick auf den Button kann eine hinzugefügte Webserver-Instanz wieder entfernt werden. Die Standardinstanz "InternalJetty" kann nicht entfernt werden.

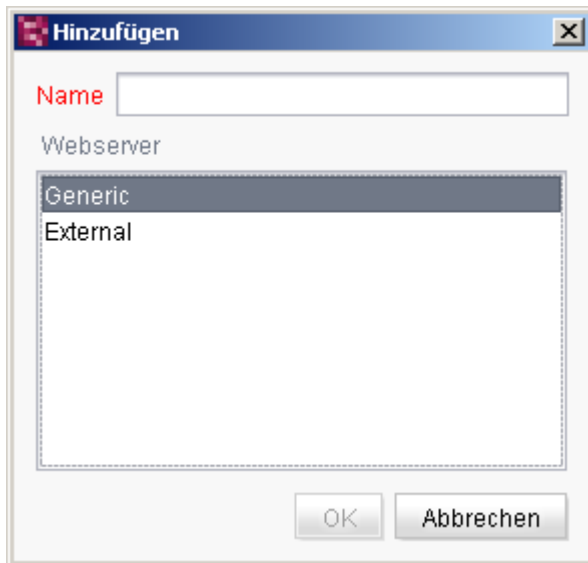
Konfigurieren

Mit einem Klick auf den Button 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 7.3.15.2 Seite 260).
- Externer Webserver: Eine Konfiguration für externe Webserver wird nicht unterstützt und muss manuell erfolgen (siehe Kapitel 7.3.15.4 Seite 263)



7.3.15.1 Hinzufügen eines generischen Webservers

**Abbildung 7-59: 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 7.3.15.2 Seite 260).

7.3.15.2 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:



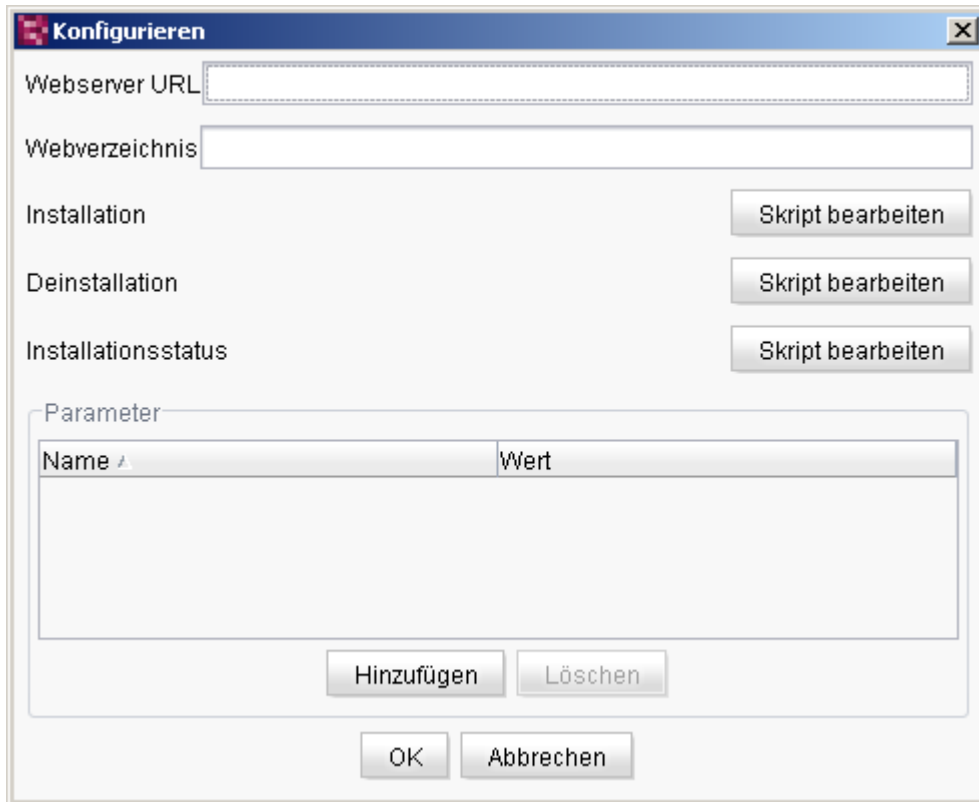


Abbildung 7-60: Webserver-Funktionen konfigurieren

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. fs4staging). Wird der generische Webserver beispielsweise für die Webanwendung fs4staging 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.





Abbildung 7-61: BeanShell Skript einfügen oder bearbeiten

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 7.3.16 Seite 264)
- Projekteigenschaften / Web-Komponenten (siehe Kapitel 7.4.18 Seite 319).

Zum Ausführen der Skripte kann das Webverzeichnis angegeben werden, indem sich der generische Webserver befindet (siehe "Webverzeichnis").

Parameter hinzufügen: Im Bereich "Parameter" (vgl. Abbildung 7-60) 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.



7.3.15.3 Hinzufügen eines externen Webserver

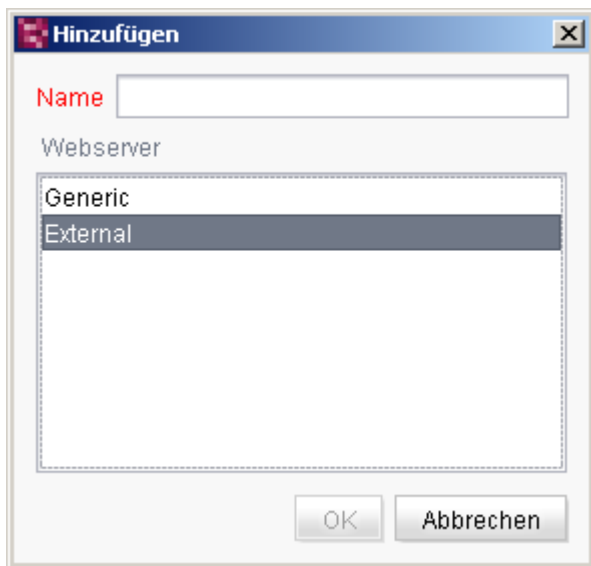



Abbildung 7-62: Externen Webserver hinzufügen

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 Webserver angegeben werden (siehe Kapitel 7.3.15.4 Seite 263). Eine weitere Unterstützung (Installation- bzw. Deinstallation) wird nicht angeboten und muss manuell erfolgen.

7.3.15.4 Konfigurieren eines externen Webserver



Abbildung 7-63: Konfiguration eines externen Webserver

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.

7.3.16 Web-Applikationen

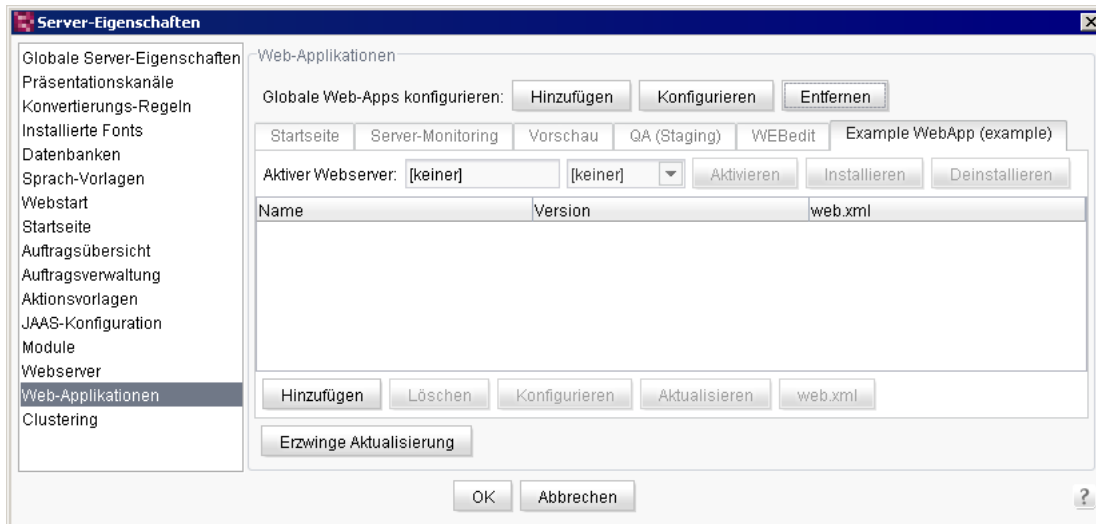


Abbildung 7-64: Server-Eigenschaften – Web-Applikationen

In diesem Bereich können die FirstSpirit Web-Applikationen konfiguriert werden. Ab Version 4.2R4 können auch eigene Web-Applikationen definiert und konfiguriert werden, die dann in allen Projekten des Servers zur Verfügung stehen ("global"). Damit können z. B. eigene Web-Anwendungen für das FirstSpirit AppCenter (siehe *FirstSpirit Release Notes Version 4.2R4*, Kapitel 3 "Das FirstSpirit AppCenter") installiert werden.

7.3.16.1 Installieren von globalen Web-Applikationen

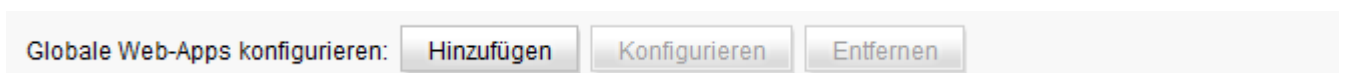
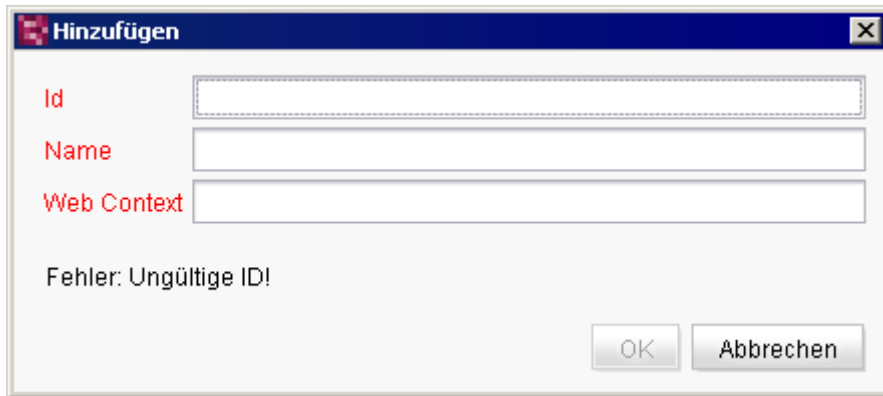


Abbildung 7-65: Globale Web-Apps konfigurieren

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





The screenshot shows a dialog box titled 'Hinzufügen' with a close button (X) in the top right corner. It contains three text input fields labeled 'Id', 'Name', and 'Web Context'. Below these fields, there is a red error message: 'Fehler: Ungültige ID!'. At the bottom right of the dialog, there are two buttons: 'OK' and 'Abbrechen'.

Abbildung 7-66: 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 Anzeigenname verwendet wird. Dieser kann später bei Bedarf geändert werden.

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

7.3.16.2 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 180).

QA (Staging): Konfiguration für die generierten Projektinhalte QA (Staging).

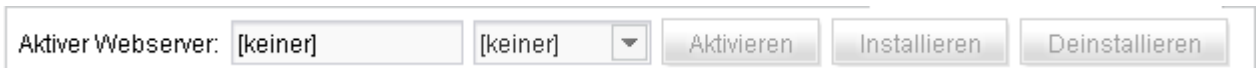
WebEdit: Konfiguration für das Redaktionssystem WebEdit.

Server-Monitoring: Konfiguration für das FirstSpirit Server-Monitoring (siehe Kapitel



8 Seite 413)

- Zu jeder Web-Applikation kann ein Webserver eingestellt werden:



Zur Auswahl stehen alle Webserver, die im Bereich "Webserver" konfiguriert wurden (vgl. Kapitel 7.3.15 Seite 258).

Abhängig von der Art des Webserver sind zur Umstellung unterschiedliche Schritte erforderlich:

- Interner Webserver (siehe Kapitel 7.3.16.3 Seite 267)
- Generischer Webserver (siehe Kapitel 7.3.16.4 Seite 267)
- Externer Webserver (siehe Kapitel 7.3.16.5 Seite 268)



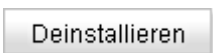
Mit einem Klick auf den Button wird die Konfiguration der Web-Applikation auf den selektierten Webserver umgestellt, der anschließend als aktiver Webserver für die Anwendung angezeigt wird. Die Verweise auf die Web-Applikation (z. B. auf der Startseite) werden dabei automatisch auf den neuen Webserver umgestellt.



Mit einem Klick auf den Button 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 7.3.16.5 Seite 268).



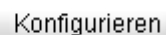
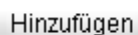
Mit einem Klick auf den Button kann eine WAR-Datei der Anwendung herunter geladen werden, die manuell auf dem Webserver installiert werden muss. Der Button wird nur für die Konfiguration externer Webserver oder die Konfiguration generischer Webserver (ohne die erforderliche Skript-Funktionalität) angezeigt.



Mit einem Klick auf den Button 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 auch aus der Konfigurationsdatei fs-server.conf entfernt (ab FirstSpirit Version 4.2.437 – in früheren Versionen wird der Beitrag auskommentiert).

- 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 319.

7.3.16.3 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 (z. B. Tomcat) 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  wird aktiv. Mit einem Klick auf den Button kann die Webapplikation auf dem internen Webserver installiert werden.
3. Nach der Installation wird der Button  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  bestätigt und gespeichert werden.

7.3.16.4 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 7.3.15.1 Seite 260). Die Steuerung für den generischen Webserver ist nicht standardmäßig vorhanden, kann aber über Skripte realisiert werden (siehe Kapitel 7.3.15.2 Seite 260). 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  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  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 7-60).
4. Alle Änderungen der Konfiguration müssen mit einem Klick auf  bestätigt und gespeichert werden.

7.3.16.5 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 7.3.15.3 Seite 263). Die Steuerung für einen externen Webserver wird über FirstSpirit nicht unterstützt und muss manuell erfolgen (siehe Kapitel 7.3.15.4 Seite 263).



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 Web-Anwendung `/fs4root` als Root-Application definiert werden kann, und z. B. direkt über <http://fs4.yourdomain.net> erreicht werden kann. Ein URL mit Präfix wie <http://appserver.yourdomain.net/fs4> 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  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 7-63).
5. Alle Änderungen der Konfiguration müssen mit einem Klick auf  bestätigt und gespeichert werden.



7.3.17 Clustering (ab V4.1)



Diese Funktionalität ist erst ab FirstSpirit-Version 4.1 freigegeben.

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 387.

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.

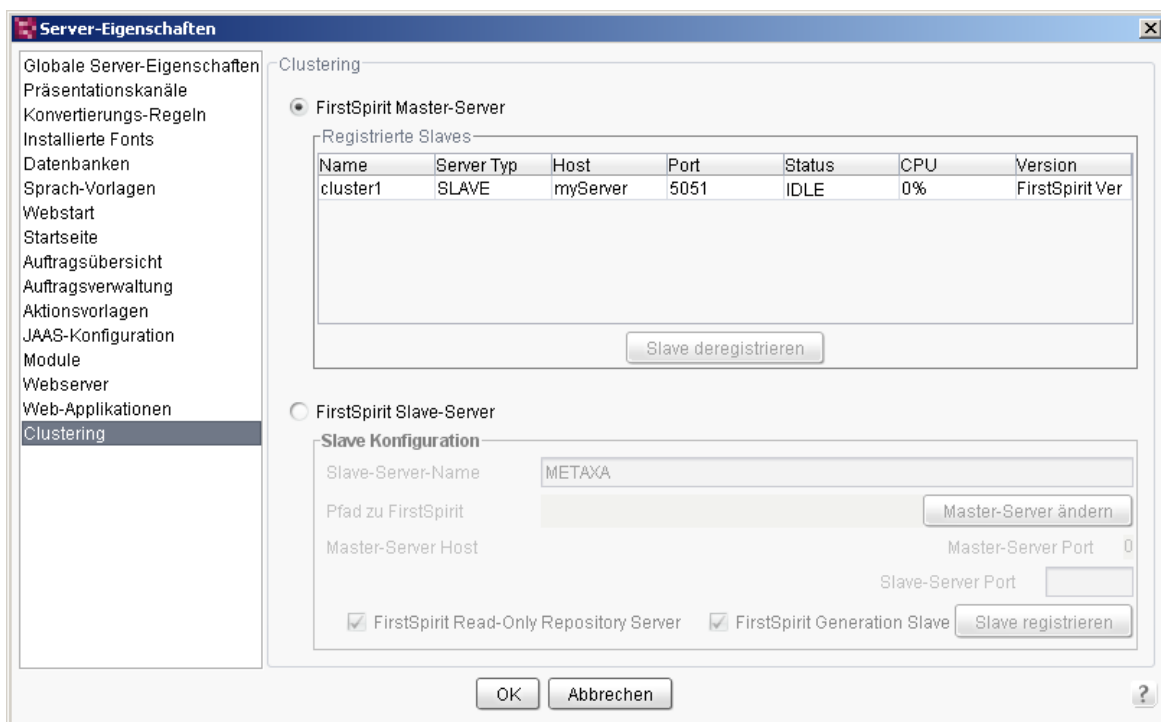


Abbildung 7-67: Servereigenschaften – Clustering

- Konfiguration des Master-Servers unter Unix: Kapitel 7.3.17.1
- Konfiguration des Slave-Servers unter Unix: Kapitel 7.3.17.2
- Konfiguration des Master-Servers unter Windows: Kapitel 7.3.17.3
- Konfiguration des Slave-Servers unter Windows: Kapitel 7.3.17.4



7.3.17.1 Konfiguration FirstSpirit Master-Server unter Unix

Zunächst wird ein FirstSpirit-Server wie im Installationshandbuch beschrieben, installiert. Für den Installationspfad wird als Beispiel `/opt/firstspirit4` angenommen und als Hostname `fs4host`. 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/firstspirit4/conf/fs-server.conf` ist folgende Zeile hinzuzufügen:

```
HOST=fs4host
```

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 `/opt/dfs/dfstab` sind folgende Zeilen hinzuzufügen, um einige ausgewählte Verzeichnisse des Master-Servers über NFS freizugeben:

```
share -F nfs -o ro /opt/firstspirit4/bin
share -F nfs -o ro /opt/firstspirit4/conf
share -F nfs -o ro /opt/firstspirit4/server
share -F nfs -o ro /opt/firstspirit4/shared
share -F nfs -o ro /opt/firstspirit4/data/projects
share -F nfs -o rw /opt/firstspirit4/data/schedule
share -F nfs -o rw /opt/firstspirit4/web/fs4staging
share -F nfs -o rw /opt/firstspirit4/log
share -F nfs -o rw /opt/firstspirit4/archive
share -F nfs -o rw /opt/firstspirit4/backup
share -F nfs -o rw /opt/firstspirit4/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 `fs4 start` (bzw. `svcadm enable fs4`) 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 458).

Folgende Log-Dateien des Masters dienen zur Fehleranalyse bei Problemen während und nach der Konfiguration:

```
/opt/firstspirit4/log/fs-wrapper.log
/opt/firstspirit4/log/fs-server.log
```

7.3.17.2 Konfiguration FirstSpirit Slave-Server unter Unix

Zunächst wird ein FirstSpirit-Server wie im Installationshandbuch beschrieben, installiert. 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/firstspirit4` angenommen und als Hostname `fs4slave`. Nach der Installation wird der FirstSpirit-Server heruntergefahren, um durch die nachfolgend beschriebenen 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/firstspirit4/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, unter Fortführung der fortlaufenden Nummerierung der bereits vorhandenen Einträge mit Namen `wrapper.java.additional`, hinzufügen:

```
wrapper.java.additional.12=-Dcmsroot=/import/fs4master
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/firstspirit4/conf/fs-server.conf` wird für den Slave-Server nicht benötigt und sollte gelöscht werden:

```
rm /slave/firstspirit4/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/firstspirit4
rm -r bin server shared
mkdir -p /slave/firstspirit/bin
mkdir -p /slave/firstspirit/server
mkdir -p /slave/firstspirit/shared
mkdir -p /import/fs4master/conf
mkdir -p /import/fs4master/data/projects
mkdir -p /import/fs4master/data/schedule
mkdir -p /import/fs4master/web/fs4staging
mkdir -p /import/fs4master/log
mkdir -p /import/fs4master/archive
mkdir -p /import/fs4master/backup
mkdir -p /import/fs4master/export
```



In der Datei `/etc/vfstab` sind nun folgende Einträge notwendig, um die vom Master-Server bereitgestellten Verzeichnisse zu verwenden:

```
fs4host:/opt/firstspirit4/bin - /slave/firstspirit4/bin nfs - yes ro,hard,intr
fs4host:/opt/firstspirit4/server - /slave/firstspirit4/server nfs - yes ro,hard,intr
fs4host:/opt/firstspirit4/shared - /slave/firstspirit4/shared nfs - yes ro,hard,intr
fs4host:/opt/firstspirit4/conf - /import/fs4master/conf nfs - yes ro,hard,intr
fs4host:/opt/firstspirit4/data/projects - /import/fs4master/data/projects nfs - yes ro,hard,intr
fs4host:/opt/firstspirit4/data/schedule - /import/fs4master/data/schedule nfs - yes rw,hard,intr
fs4host:/opt/firstspirit4/web/fs4staging - /import/fs4master/web/fs4staging nfs - yes rw,hard,intr
fs4host:/opt/firstspirit4/log - /import/fs4master/log nfs - yes rw,hard,intr
fs4host:/opt/firstspirit4/archive - /import/fs4master/archive nfs - yes rw,hard,intr
fs4host:/opt/firstspirit4/backup - /import/fs4master/backup nfs - yes rw,hard,intr
fs4host:/opt/firstspirit4/export - /import/fs4master/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 `fs4 start` (bzw. `svcadm enable fs4`) 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 458).

Folgende Log-Dateien des Slaves dienen zur Fehleranalyse bei Problemen während und nach der Konfiguration:

```
/slave/firstspirit4/log/fs-wrapper.log
/import/fs4master/log/fs4slave/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 7.6.5 Seite 391).



7.3.17.3 Konfiguration Master-Server unter Windows

Ist der Radiobutton "FirstSpirit Master-Server" ausgewählt (vgl. Abbildung 7-67), 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 7-68):

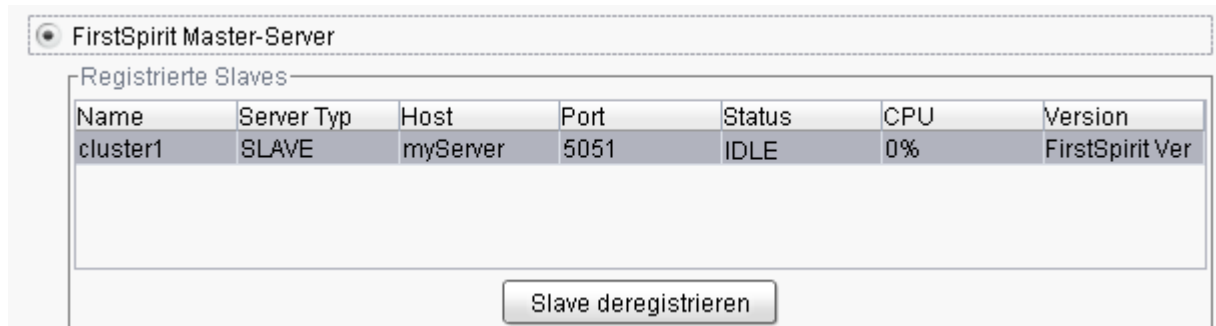


Abbildung 7-68: Konfiguration als FirstSpirit Master-Server

Mit einem Klick auf den Button kann ein "Registrierter Slave" wieder aus der Auswahl entfernt werden.

7.3.17.4 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.

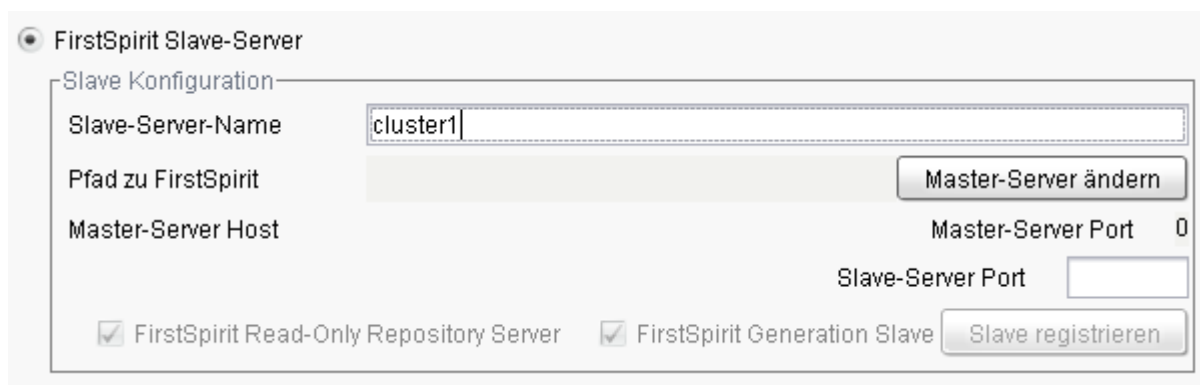


Abbildung 7-69: Konfiguration als FirstSpirit-Slave-Server



Slave Server Name: Name des FirstSpirit-Slaves.


Pfad zu FirstSpirit: Mithilfe des Buttons  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 7-70: 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:

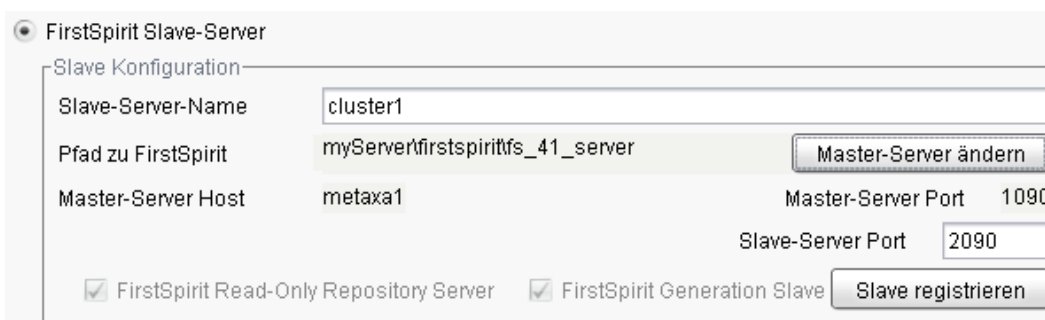


Abbildung 7-71: Konfiguration als FirstSpirit-Slave-Server

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 RORS²⁷. 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 öffnet sich eine Sicherheitsabfrage:

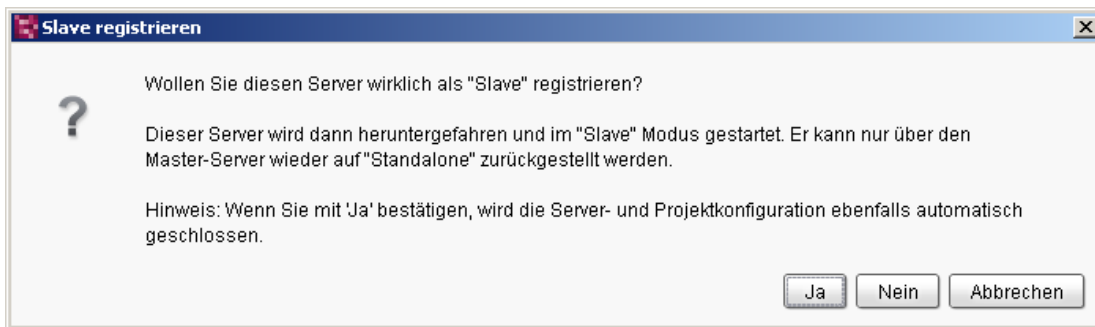


Abbildung 7-72: Slave registrieren

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.

²⁷ ReadOnly-Repository-Server (siehe Kapitel 7.6.2 Seite 388)



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 213) oder über den Button "Eigenschaften ändern".

Eigenschaften ändern Ein Klick auf den Button 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 197 und Kapitel 7.4.2 Seite 280).

Nach dem Aufruf erscheint das Dialogfenster "Projekt bearbeiten" (siehe Abbildung 7-74). 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 7-73: Fehlermeldung beim Versuch parallelen Bearbeitens



7.4.1 Projekt

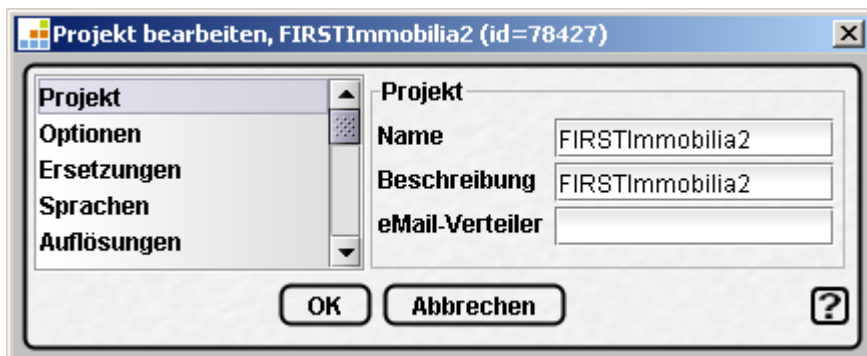


Abbildung 7-74: Projekteigenschaften – Projekt

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



Die Funktionalität ist teilweise erst ab FirstSpirit-Version 4.1 freigegeben. Daher werden Screenshots im neuen Look & Feel "LightGray" dargestellt. Im Look & Feel "Classic" kann die Darstellung geringfügig abweichen.

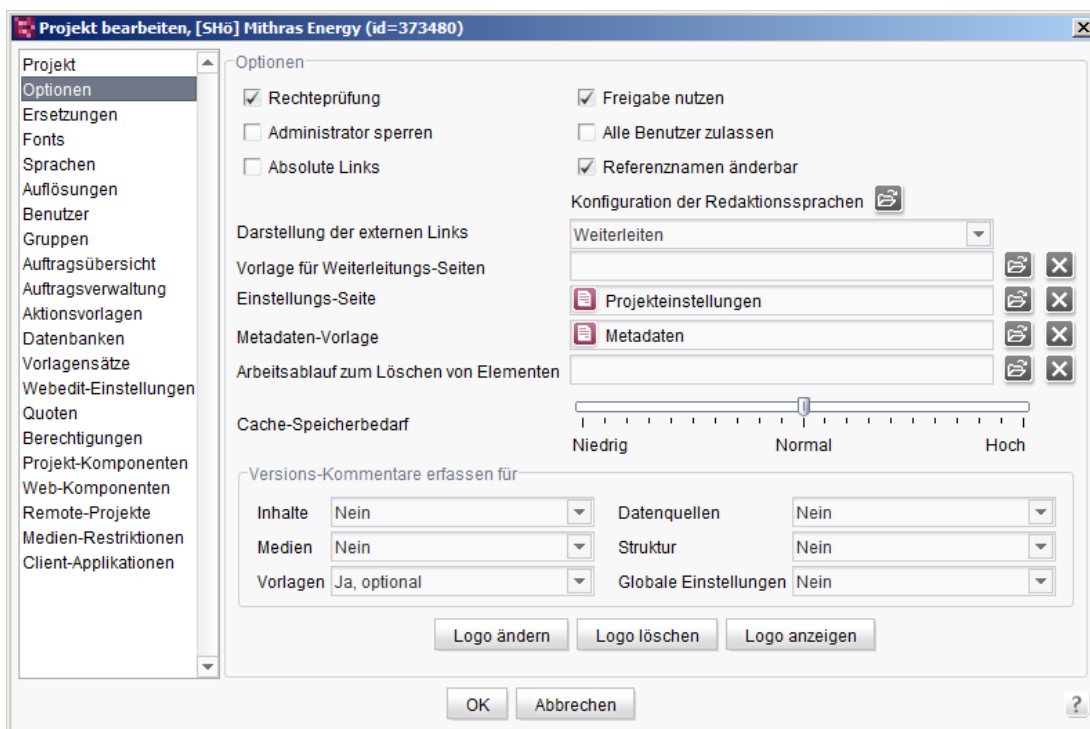


Abbildung 7-75: Projekteigenschaften – Optionen (Version 4.2R4)

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. **Ab FirstSpirit Version 4.2** ist diese Option bei der Neuanlage eines Projekts standardmäßig aktiviert, zuvor war sie standardmäßig deaktiviert.



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.

Wird die Option *neu aktiviert*, erfolgt eine Abfrage "Soll eine Freigabeversion von allen Verwaltungen erstellt werden?" Wird "Ja" gewählt, wird eine Freigabeversion aller Verwaltungen, ab FirstSpirit Version 4.1 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.

Ab FirstSpirit Version 4.2 ist diese Option bei der Neuanlage eines Projekts standardmäßig aktiviert, zuvor war sie standardmäßig deaktiviert.



Eine Deaktivierung dieser Option ist nur für kleinere Projekte sinnvoll und hat negative Auswirkungen auf die Gesamtperformance des Systems, da bei jeder Änderung eine automatische Freigabe erfolgt.

MultiViews Pfade: Diese Option unterstützt die Funktion "apache multiview" für die Erzeugung mehrsprachiger Seiten. Dabei werden die Projektdateien mit einer Spracherweiterung versehen, z. B. `index.html.de` und `index.html.en`. Dieser URL wird vom Webserver automatisch ausgewertet (sofern der Webserver "multiviews" unterstützt, z. B. der Apache Webserver). Der Webserver entscheidet dann anhand der Daten, die durch die Anforderung übermittelt werden, welche Inhalte bzw. in welcher Sprache der Benutzer die Inhalte angezeigt bekommt. Längere URL-Konstrukte der Form `.../de/index.html` oder `.../en/index.html` können damit vermieden werden (siehe Kapitel 7.5.9.2 Seite 368).



Diese Option entfällt für FirstSpirit Version 4.2. Um weiterhin Multiviews Pfade erzeugen zu können, kann in einer Aktion "Generierung ausführen" die Option "Multiview URLs" ausgewählt werden (siehe Kapitel 7.5.9.2 Seite 368, Option "Pfaderzeugung").



Administrator sperren: Ist die Checkbox *aktiviert*, wird der Zugriff des Serveradministrators auf dieses Projekt gesperrt (siehe Kapitel 7.1 Seite 197). 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*, dann kann der Serveradministrator außerdem die Aufgaben eines Projektadministrators wahrnehmen, und das Projekt über die Projektauswahlliste im JavaClient und in WebEdit 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 297) 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 5.2.3 Seite 298). Ist die Checkbox *deaktiviert*, müssen Benutzer, die Zugriff auf das Projekt erhalten sollen, explizit dem Projekt hinzugefügt werden (siehe Kapitel 7.4.7 Seite 297).

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 7.5.9.2 Seite 368). 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).

3.1 Kompatibilitätsmodus aktivieren: Durch Aktivieren dieser Option ist es möglich, die Option "Alle Medien in diesem Ordner bei der Generierung kopieren" innerhalb der Medien-Verwaltung weiterhin zu verwenden. Diese Funktion wird ab FirstSpirit Version 4.1 durch die Möglichkeit ersetzt, die Mediengenerierung in einen separaten Auftrag auszulagern (vgl. Kapitel 7.5.9.2 Seite 368).



Der Kompatibilitätsmodus für die FirstSpirit Version 3.1 entfällt für FirstSpirit Version 4.2.

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, in dem man die Option "Referenznamen anzeigen" in den Projekteigenschaften, Bereich "Optionen", "Konfiguration der Redaktionssprachen", deaktiviert. Dann ist eine Änderung nur über BeanShell (API) möglich.


Konfiguration der Redaktionssprachen: Mit einem Klick auf das Icon  hinter dem Eintrag "Konfiguration der Redaktionssprachen", öffnet sich ein Konfigurationsdialog:





Abbildung 7-76: 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 Redaktionssprachen für ein Projekt festgelegt werden. Die "aktivierten" Sprachen können anschließend in der FirstSpirit Redaktionsumgebung als "Bevorzugte Anzeigesprache" eingestellt werden (über das Menü "Extras"). Die Redaktionssprachen 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 Redaktionssprachen sind nicht zu verwechseln mit den Projektsprachen (vgl. Kapitel 7.4.5 Seite 290).

Name / Abkürzung: Name und Kürzel der gewünschten Redaktionssprache.

Mastersprache: Die aktivierte Checkbox in dieser Spalte kennzeichnet die Projekt-Mastersprache. Diese ist als Redaktionssprache standardmäßig aktiviert.

Darstellung der externen Links: Die Auswahl ist relevant, wenn in der Strukturverwaltung 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 ein Vorlage ausgewählt werden, die für die Metadaten des Projektes verwendet werden soll.



Im Gegensatz zur Auswahl von Seitenvorlagen in der Redaktionsumgebung JavaClient, 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 (ab V4.1): 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 7-77: Projekteigenschaften – Einstellung des Cache-Speicherbedarfs

Cache-Speicherbedarf: Über den Schieberegler kann der Anteil der zugeteilten Speichergröße für den Cache des Projekts erhöht (bei hoher Gewichtung) oder



verringert (bei niedriger Gewichtung) werden. Der vom CacheSizeManager zugeteilte Speicher ist außerdem noch abhängig vom real benötigten Speicher ("Preferred Size") und dem zur Verfügung stehenden Gesamtspeicher (vgl. Kapitel 9.5 Seite 468).

Versions-Kommentare erfassen für (ab FirstSpirit Version 4.1): 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.



Die Funktionalität "Versions-Kommentare erfassen für" ist erst ab FirstSpirit-Version 4.1 freigegeben. Daher werden Screenshots im neuen Look & Feel "LightGray" dargestellt. Im Look & Feel "Classic" kann die Darstellung geringfügig abweichen.



Abbildung 7-78: Versions-Kommentar eingeben

Folgende Optionen sind möglich:

Ja, erzwingen: Wenn diese Option aktiviert ist, wird dem Redakteur die in Abbildung 7-78 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 7-78 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 7-79 "Das ist ein Kommentar!"):





Abbildung 7-79: Änderungskommentar in der Versionshistorie



Abbildung 7-80: 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

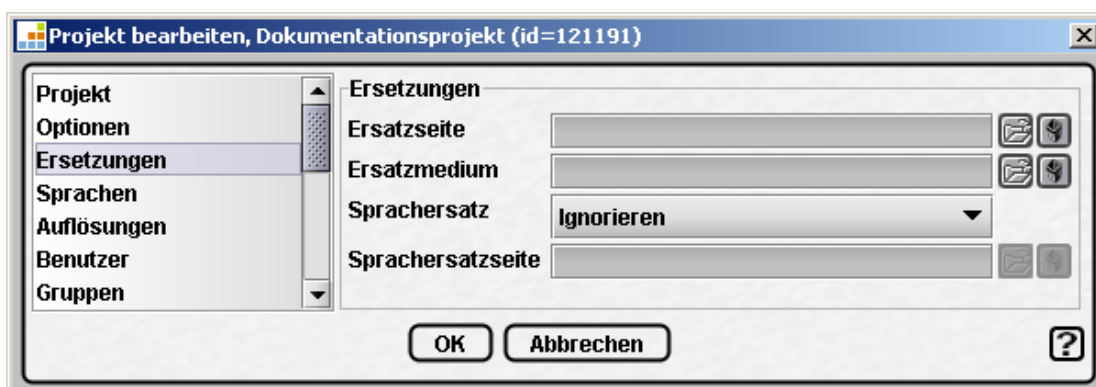


Abbildung 7-81: Projekteigenschaften – Ersetzungen

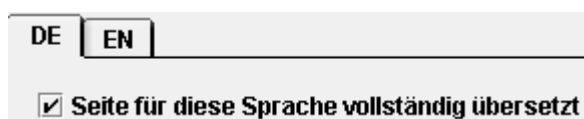
Ersatzseite: Sollte eine vom Besucher der Website angeforderte Seite einmal nicht



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



DE EN

Seite für diese Sprache vollständig übersetzt

Abbildung 7-82: 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 man sich die Verwendung dieser Option gründlich überlegen.
- **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 Struktur-Verwaltung zur Anzeige ausgewählt werden.



7.4.4 Fonts

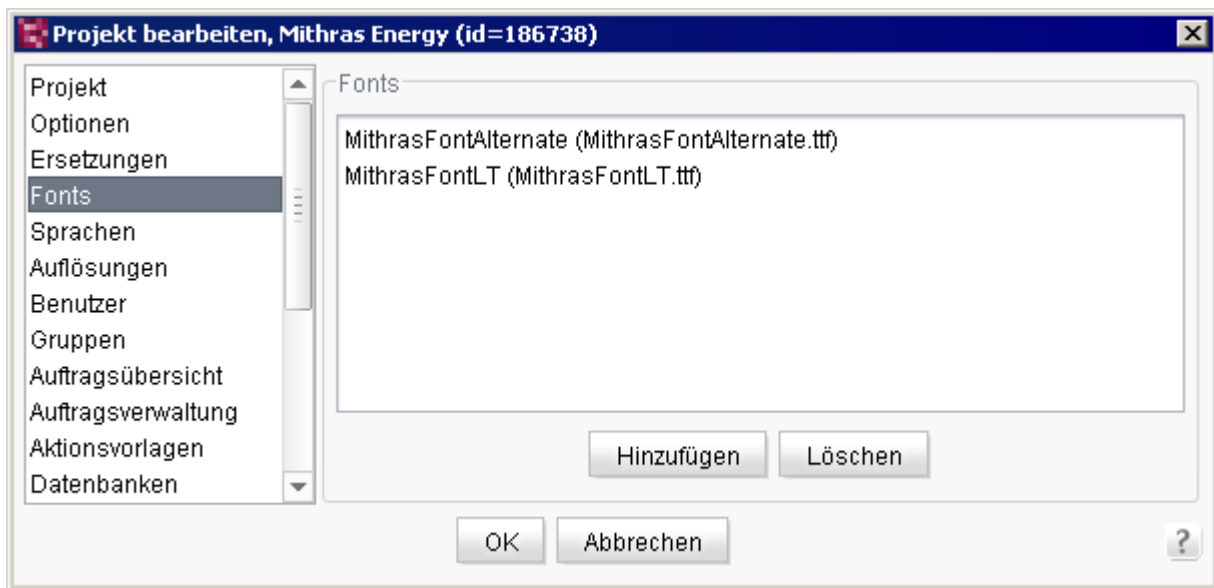


Abbildung 7-83: Projekteigenschaften – Fonts

Hier können aus den Fonts, die auf dem Server installiert sind (siehe Kapitel 7.3.5 Kapitel 236), über den Button "Hinzufügen" diejenigen ausgewählt werden, die für das aktuelle Projekt verfügbar sein sollen.



7.4.5 Sprachen

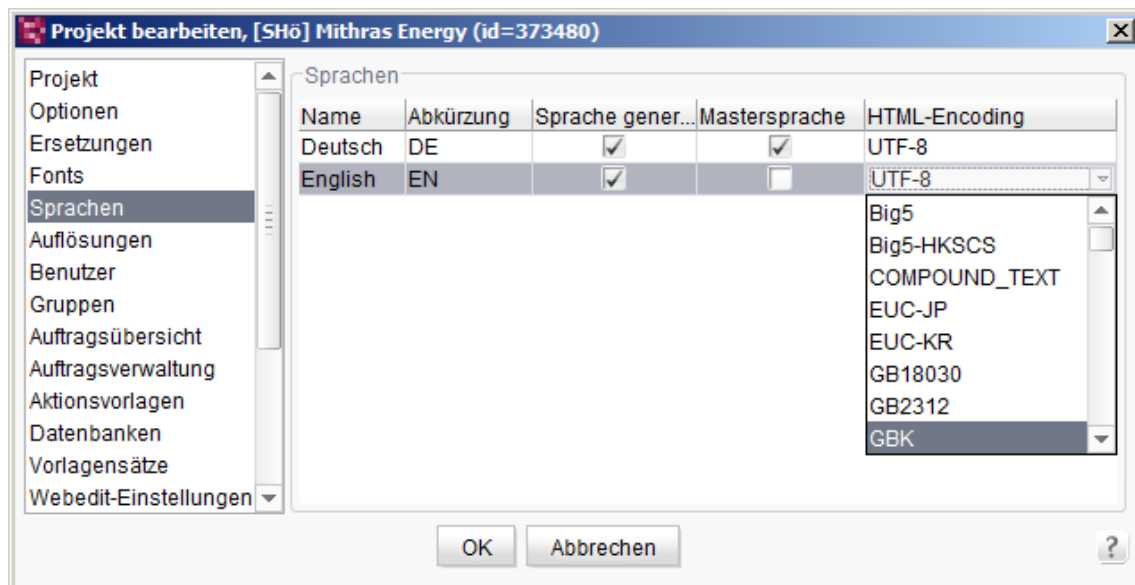


Abbildung 7-84: Projekteigenschaften – Sprachen

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 7.3.7 Seite 242).

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: Diese Spalte gibt an, in welchem Encoding die HTML-Seite später erzeugt werden soll. Die Voreinstellung für jede neu hinzugefügte Sprache ist ISO-8859-1, was dem westeuropäischen Sprachraum entspricht.

Mit FirstSpirit Version 4.2R4 wurde das Textfeld in eine Combobox geändert. In



dieser Combobox werden nun alle Encodings angezeigt, die von der jeweiligen Java-Version, unter der der FirstSpirit-Server läuft, unterstützt werden. Mit einem Klick auf das Feld öffnet sich die Combobox, und es kann das gewünschte Encoding ausgewählt werden.

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

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

ausgegeben.

Mit einem rechten Mausklick im Übersichtsfenster wird das Kontextmenü aufgerufen:



Abbildung 7-85: 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!

Löschen: Mit dieser Funktion wird die markierte Sprache aus dem Projekt entfernt.





Beim Entfernen einer Sprache können Projektinhalte 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 und ab FirstSpirit Version 4.2 auch die sprachabhängige Anzeigenamen) geändert werden, mit Ausnahme des Kürzels, das zur eindeutigen Identifizierung der Sprache im Projekt benötigt wird und nicht verändert werden darf.

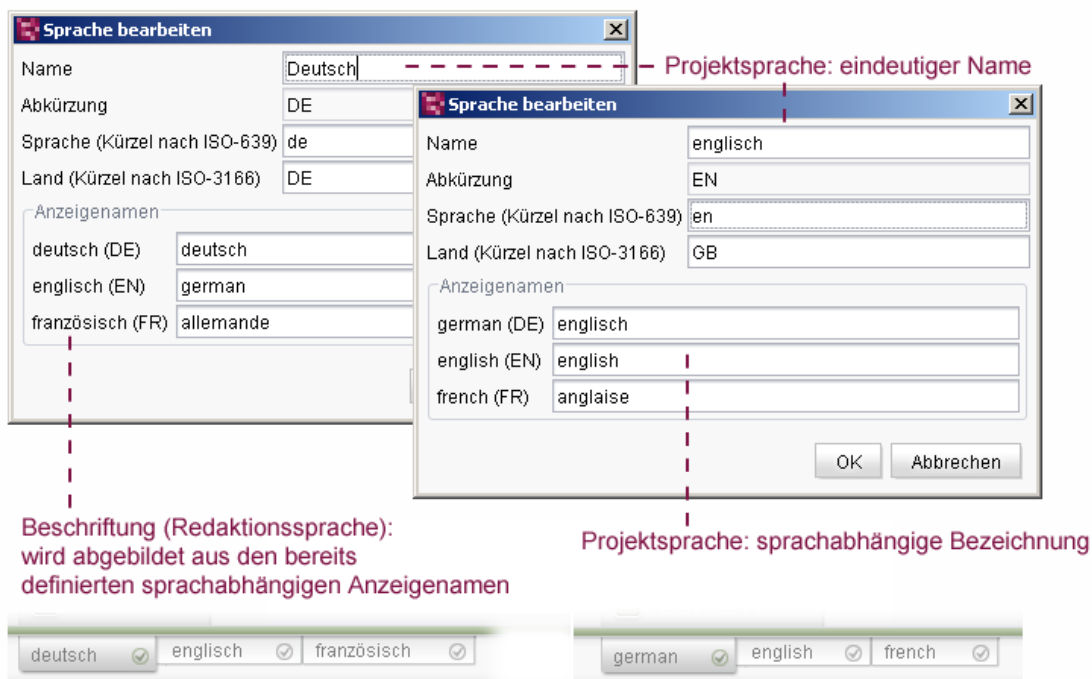


Abbildung 7-86: Sprachen bearbeiten

Ab FirstSpirit Version 4.2 können für eine Projektsprache sprachabhängige Anzeigenamen definiert werden. Im Formularbereich "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

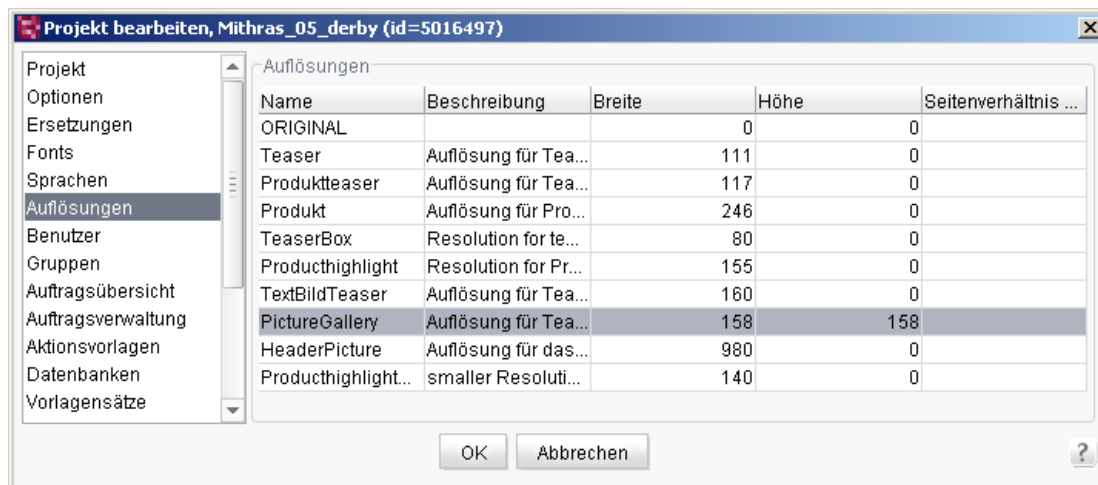


Abbildung 7-87: Projekteigenschaften – Auflösungen

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: Hier wird die Breite einer Auflösung in Pixel angezeigt. Eine "0" symbolisiert, dass sich die Breite aus dem Seitenverhältnis zur Originalauflösung ergibt.

Höhe: Hier wird die Höhe einer Auflösung in Pixel angezeigt. Eine "0" symbolisiert, dass sich die 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:

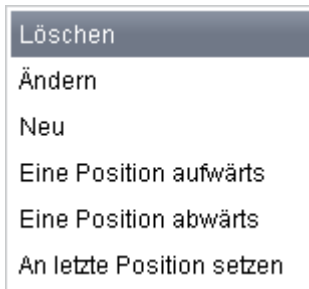


Abbildung 7-88: Kontextmenu (Bereich Auflösungen)

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. Ab FirstSpirit Version 4.2 können darüber die sprachabhängigen Anzeigenamen und die sprachabhängigen Beschreibungen für eine Auflösung geändert werden. Alle anderen Eigenschaften können nachträglich nicht mehr verändert werden.

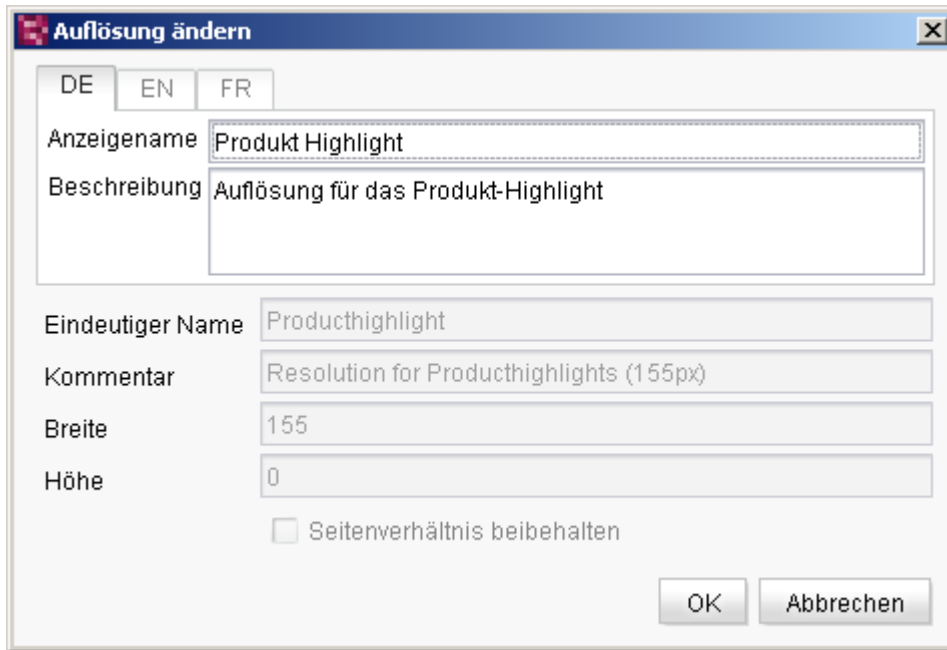
Neu: Mithilfe dieser Funktion kann dem Projekt eine neue Auflösung hinzugefügt werden. Durch Aufrufen der Funktion öffnet sich ein Fenster in dem die Einstellungen für die neue Auflösung vorgenommen werden können.

Neben den bekannten Eingabefeldern für den eindeutigen, technischen Namen, die Beschreibung und die Definition der Breite und Höhe einer Auflösung, können **ab FirstSpirit Version 4.2** zusätzlich sprachabhängige Anzeigenamen und sprachabhängige Beschreibungen für eine Auflösung in allen Redaktionssprachen definiert werden.

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.





Auflösung ändern

DE EN FR

Anzeigename Produkt Highlight

Beschreibung Auflösung für das Produkt-Highlight

Eindeutiger Name Producthighlight

Kommentar Resolution for Producthighlights (155px)

Breite 155

Höhe 0

Seitenverhältnis beibehalten

OK Abbrechen

Abbildung 7-89: 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

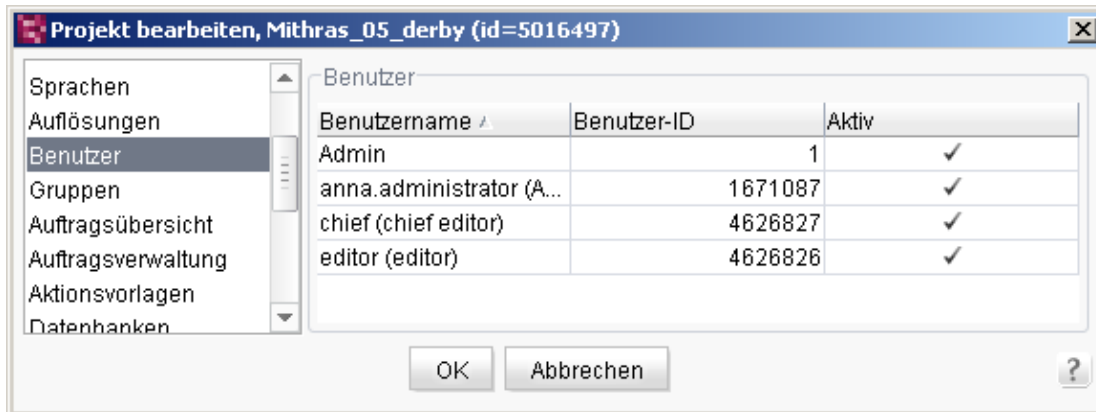


Abbildung 7-90: Projekteigenschaften – Benutzer

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 7-23). 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:

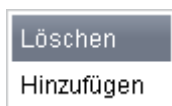


Abbildung 7-91: Kontextmenü (Bereich Benutzer)

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 5.2.3 Seite 298).

FirstSpirit unterscheidet zwischen Benutzern, die manuell auf dem FirstSpirit Server angelegt wurden und solchen, die automatisch aus einem Fremdsystem importiert wurden:

3. 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 219). Ein manuell auf dem Server angelegter Benutzer kann, innerhalb der Projekteigenschaften, einem Projekt zugeordnet werden (siehe 7.4.7 Seite 297) 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 305).
4. 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 301).



7.4.8 Gruppen

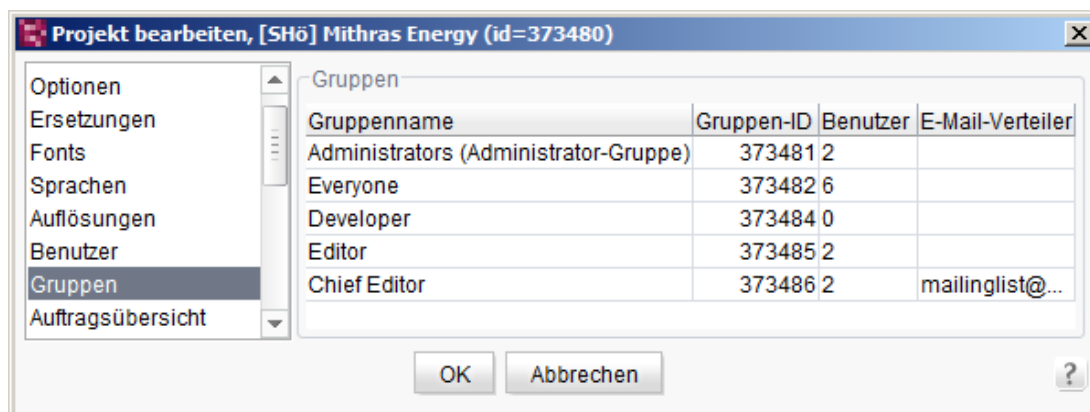


Abbildung 7-92: Projekteigenschaften – Gruppen

In diesem Bereich werden alle Gruppen aufgeführt, die Zugriff auf das Projekt besitzen. Jedes Projekt besitzt initial die Standardgruppen "Administratoren" und "Everyone", die nicht gelöscht werden können. 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 den 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 297). 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 7-93). (Weiterführende Informationen zu internen und externen Gruppen siehe Kapitel 7.4.8.2 Seite 301.)

E-Mail-Verteiler: Ab FirstSpirit Version 4.2R4 gibt es die Möglichkeit, einen E-Mail-Verteiler für Gruppen anzugeben, 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.

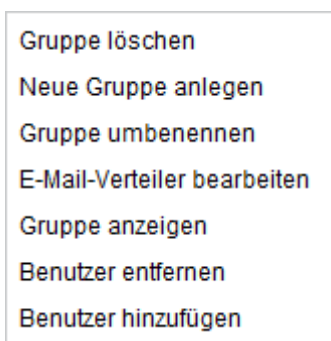


| Gruppenname | Gruppen-ID | Benutzer | E-Mail-Verteiler |
|------------------|------------|----------|--------------------------|
| Administrator... | 373481 | 2 | |
| Everyone | 373482 | 6 | |
| Developer | 373484 | 0 | |
| Editor | 373485 | 2 | |
| Chief Editor | 373486 | 2 | mailinglist@myserver.com |

Abbildung 7-93: 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 7-94: 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 301)
- Neue Gruppe anlegen (siehe Kapitel 7.4.8.2 Seite 301)
- E-Mail-Verteiler bearbeiten (siehe Kapitel 7.4.8.3 Seite 303)
- Gruppe anzeigen (siehe Kapitel 7.4.8.4 Seite 304)
- Benutzer entfernen (siehe Kapitel 7.4.8.5 Seite 304)
- Benutzer hinzufügen (siehe Kapitel 7.4.8.6 Seite 305)



7.4.8.1 Gruppe löschen

Über den Kontextmenüeintrag "Gruppe löschen" können Gruppen, die einem Projekt hinzugefügt wurden, wieder gelöscht 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 "Administratoren" 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 7-95). 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 305).



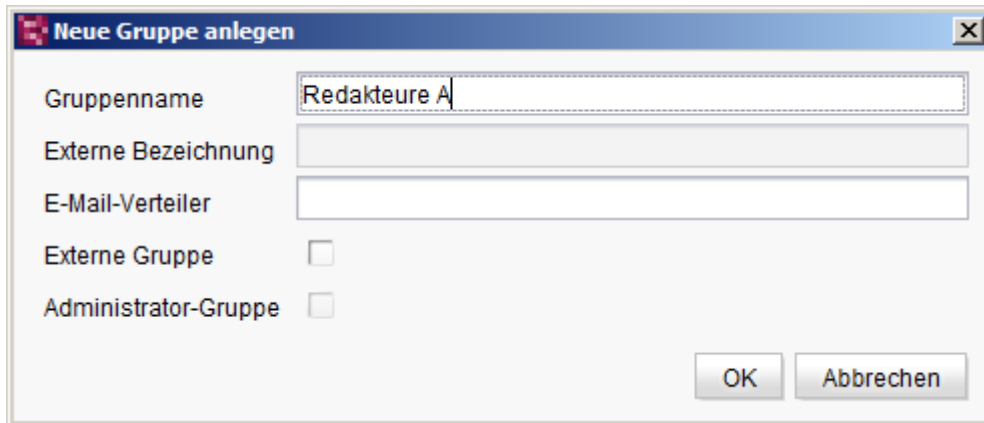


Abbildung 7-95: Neue interne Gruppe anlegen

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 305). 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 beispielsweise 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.

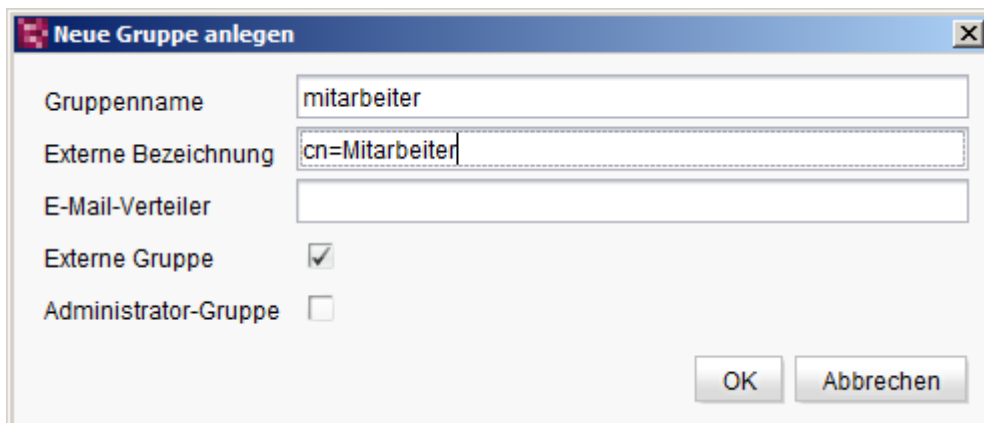


Abbildung 7-96: Neue externe Gruppe anlegen



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



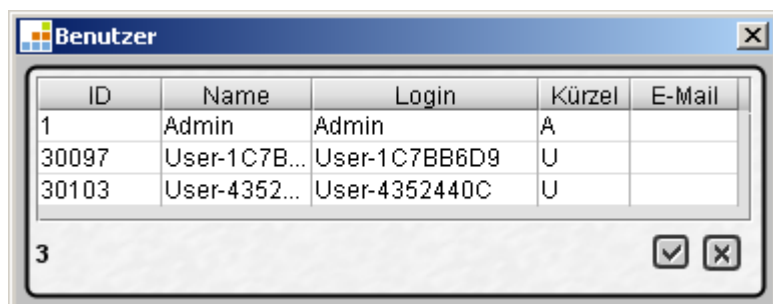
Abbildung 7-97: E-Mail-Verteiler angeben

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



| ID | Name | Login | Kürzel | E-Mail |
|-------|--------------|---------------|--------|--------|
| 1 | Admin | Admin | A | |
| 30097 | User-1C7B... | User-1C7BB6D9 | U | |
| 30103 | User-4352... | User-4352440C | U | |

Abbildung 7-98: 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 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 5.2.3 Seite 298). 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 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 7-99: 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 7.3.10 Seite 248). Eine vollständige Übersicht der Auftragsplanung mit FirstSpirit erfolgt daher im übergeordneten Kapitel 7.5 (Seite 342 ff.).



7.4.10 Auftragsverwaltung

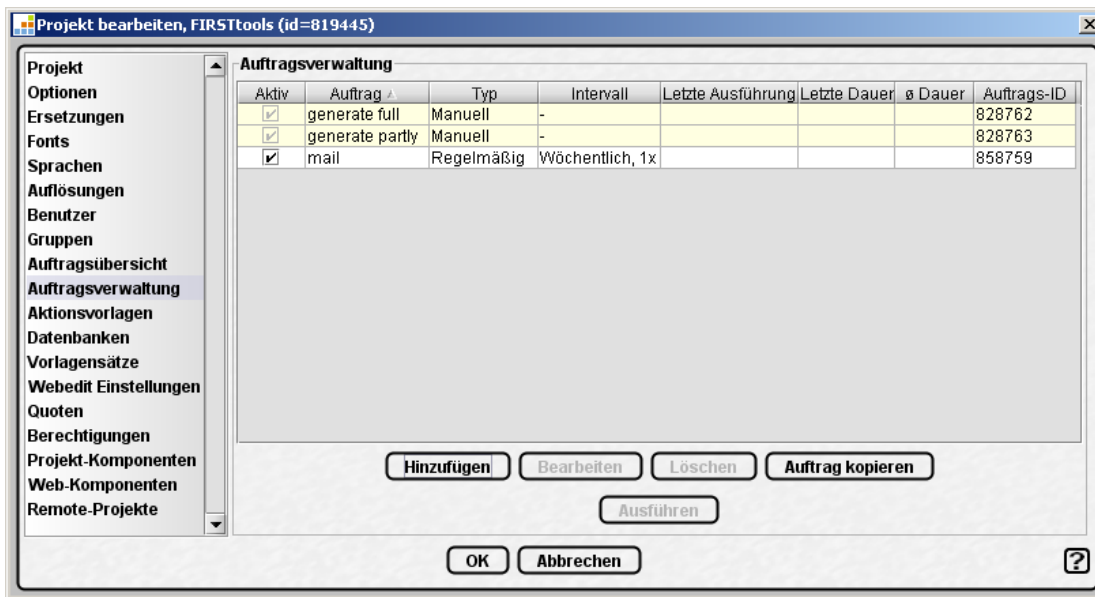


Abbildung 7-100: Projekteigenschaften – Auftragsverwaltung

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 346).

7.4.11 Aktionsvorlagen

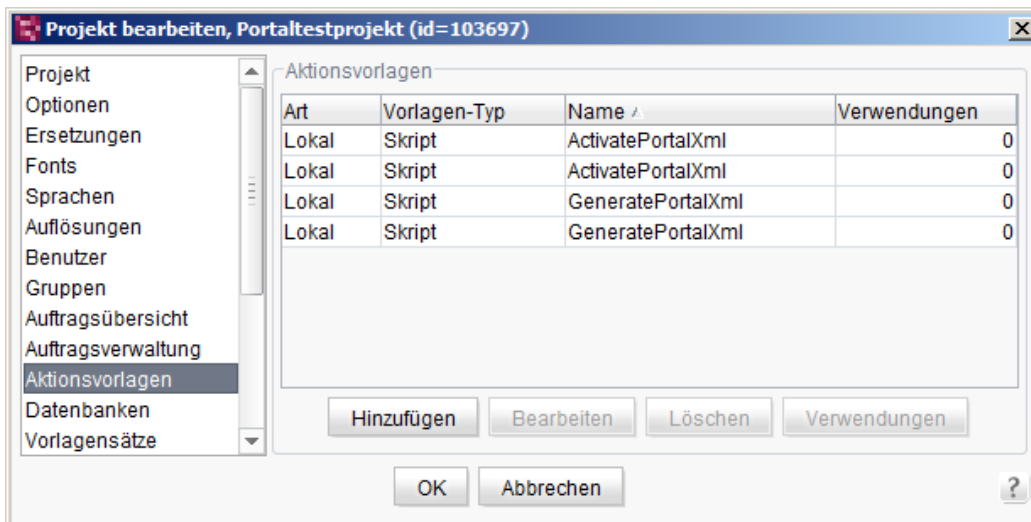
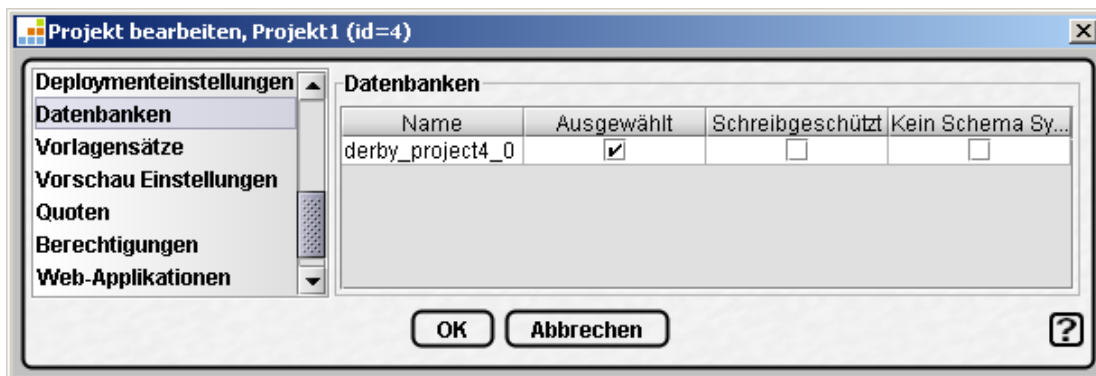


Abbildung 7-101: Projekteigenschaften – Aktionsvorlagen

Siehe Kapitel 7.5.3 Seite 349.



7.4.12 Datenbanken

**Abbildung 7-102: 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 163).



7.4.13 Vorlagensätze

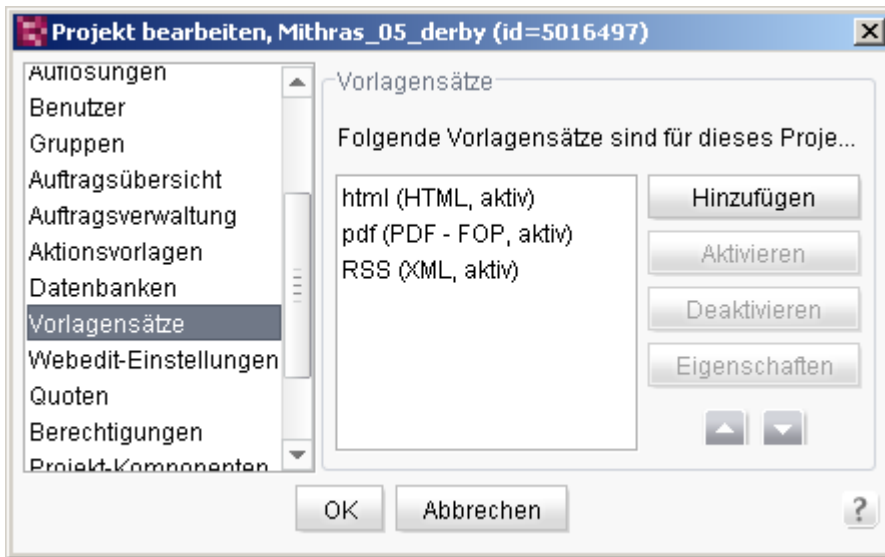



Abbildung 7-103: Projekteigenschaften – Vorlagensätze

Aus den Präsentationskanälen des Servers werden Vorlagensätze definiert, die in dem Projekt zur Verfügung stehen sollen.

 Durch Klick auf diesen Button öffnet sich ein Fenster, in dem die Einstellungen für den neuen Vorlagensatz vorgenommen werden können.

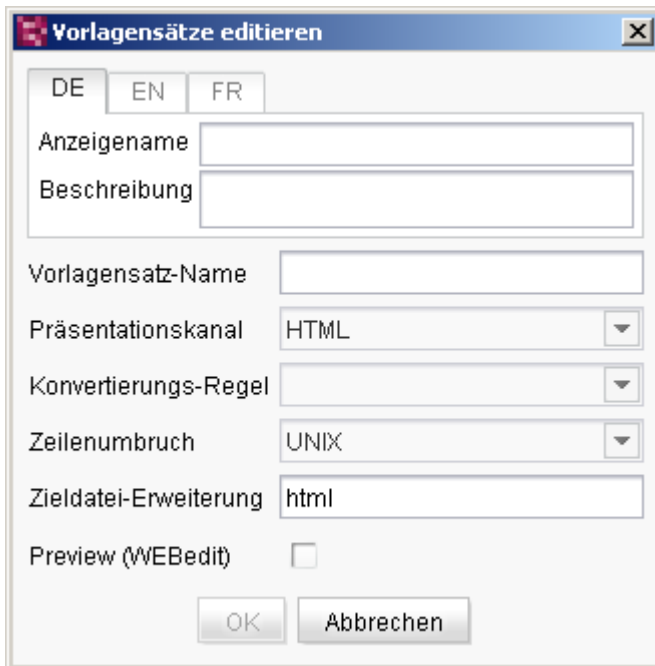


Abbildung 7-104: Neuen Vorlagensatz hinzufügen



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 231)

Konvertierungs-Regel: In diesem Feld kann eine in den Server-Eigenschaften definierte Konvertierungsregel ausgewählt werden. (Siehe Kapitel 7.3.3 Seite 233)

Zeilenumbbruch: Alle zu erzeugenden Zeilenumbüche aus den Vorlagensätzen werden durch Betriebssystem spezifische Zeilenumbüche ersetzt. Für die Standardeinstellung "Windows" wird der Zeilenumbuch 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 /fs4preview/preview. Die Vorschau der einzelner Projekte erfolgt, soweit konfiguriert, auf /fs4preview_{PID}/preview. Wird die Checkbox "Preview (WebEdit)" aktiviert, wird die aktuell angeforderte Vorschau im WebClient angezeigt.



Durch einen Klick auf diesen Button wird der ausgewählte Vorlagensatz aktiviert. (Der aktuelle Status wird in Klammern angezeigt.)



Durch einen Klick auf diesen Button wird der ausgewählte Vorlagensatz deaktiviert. (Der aktuelle Status wird in Klammern angezeigt.)



Durch Klick auf diesen Button öffnet sich ein Fenster, in dem die Einstellungen für den ausgewählten Vorlagensatz bearbeitet werden können. (Siehe Abbildung 7-104: Neuen Vorlagensatz hinzufügen.)

Ab FirstSpirit Version 4.2:

Anzeigename / Beschreibung: Für jeden Vorlagensatz kann ab FirstSpirit Version 4.2 optional 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.

▲ ▼ Ab FirstSpirit Version 4.2 können die Vorlagensätze eines Projekts 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

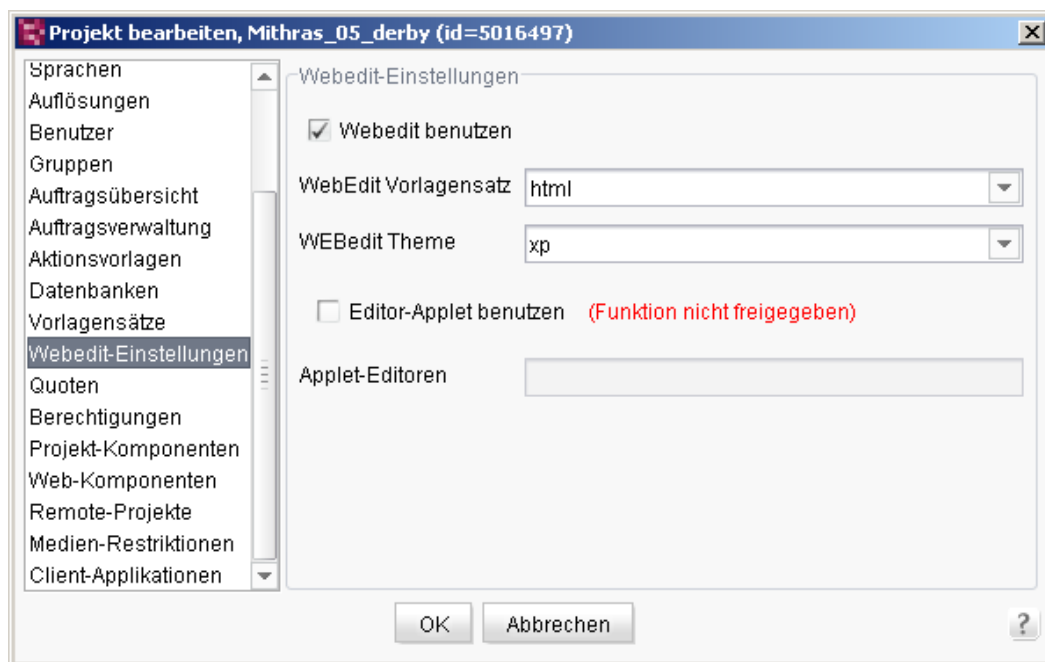


Abbildung 7-105: Projekteigenschaften – WebEdit Einstellungen

WebEdit benutzen: Ist diese Option aktiviert, dann kann dieses Projekt im WebClient (Autorenumgebung) bearbeitet werden. Ist diese Option deaktiviert, kann das Projekt nicht mehr im WebClient bearbeitet werden. Beim Deaktivieren der Option wird folgende Meldung angezeigt: "Sollen die WebEdit Vorlagen aus dem Projekt entfernt werden?" Wurde beim Anlegen des Projektes die Option "WebEdit Formatvorlagen" aktiviert (siehe Abbildung 7-5), wird der Ordner mit den WebEdit-Formatvorlagen mit "Ja" gelöscht. Auch nachträglich in diesem Ordner angelegte Formatvorlagen werden mit gelöscht. Mit "Nein" bleiben die WebEdit-Formatvorlagen bestehen, die WebEdit-Nutzung wird aber unterbunden.

Ab FirstSpirit Version 4.2 wird für den Einsatz von WebEdit neben den WebEdit Formatvorlagen auch auf spezielle Medien zurückgegriffen, und zwar für die Funktionalität Easy-Edit. Diese Medien befinden sich im Ordner "WebClient Medien



(EasyEdit)" in der Medien-Verwaltung, direkt unter dem Wurzelknoten. Wird die Option "Webedit benutzen" deaktiviert, erfolgt bei Projekten, die die Freigabe-Option nutzen, diese Abfrage: "Sollen die WEBedit-Vorlagen aus dem Projekt entfernt werden? ACHTUNG: Das Entfernen der WEBedit-Vorlagen führt dazu, dass der übergeordnete Ordner von "WebClient Medien (EasyEdit)" freigegeben wird." Mit "Ja" werden sowohl die WebEdit-Formatvorlagen als auch die Medien entfernt und der übergeordnete Ordner von "WebClient Medien (EasyEdit)" freigegeben. Dies ist erforderlich, damit bei einer späteren Reaktivierung der WebEdit-Funktion die UUIDs der WebEdit-Medien wiederverwendet werden können. Ohne die Freigabe würde Easy-Edit nach einer Reaktivierung des WebEdit-Modus nicht mehr funktionieren. Mit "Nein" bleiben die WebEdit-Formatvorlagen und die WebEdit-Medien bestehen, die WebEdit-Nutzung wird aber unterbunden.



Beim Aktualisieren von WebEdit-Formatvorlagen ist zu beachten, dass die Zuordnung des WebEdit-Kanals beim Import NICHT berücksichtigt wird.

Heißt der WebEdit-Kanal im konkreten Projekt nicht "html", wird eine Zuordnung anhand des Kanaltyps versucht. Dabei werden die Formatvorlagen in den ersten Kanal vom Typ "html" kopiert (unabhängig vom Namen des Kanals). Ist das nicht der gewünschte WebEdit-Kanal, müssen die Formatvorlagen manuell vom HTML-Kanal in den WebEdit-Kanal kopiert werden.

Daher gilt: Beim Aktualisieren der Formatvorlagen muss immer darauf geachtet werden, dass die neuen Vorlagen auch wirklich in den WebEdit-Kanal importiert werden. Dazu sollte die Versionsnummer der Formatvorlage überprüft werden.

WebEdit Vorlagensatz: In diesem Feld kann ausgewählt werden, welcher Vorlagensatz des Projektes für die Darstellung der WebEdit Seiten verwendet werden soll.

WebEdit Theme: Bis Version 4.2R2 einschließlich kann in diesem Feld ausgewählt werden, welches Theme das Aussehen der WebEdit Menüleisten festlegen soll.





Beim Umschalten der Themes muss sichergestellt werden, dass der Preview-Cache vollständig gelöscht wird. Dazu muss die Preview-Webanwendung ("fs4preview") neu gestartet werden (siehe Kapitel 7.3.14 Seite 251).

Das Erstellen **neuer** "WebEdit Themes" wird **ab FirstSpirit Version 4.0** nicht mehr unterstützt. Die Theming-Funktionalität von WebEdit wird mit dem Release der WebEdit Version 5.0 entfallen. Eine Unterstützung für das Erstellen neuer Themes für WebEdit ist daher nicht mehr vorgesehen.

Die FirstSpirit Standard-Themes ("default", "xp") und das spezifische SAP-Theme ("sap") können **in FirstSpirit Version 4.1** jedoch weiterhin eingesetzt werden. **Ab FirstSpirit Version 4.2** ist die Theming-Funktionalität für WebEdit nur noch eingeschränkt nutzbar. Unter anderem wurde die Symbolleiste im XP-Theme vollständig überarbeitet und ist damit im XP-Theme nicht mehr enthalten. Für alle anderen Themes gilt: Alle css-Dateien aus dem Theme-Projekt wurden manuell überarbeitet und sind nicht mehr im Theme-Projekt enthalten. Das gleiche gilt für neue (und geänderte) Eingabekomponenten in WebEdit 4.2 (unter anderem CMS_INPUT_OBJECTCHOOSER, CMS_INPUT_REF). Diese Komponenten sind im Theme-Projekt nicht mehr enthalten.

Editor-Applet benutzen: Die Einstellung "Editor-Applet benutzen" ist nur möglich, wenn auch die Option "WebEdit benutzen" für ein Projekt aktiviert ist. Ist "Editor-Applet benutzen" aktiviert, dann können einige Eingabekomponenten in WebEdit als Java-Applet (statt eines JSP-Editors) verwendet werden.

Die Entwicklung in diesem Bereich ist noch nicht vollständig abgeschlossen. Die Funktionalität ist daher noch nicht freigegeben.

Applet-Editoren: In diesem Feld werden die Eingabekomponenten aufgelistet, die als Applet gerendert werden sollen, beispielsweise die Eingabekomponente CMS_INPUT_TEXT. Dabei ist eine kommaseparierte Eingabe mehrerer Komponenten möglich. (Die Änderung wirkt sich auf alle Eingabekomponenten dieses Typs aus.)

Die Entwicklung in diesem Bereich ist noch nicht vollständig abgeschlossen. Die Funktionalität ist daher noch nicht freigegeben.



7.4.15 Quoten

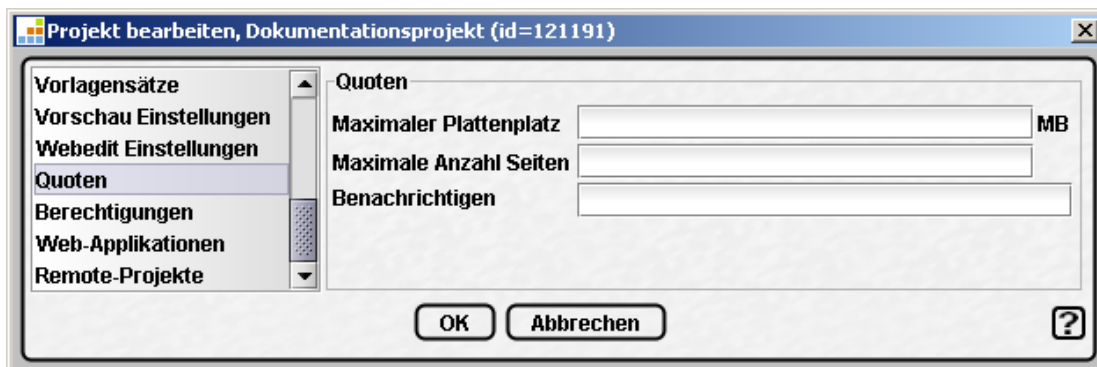


Abbildung 7-106: Projekteigenschaften – Quoten

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

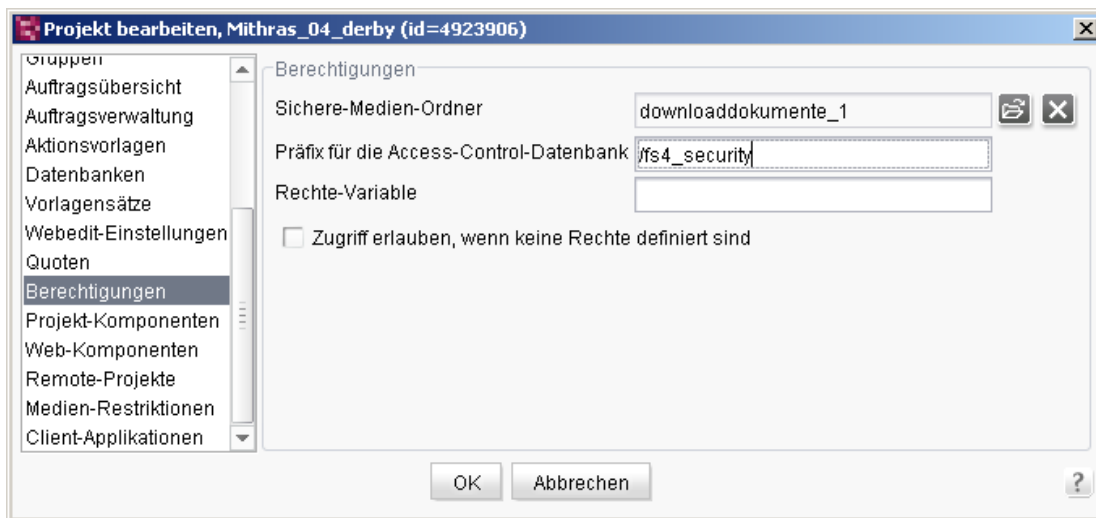





Abbildung 7-107: Projekteigenschaften – Berechtigungen

Sichere Medien Ordner: Über das 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 Icon  kann eine bereits getroffene Auswahl wieder entfernt werden (vgl. Kapitel 11.3 Seite 502).



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 möglich²⁸ (vgl. Kapitel 11.3 Seite 502).

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:

- URL der Webanwendung (z. B. http://meinServer.de)
- Präfix für die Access-Control-Datenbank (/fs4_security)
- Pfad zu einer Datei (/de/index.html)

²⁸ Siehe Dokumentation: SECU40DE_FirstSpirit_Modules_Security.pdf



Der vollständige Pfad innerhalb der ACL-Datenbank setzt sich aus dem Präfix und dem Pfad zur Datei zusammen, z. B. `/fs4_security/de/index.html`

In Abbildung 7-107 lautet das Präfix `/fs4_security` und entspricht dem direkten Unterverzeichnis `fs4_security` der Webanwendung, beispielsweise `live`:
`"~Webserver/webapps/live/fs4_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 7.5.9.3.3 Seite 378).

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.

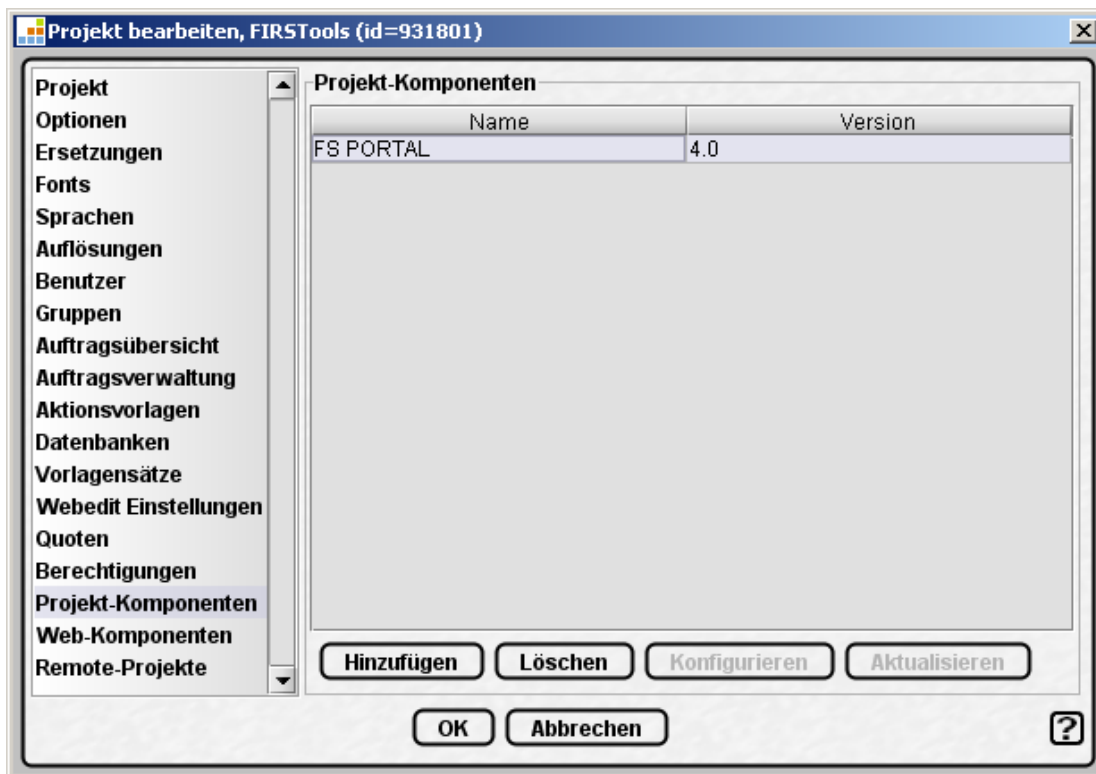
Zugriff erlauben, wenn keine Rechte definiert sind: Ist die Checkbox aktiviert, wird der Zugriff auf `"Sichere Medien"` für alle Gruppenmitglieder (aus der Datei `groups.xml`) erlaubt, wenn keine anderslautende Definition über die Rechte-Variable definiert wurde. Ist die Checkbox deaktiviert, werden nur die über die Rechte-Variable gesetzten Werte ausgewertet.

Weitere Informationen zu Benutzerrechten siehe Kapitel 11 Seite 491.

Weitere Informationen zum Konzept "Sichere Medien" siehe Kapitel 11.3 Seite 502.



7.4.17 Projekt-Komponenten

**Abbildung 7-108: 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 7.3.14 Seite 251). Ein Beispiel für einen Projekt-Komponente ist die FirstSpirit-Portal-Komponente, die Bestandteil des Moduls "FS Portal" ist (siehe Dokumentation FirstSpirit Portal).

Hinzufügen

Mit einem Klick auf den Button 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 7-109: Hinzufügen einer Projekt-Komponente

Löschen

Mit einem Klick auf den 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.

Konfigurieren

Mit einem Klick auf den Button kann die Konfiguration für eine zuvor hinzugefügte Projekt-Komponente bearbeitet werden (vgl. Kapitel 7.3.14 Seite 251). Die Konfiguration erfolgt abhängig von der Komponente entweder über eine von der Komponente erzeugte oder eine generische GUI.

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, 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 7-110: Projekteigenschaften – Web-Komponenten

In diesem Bereich können die Web-Komponenten für ein Projekt aktiviert werden.

Web-Bereiche:

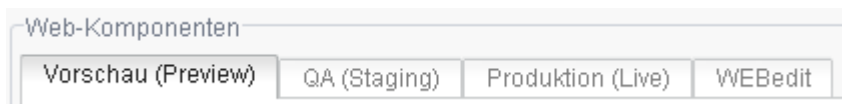


Abbildung 7-111: Web-Bereiche innerhalb eines Projekts

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 171) (ab FirstSpirit Version 4.1).

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 7.3.15 Seite 258). 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 321)
- Generischer Webserver (siehe Kapitel 7.4.18.2 Seite 321)
- Externer Webserver (siehe Kapitel 7.4.18.3 Seite 322)

Installieren

Mit einem Klick auf den Button 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 Web-Anwendung bereits installiert, anschließend aber die Konfiguration geändert (siehe Kapitel 7.4.18.6) oder eine Komponente hinzugefügt (siehe Kapitel 7.4.18.4) oder gelöscht (siehe Kapitel 7.4.18.5), wird statt des Buttons "Installieren", der **Button "Aktualisieren"** angezeigt.

Deinstallieren

Mit einem Klick auf den Button 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.

Download

Mit einem Klick auf den Button kann eine WAR-Datei der Anwendung heruntergeladen werden, die manuell auf dem Webserver installiert werden muss (siehe Kapitel 7.3.16.5 Seite 268). Der Button wird nur für die Konfiguration externer Webserver oder die Konfiguration generischer Webserver (ohne die erforderliche Skript-Funktionalität) angezeigt.

Aktivieren




Mit einem Klick auf den Button 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 (z. B. Tomcat) 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  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  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  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 7.3.15.1 Seite 260). Die Steuerung für den generischen Webserver ist nicht standardmäßig vorhanden, kann aber über Skripte realisiert werden (siehe Kapitel 7.3.15.2 Seite 260). 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 (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  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  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  bestätigt und gespeichert werden.

7.4.18.3 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 7.3.15.3 Seite 263). Die Steuerung für einen externen Webserver wird über FirstSpirit nicht unterstützt und muss manuell erfolgen (siehe Kapitel 7.3.15.4 Seite 263). 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 Webserver oder automatisch aus dem Dateisystem des Webserver heraus
4. Nach der Installation kann mit einem Klick auf , 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  bestätigt und gespeichert werden.



7.4.18.4 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 7.3.14 Seite 251).



Abbildung 7-112: 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").

7.4.18.5 Löschen einer Web-Komponente

 Mit einem Klick auf den Button 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").



7.4.18.6 Konfigurieren einer Web-Komponente

 Mit einem Klick auf den Button kann die Konfiguration für eine zuvor hinzugefügte Komponente bearbeitet werden (vgl. Kapitel 7.3.14 Seite 251). 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:

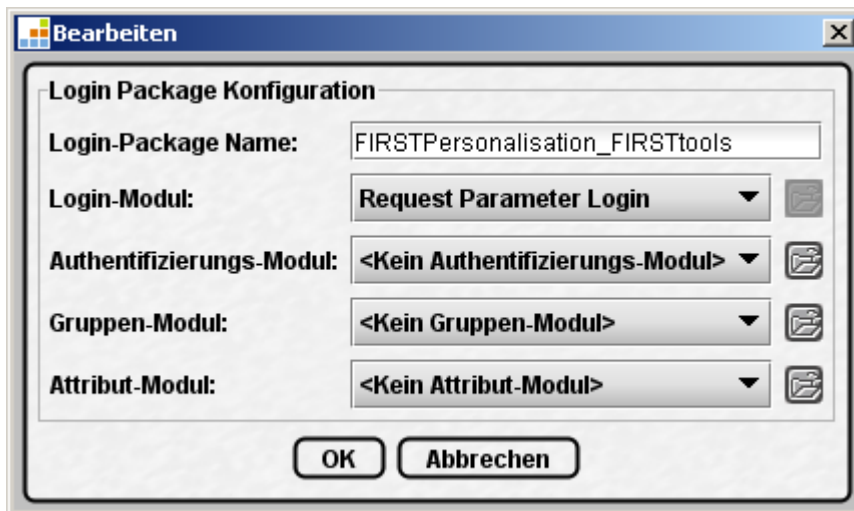


Abbildung 7-113: Konfiguration einer weblokalen Komponente (Beispiel)

7.4.18.7 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.

7.4.18.8 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.

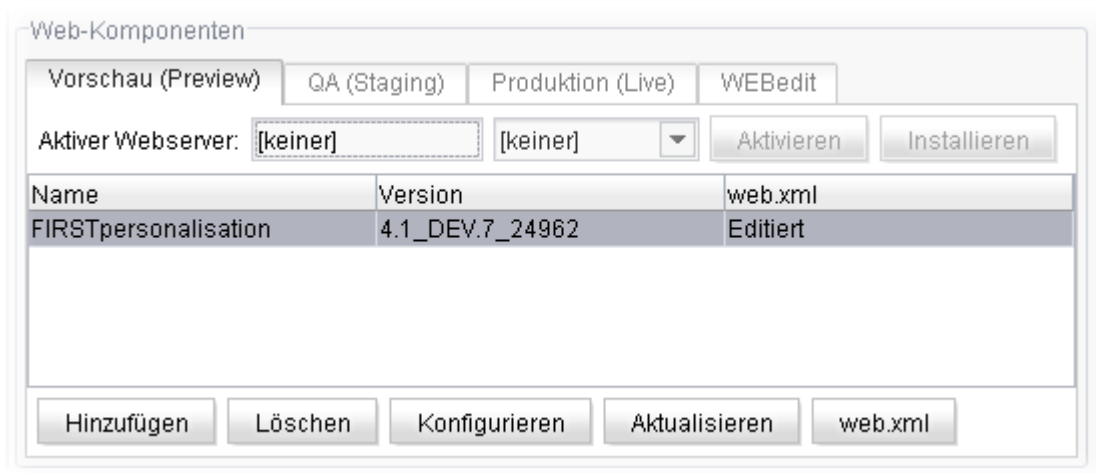


Abbildung 7-114: Nach manuellem Editieren der web.xml

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 7-115: 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.





Remote Projekt Konfiguration bearbeiten

Symbolischer Name:

Projekt:  

Aktueller Benutzer

Benutzer:  

Passwort:

Prefix:

Remote-Kategorie:

Remote-Typen:

- Remote-Media
- Verwandte Projekte
- Remote-Schemata

OK Abbrechen

Abbildung 7-116: 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. Die Bezeichnung ist hinter dem Doppelpunkt nach dem Schlüsselbegriff "category" anzugeben, z. B. "category:meineKategorie". 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 "category:default" (Standardkategorie) vorbelegt.

Remote-Typen: Es sind drei unterschiedliche Arten des Remote-Zugriffs möglich:

- Remote-Media (siehe Kapitel 7.4.19.1 Seite 327)
- Verwandte Projekte (siehe Kapitel 7.4.19.2 Seite 328)
- Remote-Schemata (siehe Kapitel 7.4.19.3 Seite 328)



Bei allen Arten von Remote-Zugriff 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 Paket-Verwaltung 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 (für weitere Informationen zur Paket-Verwaltung siehe Dokumentation zur Funktionalität "FirstSpirit Paket-Verwaltung").

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 Remote-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 Integration" benötigt, das zur Anbindung unterschiedlicher Datenbank-Technologien an FirstSpirit dient. Mithilfe der FirstSpirit Integrationsplattform können Inhalte aus einer Datenbank über eine Webanwendung dargestellt und editiert werden. FirstSpirit Integration 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 Integration siehe die entsprechende Modul-Dokumentation.



7.4.20 Medien-Restriktionen (ab V4.1)



Diese Funktionalität ist erst ab FirstSpirit-Version 4.1 freigegeben. Daher werden Screenshots im neuen Look & Feel "LightGray" dargestellt. Im Look & Feel "Classic" kann die Darstellung geringfügig abweichen.



Abbildung 7-117: 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 (ab V4.1)

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 (ab V4.1)

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 (ab V4.1)

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.

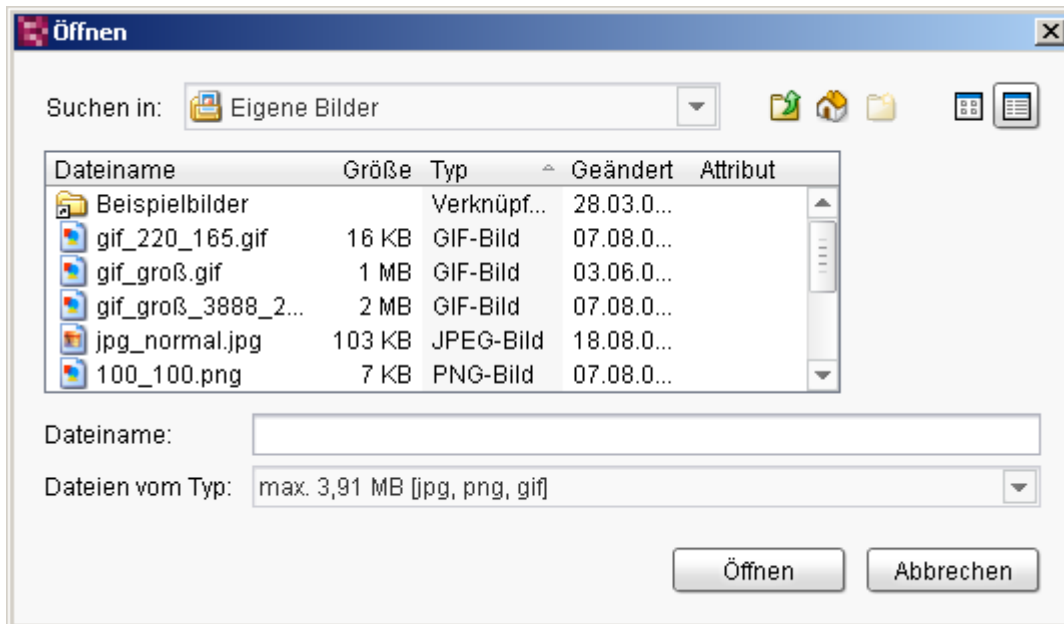


Abbildung 7-118: Aktivierte Beschränkung auf Dateigrößen und Dateinamen-Erweiterungen

Rechts neben "Dateien vom Typ:" werden die Restriktionen noch einmal angezeigt. Im Beispiel in Abbildung 7-118 dürfen Dateien mit den Dateinamen-Erweiterungen "jpg", "png" und "gif" bis zu einer Größe von 3,91 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 (ab V4.1)

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.

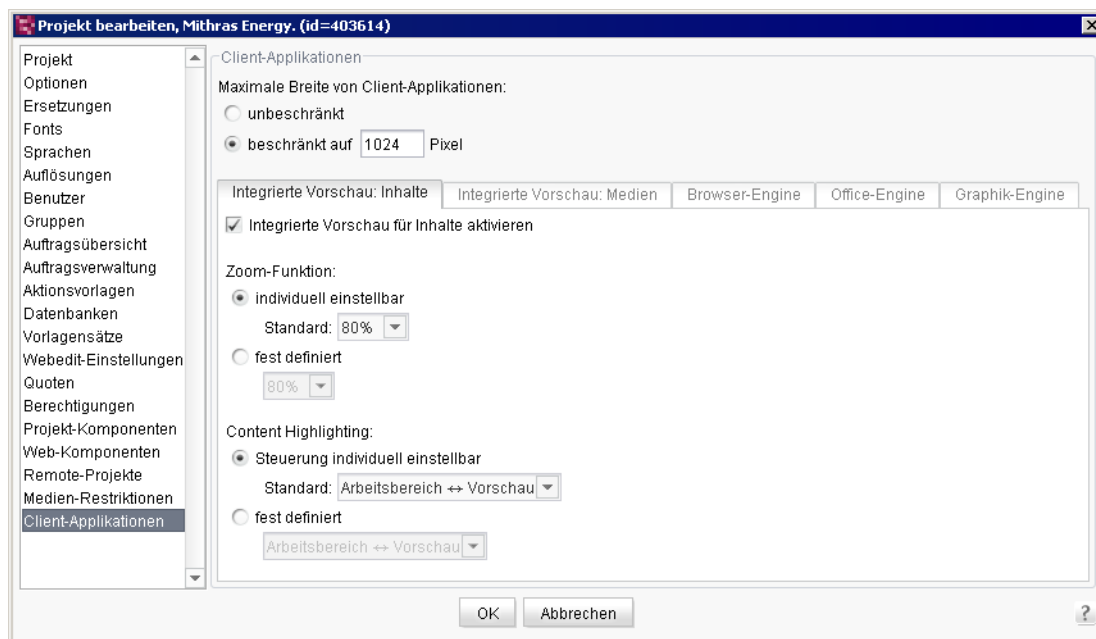


Abbildung 7-119: Projekteigenschaften – Client-Applikationen

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 |
|--|------------------------------|----------------|---------------|----------------|
| <input checked="" type="checkbox"/> Integrierte Vorschau für Medien aktivieren | | | | |
| Office-Engine verwenden für folgende Datei-Endungen (kommasepariert): | | | | |
| <input type="text" value="odt,ott,sxw,doc,odg,otg,sxd,ods,ots,sxc,cls,xls,odf,sxm,odp,otp,sxi,ppt,odm,oth,odb"/> | | | | |
| Browser-Engine verwenden für folgende Datei-Endungen (kommasepariert): | | | | |
| <input type="text" value="pdf,html,htm,url,swf"/> | | | | |
| Integrierten Text-Editor verwenden für folgende Datei-Endungen (kommasepariert): | | | | |
| <input type="text" value="css,js,txt,xml,csv,json,as"/> | | | | |
| Integrierte Bild-Ansicht verwenden für folgende Datei-Endungen (kommasepariert): | | | | |
| <input type="text" value="png,jpg,jpeg,bmp,gif,psd"/> | | | | |
| Microsoft Media Player (nur Windows) verwenden für folgende Datei-Endungen (kommasepariert): | | | | |
| <input type="text" value="avi,mpg,mpeg,wmv,asf,mp3,mp4"/> | | | | |
| <input type="button" value="Default-Werte wiederherstellen"/> | | | | |

Abbildung 7-120: 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, die mit einer FirstSpirit-Server-Version ab 4.2.400 erstellt werden, ist die Option standardmäßig aktiviert, so dass mittels der folgenden Felder eine Zuordnung vorgenommen werden kann, welche Dateiformate von welchen Anwendungen in der Integrierten Vorschau dargestellt werden sollen. Für Projekte, die mit einer FirstSpirit-Server-Version unterhalb von 4.2.400 erstellt wurden, ist die Option standardmäßig deaktiviert. Ab Version 4.2.438.44956 gehören auch die Microsoft Office 2010-Formate "docx", "pptx" und "xlsx" zu den in Abbildung 7-120 gezeigten Default-Werten.



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

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

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

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

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

Default-Werte wiederherstellen: Mit einem Klick auf diese Schaltfläche kann die Standard-Einstellung wiederhergestellt werden.



7.4.21.3 Browser-Engine



The screenshot shows a settings window with five tabs: 'Integrierte Vorschau: Inhalte', 'Integrierte Vorschau: Medien', 'Browser-Engine', 'Office-Engine', and 'Graphik-Engine'. The 'Browser-Engine' tab is active. It contains a checked checkbox 'Browser-Engine aktivieren'. Below it are two radio buttons: 'individuell einstellbar' (selected) and 'fest definiert'. Under 'individuell einstellbar', there is a 'Standard:' label followed by a dropdown menu showing 'Mozilla Firefox'. Under 'fest definiert', there is another dropdown menu also showing 'Mozilla Firefox'.

Abbildung 7-121: 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 334) 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 Release Notes Version 4.2R4*, Kapitel 3.1.3 "Web-Browser-Integration".



7.4.21.4 Office-Engine

The screenshot shows a configuration window with several tabs: 'Integrierte Vorschau: Inhalte', 'Integrierte Vorschau: Medien', 'Browser-Engine', 'Office-Engine', and 'Graphik-Engine'. The 'Office-Engine' tab is active. It contains a checked checkbox labeled 'Office-Engine aktivieren'. Below this, there are two radio button options: 'individuell einstellbar' (selected) and 'fest definiert'. Under 'individuell einstellbar', there is a dropdown menu labeled 'Standard:' with the value 'Microsoft Office (nur Windows)'. Under 'fest definiert', there is another dropdown menu with the same value 'Microsoft Office (nur Windows)'.

Abbildung 7-122: 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 334) 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 334) 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.




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. (Siehe auch FirstSpirit Release Notes Version 4.2R4, Kapitel 3 "Das FirstSpirit AppCenter").

Weitere Hinweise zur Verwendung von Microsoft Office siehe *FirstSpirit Release Notes Version 4.2R4*, Kapitel 3.1.4 "Microsoft Office-Integration", zur Verwendung von OpenOffice und Google Docs siehe *FirstSpirit Release Notes Version 4.2R4*, Kapitel 3.3 "Applikationsintegrationen im BETA-Stadium".

7.4.21.5 Graphik-Engine

Abbildung 7-123: 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 333) 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. (Siehe auch FirstSpirit Release Notes Version 4.2R4, Kapitel 3 "Das FirstSpirit AppCenter").



*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 **Picnik** oder **Pixlr** ist eine aktive Verbindung zum Internet erforderlich. Die zu bearbeitenden Bilder werden faktisch auf Server des jeweiligen Anbieters hochgeladen und dort bearbeitet. Dies sollte bei der Verwendung im Hinblick auf den Datenschutz berücksichtigt werden*

Zu Systemvoraussetzungen und Einschränkungen hinsichtlich der Grafik Engine siehe auch *FirstSpirit Release Notes Version 4.2R4*, Kapitel 3.2 "Java-



Applikationsintegration in 4.2R4 (Java Image Editor)".



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 der 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).

Innerhalb der Projekteigenschaften können nur folgende projektbezogene Aktionen ausgewählt werden:

- Generierung ausführen: siehe Kapitel 7.5.9.2 Seite 368
- Veröffentlichung ausführen: siehe Kapitel 7.5.9.3 Seite 372
- Projektsicherung durchführen: siehe Kapitel 7.5.9.5 Seite 384
- Alte Projektstände archivieren: siehe Kapitel 7.5.9.1 Seite 363

Lediglich die beiden Aktionstypen

- Skript ausführen: siehe Kapitel 7.5.9.4 Seite 381
- Mail verschicken: siehe Kapitel 7.5.9.6 Seite 385

können sowohl in server- als auch projektbezogenen Aufträgen genutzt werden.

Einmal erstellte Aktionen können einfach aus bereits bestehenden Aufträgen kopiert werden (siehe Kapitel 7.5.7). Sollen gleiche Aktionen in mehreren Aufträgen wieder verwendet werden, kann analog zum Vorlagenkonzept von FirstSpirit eine "Aktionsvorlage" erstellt werden (siehe Kapitel 7.5.3). Über die Aktionsvorlage kann



eine Aktion beliebig vielen Aufträgen hinzugefügt werden (siehe Kapitel 7.5.8). Änderungen an einer Aktion werden in diesem Fall zentral über die Vorlage gepflegt.

Neben der Ausführung über die Server- und Projektkonfiguration, können Aufträge auch interaktiv über den FirstSpirit-JavaClient ausgeführt werden. Dazu muss der Benutzer die erforderlichen Rechte für die interaktive Ausführung besitzen (siehe Kapitel 7.5.4 Seite 351), also in der Liste der berechtigten Benutzer aufgeführt oder einer Benutzergruppe aus der Liste der berechtigten Gruppen angehören.

7.5.1 Auftragsübersicht

| Startzeitpunkt | Typ | Auftrag | Status | Letzte Dauer | g Dauer | Auftrags-ID |
|------------------|---------|---------------|-------------|--------------|-------------|-------------|
| 07.07.2009 09:03 | Manuell | generate full | Erfolgreich | 0:02:09.176 | 0:00:56.126 | 82081 |
| 03.07.2009 15:36 | Manuell | generate full | Erfolgreich | 0:01:35.503 | 0:00:56.126 | 82081 |
| 01.07.2009 10:57 | Manuell | generate full | Erfolgreich | 0:01:12.929 | 0:00:56.126 | 82081 |
| 01.07.2009 10:25 | Manuell | generate full | Erfolgreich | 0:01:50.555 | 0:00:56.126 | 82081 |

Abbildung 7-124: 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 344). Je nach dem, ob man sich in den Servereigenschaften (siehe Kapitel 7.3.10) 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 343).

Durch einen Doppelklick auf den gewünschten Auftrag wird der Dialog mit den Auftrags-Details geöffnet (siehe Kapitel 7.5.1.3 Seite 345).


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 Uhrensymbol  ausgewählt werden.

 : Wurde eine neue Option bzw. ein anderes Anfangs- oder Enddatum ausgewählt, muss die Auswahl per Klick auf den Button 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 351).

- 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

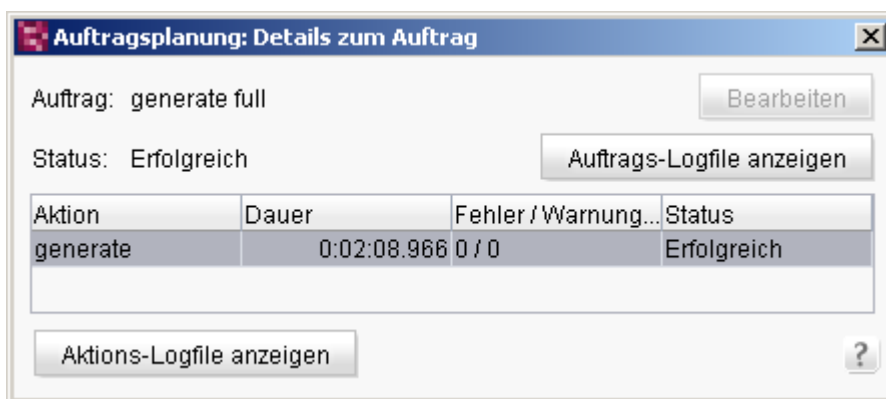


Abbildung 7-125: Details zum Auftrag

Durch einen Doppelklick auf einen Auftrag aus der Auftragsübersicht (siehe Abbildung 7-124) 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 344).

Bearbeiten

Dieser Button 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 351).

Auftrags-Logfile anzeigen

Durch einen Klick auf diesen Button 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 344).

Aktions-Logfile anzeigen

Durch Klick auf diesen Button 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 7-126: Auftragsverwaltung

Die Auftragsverwaltung enthält alle Aufträge, die (je nachdem, ob man sich in den Server- oder Projekteigenschaften befindet) entweder server- (siehe Kapitel 7.3.10) oder projektabhängig (siehe Kapitel 7.4.9) angelegt wurden.

Aktiv: Ist diese Option *deaktiviert*, wird die automatische Ausführung dieses Auftrags ausgesetzt. Ab FirstSpirit Version 4.2R4 gilt dies 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 351).

- 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.

Hinzufügen

Durch Klick auf diesen Button wird ein neuer Auftrag für dieses Projekt bzw. den Server angelegt (siehe Kapitel 7.5.4 Seite 351).

Bearbeiten

Durch Klick auf diesen Button (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 351).

Löschen

Durch Klick auf diesen Button 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.

Ausführen

Durch einen Klick auf den Button kann der gewünschte Auftrag, unabhängig vom eingestellten Ausführungsintervall, direkt manuell ausgeführt werden.

Auftrag kopieren

Durch Klick auf diesen Button 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.

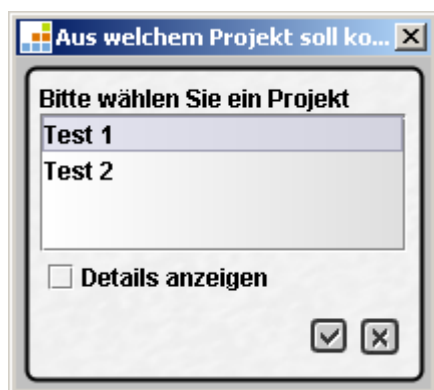


Abbildung 7-127: Auftrag kopieren – Auswahl eines Projekts

Im nächsten Schritt öffnet sich ein neuer Dialog, in dem alle Aufträge des ausgewählten Projekts aufgelistet werden.



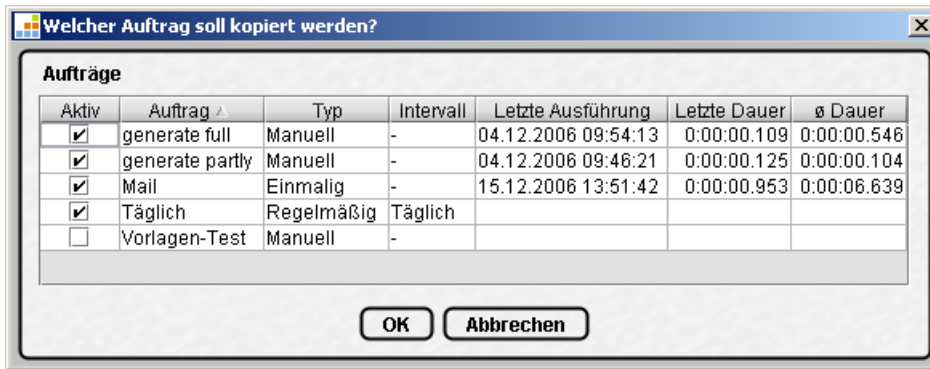


Abbildung 7-128: Auftrag kopieren – Auswahl des zu kopierenden Auftrags

Hier lässt sich nun ein einzelner Auftrag per Doppelklick in den bearbeiteten Auftrag kopieren.

7.5.3 Aktionsvorlagen

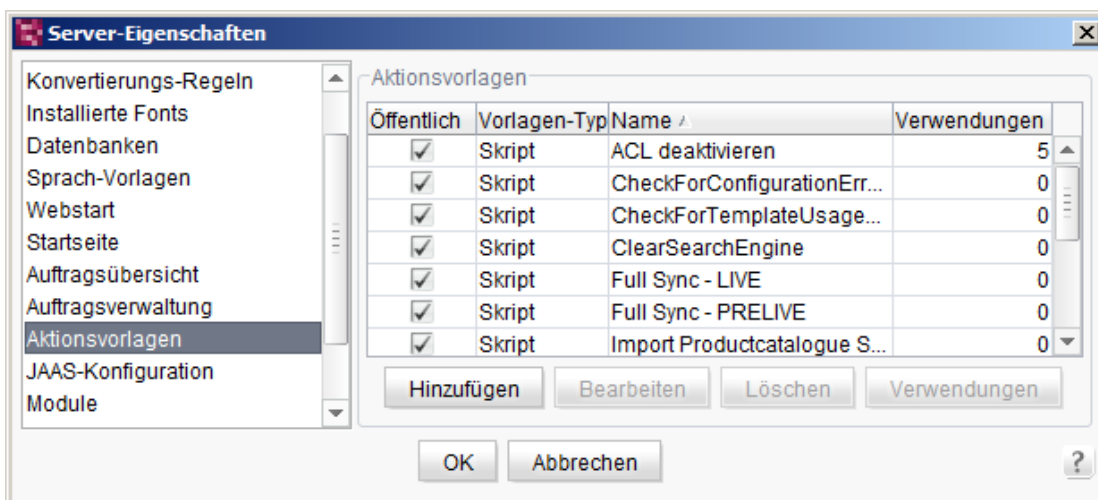


Abbildung 7-129: Aktionsvorlagen

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 7.5.9.3 Seite 372
- Skript ausführen: siehe Kapitel 7.5.9.4 Seite 381
- Mail verschicken: siehe Kapitel 7.5.9.6 Seite 385

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.

 Hinzufügen

Durch Klick auf diesen Button wird eine neue Aktionsvorlage erstellt. Dafür muss zunächst der Aktions-Typ ausgewählt werden (siehe Kapitel 7.5.6 Seite 358).

 Bearbeiten

Durch Klick auf diesen Button kann die selektierte Aktionsvorlage bearbeitet werden. Es öffnet sich ein neuer Dialog mit der entsprechenden Eingabemaske (siehe Kapitel 7.5.9 Seite 363 und Kapitel 7.5.10 Seite 386).



Änderungen, die an einer Aktionsvorlage vorgenommen werden, wirken sich auf alle Aufträge aus, die diese Vorlage verwenden.

 Löschen

Durch Klick auf diesen Button 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.

 Verwendungen

Durch Klick auf diesen Button öffnet sich ein neuer Dialog, der alle Aufträge anzeigt, in denen diese Aktionsvorlage verwendet wird:



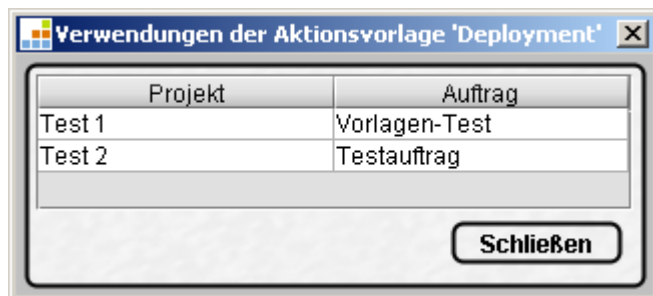


Abbildung 7-130: Anzeige der Verwendungen einer selektierten Aktionsvorlage

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.



Auftragsplanung: Auftrag bearbeiten

Eigenschaften | **Aktionen**

Auftragsname:

eMail-Verteiler:

Manuell

Einmalig Ausführung am um

Regelmäßig

Täglich Erste Ausführung am um

Wöchentlich Ausführungsregel:

Monatlich

Intervall Ausführung alle Minuten

Interaktive Ausführung

Interaktive Ausführung erlaubt für:

Benutzer | **Gruppen**

| ID | Name | Login ^ | Kürzel | E-Mail |
|-------|------|---------|--------|--------|
| 47057 | | test1 | | |
| 47059 | | test3 | | |
| 47060 | | test4 | | |
| 47061 | | test5 | | |
| 47065 | | test9 | | |

Parallele Ausführung:

Abbildung 7-131: Auftrag bearbeiten – Eigenschaften

Auftragsname: Eindeutiger Name des Auftrags. Unter diesem Namen wird der Auftrag sowohl in der Server- und Projektkonfiguration (siehe Abbildung 7-124 und Abbildung 7-126) 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 7.5.9.6 Seite 385) vererbt und von diesen genutzt werden können.

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 Uhrensymbol  definiert werden.
- **Regelmäßig:** Ist dieser Ausführungstyp ausgewählt, wird der Auftrag automatisch in regelmäßigen Abständen ausgeführt. Dazu müssen per Klick auf das Uhrensymbol  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).



Ausführungsregel

Montag Dienstag Mittwoch Donnerstag

Freitag Samstag Sonntag

Abbildung 7-132: 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 7-133: Ausführungsregel – Monatliche Ausführung

- **Intervall**
Bei Auswahl des Radiobuttons "Intervall" muss lediglich das Intervall zwischen den Ausführungen in Minuten angegeben werden (Standardwert: 60 Min).

Abbildung 7-134: Ausführungsregel – Intervall-Ausführung

Abhängig vom selektierten Ausführungs-Typ können weitere Auftragseigenschaften editiert werden (siehe Abbildung 7-131):

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 Auftrag ist nur aus dem JavaClient heraus möglich.

Benutzer

Diese Tabelle listet alle Benutzer auf, welche berechtigt sind diesen Auftrag interaktiv (aus dem JavaClient heraus) auszuführen.

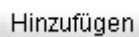
Hinzufügen Durch einen Klick auf diesen Button öffnet sich ein Dialog, der alle Benutzer des aktuellen Projekts anzeigt. Bereits ausgewählte Benutzer werden durch Fettschrift hervorgehoben.

Löschen Durch einen Klick auf diesen 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.



 Hinzufügen

Durch einen Klick auf diesen Button öffnet sich ein Dialog, der alle Benutzergruppen des aktuellen Projekts anzeigt. Bereits ausgewählte Gruppen werden durch Fettschrift hervorgehoben.

 Löschen

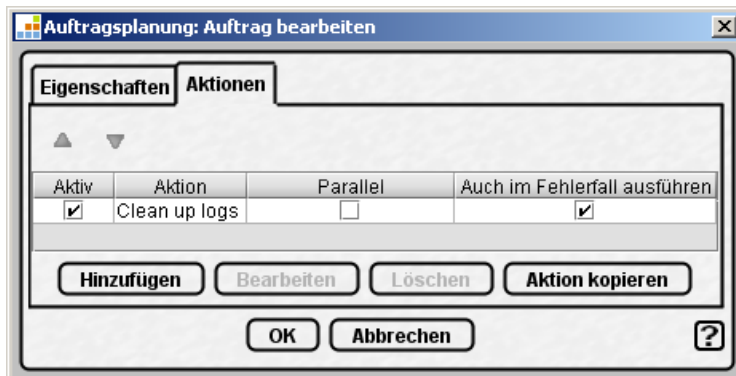
Durch einen Klick auf diesen Button werden alle selektierten Benutzergruppen aus der Tabelle gelöscht.

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 7-135: 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 "Auftrag bearbeiten". 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 eine Position nach 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 Fehlerfall ausführen |
|-------------------------------------|---------------|-------------------------------------|-------------------------------------|
| <input checked="" type="checkbox"/> | Clean up logs | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| <input checked="" type="checkbox"/> | Mail | <input 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. Schlägt also beispielsweise die Aktion "Clean up logs" fehl, wird die Aktion "Mail" dennoch ausgeführt.

Hinzufügen

Durch einen Klick auf diesen Button wird eine neue Aktion für diesen Auftrag angelegt (siehe Kapitel 7.5.6 Seite 358).

Bearbeiten

Durch einen Klick auf diesen Button (oder einen Doppelklick auf den Tabelleneintrag), öffnet sich ein neuer Dialog zum Bearbeiten der selektierten Aktion (siehe Kapitel 7.5.9 Seite 363 und Kapitel 7.5.10 Seite 386).

Löschen

Durch einen Klick auf diesen Button wird die selektierte Aktion aus diesem Auftrag gelöscht. Das Löschen muss durch eine Sicherheitsabfrage bestätigt werden.

Aktion kopieren

Durch einen Klick auf diesen Button besteht die Möglichkeit eine Aktion aus einem anderen Auftrag heraus zu kopieren und dem aktuellen Auftrag hinzuzufügen (siehe Kapitel 7.5.7 Seite 359).

Neu aus Vorlage

Durch einen Klick auf diesen Button kann eine Aktion, die zuvor in den Aktionsvorlagen definiert wurde (siehe Kapitel 7.5.3 Seite 349), zum aktuellen Auftrag hinzugefügt werden (siehe Kapitel 7.5.8 Seite 360).





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 349).

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

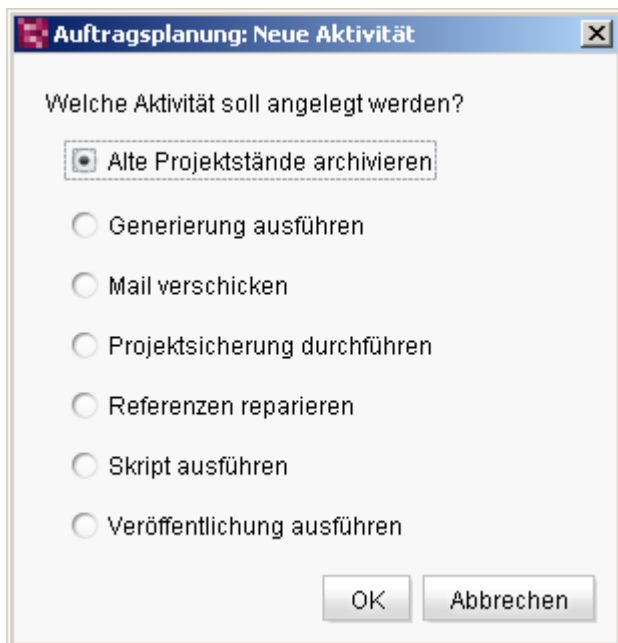


Abbildung 7-136: Anlegen einer projektbezogenen Aktion

Abhängig von der selektierten Aktion wird die Eingabemaske des ausgewählten Aktionstyps geöffnet:



- Alte Projektstände archivieren siehe Kapitel 7.5.9.1 Seite 363
- Generierung ausführen: siehe Kapitel 7.5.9.2 Seite 368
- Veröffentlichung ausführen: siehe Kapitel 7.5.9.3 Seite 372
- Skript ausführen: siehe Kapitel 7.5.9.4 Seite 381
- Projektsicherung durchführen siehe Kapitel 7.5.9.5 Seite 384
- Mail verschicken: siehe Kapitel 7.5.9.6 Seite 385
- Referenzen reparieren siehe Kapitel 7.5.9.7 Seite 386

7.5.6.2 Anlegen einer serverbezogenen Aktion

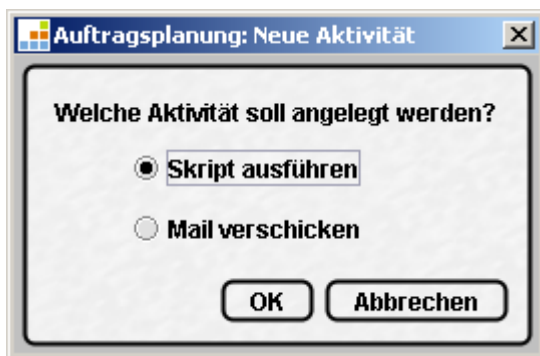


Abbildung 7-137: Anlegen einer serverbezogenen Aktion

Entsprechend der getroffenen Auswahl wird die Eingabemaske des ausgewählten Aktionstyps geöffnet:

- Skript ausführen: siehe Kapitel 7.5.9.4 Seite 381
- Mail verschicken: siehe Kapitel 7.5.9.6 Seite 385

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 356) kann eine bereits erstellte Aktion, aus einem anderen Auftrag, in den aktuell bearbeiteten Auftrag kopiert werden. Es öffnet sich das folgende Dialogfenster:



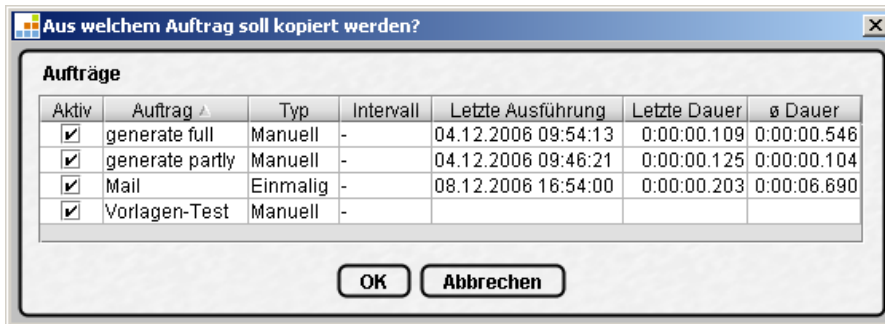


Abbildung 7-138: Aktion kopieren

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:

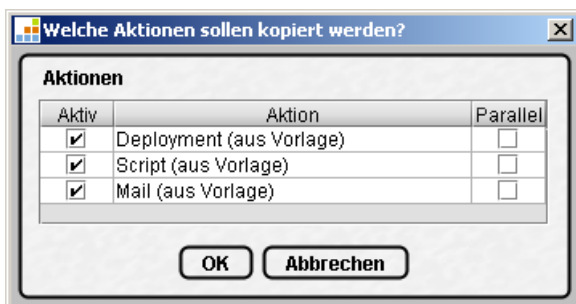


Abbildung 7-139: Aktion kopieren – Auswahl der zu kopierenden Aktion

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 wieder verwendet 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 356) kann eine Aktion auf Basis einer bereits erstellten Aktionsvorlage erstellt und dem aktuell bearbeiteten Auftrag zugefügt werden. Es öffnet sich das folgende Dialogfenster:

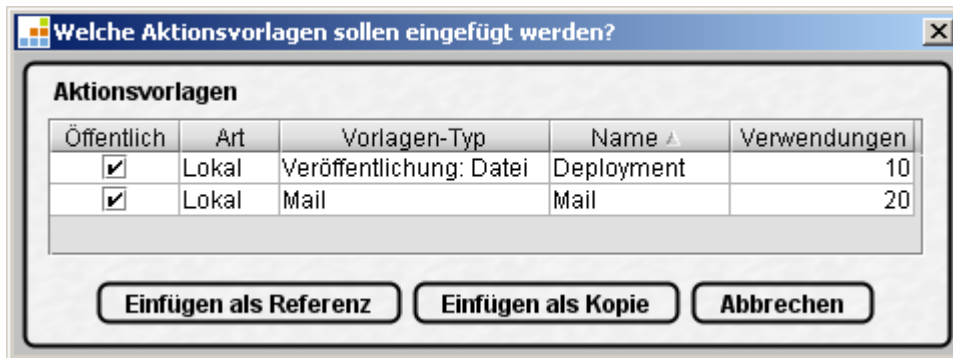


Abbildung 7-140: Aktion hinzufügen – Auswahl einer Aktionsvorlage

Abbildung 7-140 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 349). 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.

Einfügen als Referenz

Mit einem Klick auf den Button wird eine Aktion als Referenz auf eine Aktionsvorlage dem Auftrag hinzugefügt. Aktionen, welche 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.

Einfügen als Kopie

Mit einem Klick auf den Button wird eine Aktion als Kopie der selektierten Aktionsvorlage dem Auftrag hinzugefügt. Kopien einer Aktion können normal bearbeitet werden.



Abbrechen

Mit einem Klick auf den Button wird das Dialogfenster geschlossen.
Es wird keine Aktion zum Auftrag hinzugefügt.



7.5.9 Projektbezogene Aktionen

Projektbezogene Aktionen werden innerhalb der Projekteigenschaften der einzelnen Projekte angelegt und zu projektbezogenen Aufträgen hinzugefügt.

Folgende Aktionen stehen zur Verfügung:

- Alte Projektstände archivieren siehe Kapitel 7.5.9.1 Seite 363
- Generierung ausführen: siehe Kapitel 7.5.9.2 Seite 368
- Veröffentlichung ausführen: siehe Kapitel 7.5.9.3 Seite 372
- Skript ausführen: siehe Kapitel 7.5.9.4 Seite 381
- Projektsicherung durchführen siehe Kapitel 7.5.9.5 Seite 384
- Mail verschicken: siehe Kapitel 7.5.9.6 Seite 385
- Referenzen reparieren siehe Kapitel 7.5.9.7 Seite 386

7.5.9.1 Alte Projektstände archivieren (ab V4.1)



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".



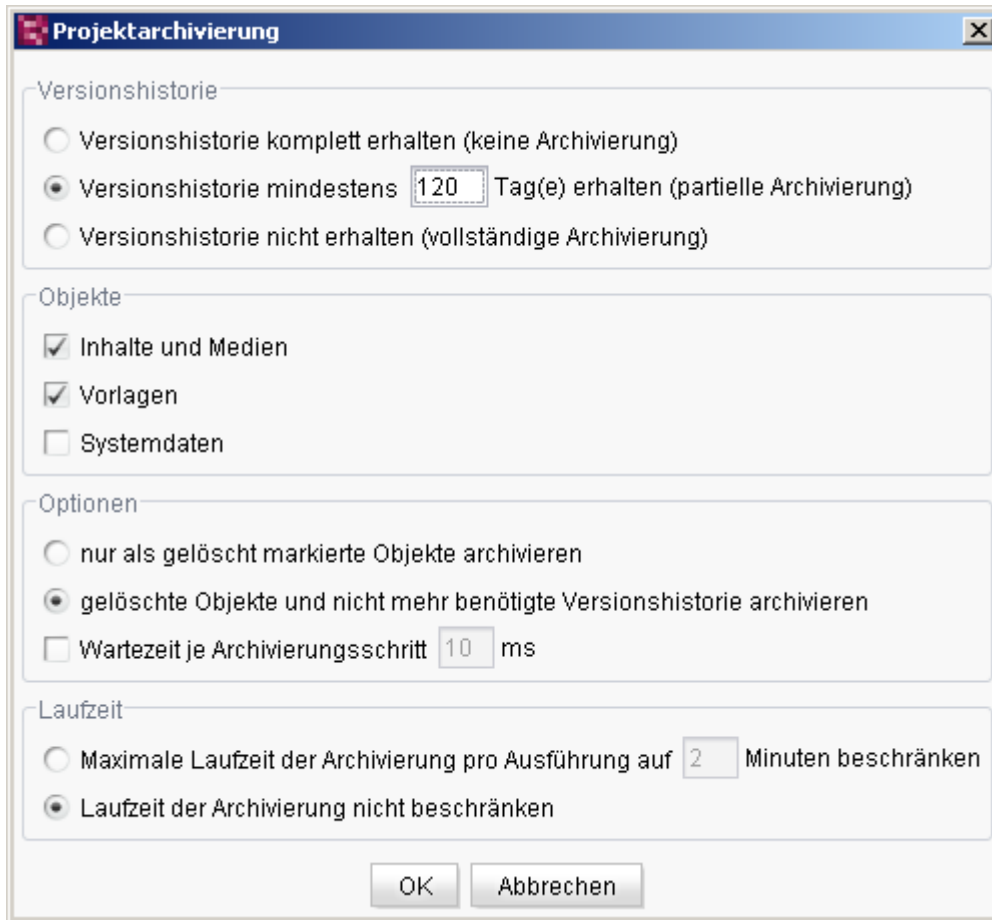


Abbildung 7-141: Auftrag anlegen – Projektarchivierung

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 213).

Weitere Informationen zum FirstSpirit Archivierungs-Konzept befinden sich in Kapitel 7.9 ab Seite 408.



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 210) 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.



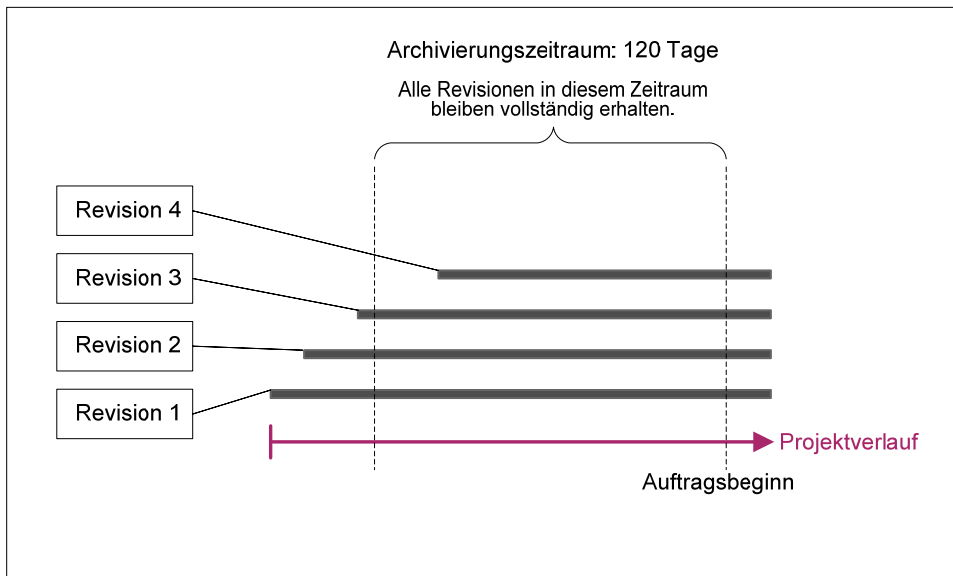


Abbildung 7-142: Partielle Archivierung über 120 Tage

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 409) 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).

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 408 und Kapitel 7.9.2 Seite 408). Ist diese Option aktiviert, werden nicht mehr verwendete Systemdaten archiviert. Auch Revisionen gehören zu den Systemdaten. Die archivierten Revisionen werden später über die Funktion "Archiv" auf dem Register "Revisionen" angezeigt (siehe Abbildung 7-20).





Ab FirstSpirit Version 4.2R2 können auch geschlossene Aufgaben archiviert werden. Dazu 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 Zeitraum 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.



7.5.9.2 Generierung ausführen

Auftragsplanung: Generierung ausführen

Eigenschaften | Erweitert

Name:

Vollgenerierung durchführen

Teilgenerierung durchführen für folgende Startpunkte

Startpunkte

| ID |
|----|
| |

Vom Benutzer definierbar? (nur für interaktive Aufträge)

Generieren nur wenn erforderlich

Generierungsverzeichnis vorher leeren

Medien im Generierungsverzeichnis erzeugen

ACL-Datenbank nutzen

Pfad erzeugung:

Präfix für absolute Pfade:

Erfolgreich nur wenn:

Schwellwert für normale Fehler:

Schwellwert für Warnungen:

Abbildung 7-143: Aktion anlegen – Generierung ausführen

Diese Aktion dient dazu eine Voll- bzw. Teilgenerierung des gewählten Projekts durchzuführen (siehe auch Kapitel 8.4 Seite 433).

Eine Vollgenerierung generiert alle Inhalte des Projekts, eine Teilgenerierung nur die ausgewählten "Startpunkte" und deren Kinder.

Vollgenerierung durchführen: Ist diese Option *aktiviert*, wird das Projekt bei der Auftragsausführung vollständig generiert.

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

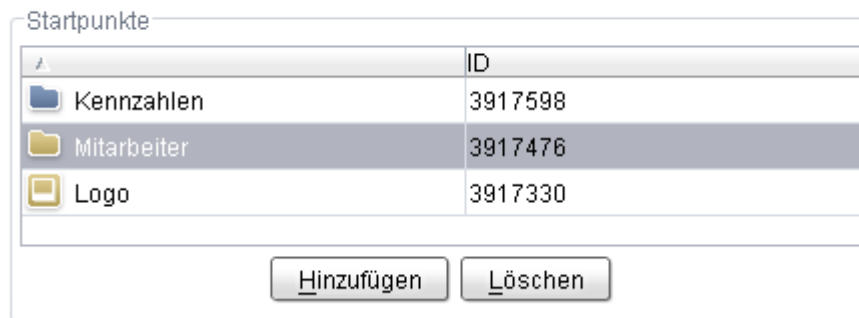


Abbildung 7-144: Aktion anlegen – Teilgenerierung ausführen

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.

Hinzufügen

Durch einen Klick auf diesen Button öffnet sich ein Dialog, der alle verfügbaren und noch nicht ausgewählten Startpunkte des Projekts anzeigt. Dabei werden ausschließlich freigegebene Objekte eingeblendet.

Löschen

Durch einen Klick auf diesen Button 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.

Ab FirstSpirit Version 4.1 können innerhalb eines Generierungsauftrags auch Startpunkte für die Generierung von Medien definiert werden.

Hintergrund: Beim Veröffentlichen von Projektinhalten 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.

Unter FirstSpirit Version 4.0 konnte man diese Medien über die Option "Alle Medien in diesem Ordner bei der Generierung kopieren" zur Generierung



hinzufügen. Diese Option wurde für alle Generierungen berücksichtigt. Die Option ist nur noch im "Kompatibilitätsmodus 3.1" weiterhin verfügbar (siehe Kapitel 7.4.2 Seite 280). Diese Option entfällt mit Version 4.2 (vgl. *FirstSpirit Release Notes* für die Version 4.1 und 4.2).



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.

Ab FirstSpirit Version 4.1 kann die Generierung dieser Medien 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.

| | |
|--|---|
| <input type="checkbox"/> | Generieren nur wenn <u>e</u> rforderlich |
| <input type="checkbox"/> | Generierungsverzeichnis vorher <u>l</u> eeren |
| <input checked="" type="checkbox"/> | <u>M</u> edien im Generierungsverzeichnis erzeugen |
| <input checked="" type="checkbox"/> | ACL-Datenbank <u>n</u> utzen |
| Pfaderzeugung | Default URLs <input type="button" value="v"/> |
| Präfix für <u>a</u> bsolute Pfade | <input type="text"/> |
| Erfolgreich <u>n</u> ur wenn | Keine schweren Fehler und Schwellwerte eingehalten <input type="button" value="v"/> |
| Schwellwert für normale <u>F</u> ehler | <input type="text" value="0"/> |
| Schwellwert für <u>W</u> arnungen | <input type="text" value="0"/> |

Abbildung 7-145: Ausschnitt (siehe Abbildung 7-143)



Generieren nur wenn erforderlich: Ist diese Option *aktiviert*, wird vor der Generierung eines Knotens überprüft, ob der Knoten seit der letzten Generierung geändert wurde. Wurde der Knoten nicht geändert, wird er auch nicht neu generiert. Ist die Option *deaktiviert*, werden alle Knoten neu generiert, auch wenn seit der letzten Generierung keine Änderungen vorgenommen wurden.

Generierungsverzeichnis vorher leeren: Ist diese Option *aktiviert*, wird das Generierungsverzeichnis vor Beginn der Generierung geleert.

Medien im Generierungsverzeichnis erzeugen: Ist diese Option *aktiviert*, werden Medien bei der Generierung im Generierungsverzeichnis erzeugt. Ist die Option *deaktiviert*, werden keine Medien bei der Generierung erzeugt. Sinnvoll ist diese Einstellung, wenn ausschließlich eine Veröffentlichung über die Struktur-Verwaltung erfolgen soll und referenzierte Medien (auf der Seite) nicht berücksichtigt werden sollen, beispielsweise weil diese in einem gesonderten Auftrag generiert werden.

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 drei verschiedene Arten der Erzeugung:

- **Default-URLs:** Bei dieser Methode existiert für jede Sprache des Projekts ein eigener Unterordner auf dem Webserver (de, en, etc.).
- **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 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

7.5.9.3 Veröffentlichung ausführen

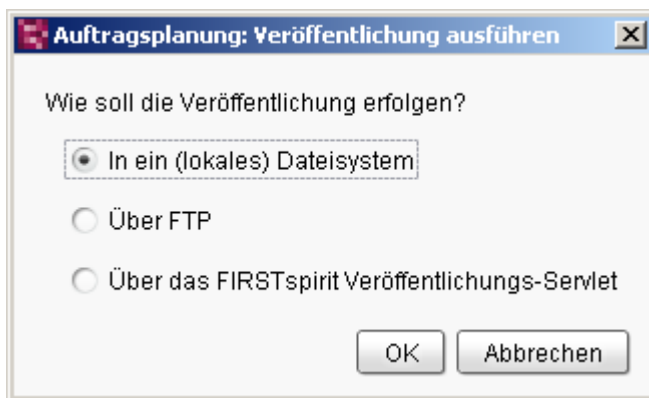


Abbildung 7-146: Aktion anlegen – Veröffentlichung ausführen

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 7.5.9.3.1 Seite 373)
- Veröffentlichung über FTP (siehe Kapitel 7.5.9.3.2 Seite 375)
- Veröffentlichung über ein FirstSpirit Veröffentlichungs-Servlet (siehe 7.5.9.3.3 Seite 378)





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.

7.5.9.3.1 Veröffentlichung in ein lokales Dateisystem



Abbildung 7-147: 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

Optionen – 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.

Optionen – 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.

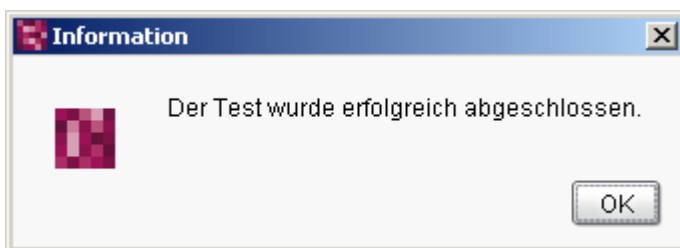
Pfad ins lokale Dateisystem: Hier wird festgelegt, in welches lokale Verzeichnis



veröffentlicht werden soll.

Optionen – Datum an Verzeichnisnamen anhängen: Ist diese Option *aktiviert*, wird den oben angegebenen Pfaden das aktuelle Datum angehängen.

Per Klick auf diesen Button 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 öffnet sich ein Dialog mit entsprechender Log-Datei, welche die aufgetretenen Fehler aufzeigt.

Zum Konzept "Sichere Medien" siehe "Dokumentation zum Modul FirstSpirit Security".



7.5.9.3.2 Veröffentlichung über FTP

The screenshot shows a dialog box titled "Veröffentlichung: FTP". It is divided into several sections:

- Allgemeine Veröffentlichungseigenschaften:** A text field labeled "Name" contains the text "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 several input fields: "FTP-Server", "Benutzer", "FTP-Server-Typ" (a dropdown menu currently showing "UNIX"), "Passwort", and a checkbox labeled "Passiv Modus".
- Basis Pfad auf FTP-Server:** An empty text input field.
- FTP-Proxy Einstellungen:** A checkbox "FTP-Proxy verwenden?" is unchecked. Below it are fields for "Proxy-Server" and "Port". Two radio buttons are present: "USER@SITE"-Protokoll (unchecked) and "USER with Login"-Protokoll (checked). 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 7-148: 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

Optionen – 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.

Optionen – 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 eingetragen 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.

Optionen – 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

Optionen – 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.

Optionen – "USER@SITE"-Protokoll: Ist diese Option *aktiviert* erfolgt die Authentifizierung beim Proxy per "USER@SITE"-Protokoll, also ohne gesonderte Benutzererkennung.

Optionen – "USER with Login"-Protokoll: Ist diese Option *aktiviert* erfolgt die Authentifizierung beim Proxy per "User with Login"-Protokoll mit folgender Benutzererkennung. Die Angabe von Benutzername und Kennwort sind in diesem Fall



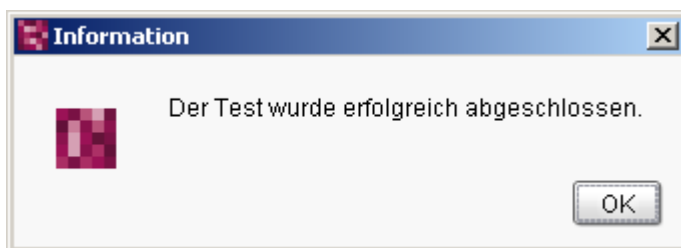
zwingend erforderlich.

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.


 Konfiguration testen

Per Klick auf diesen Button 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".



7.5.9.3.3 Veröffentlichung per FirstSpirit Veröffentlichungs-Servlet

Veröffentlichung: FIRSTspirit Veröffentlichungs-Servlet

Allgemeine Veröffentlichungseigenschaften

Name

Veröffentlichungsoptionen

Komplett-Abgleich (nur für Vollgenerierung) Abgleich (ohne Löschen)

Veröffentlichung: FIRSTspirit Veröffentlichungs-Servlet

Servlet-URL Benutzer

Timeout Sekunden Passwort

Pfad auf Live-Server

HTTP-Proxy Einstellungen

HTTP-Proxy verwenden?

Proxy-Server Port

Benutzer Passwort

Konfiguration testen OK Abbrechen ?

Abbildung 7-149: 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-Datenbank²⁹ 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).

²⁹ 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

Optionen – Komplet-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.

Optionen – 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 Web-Browser aufgerufen werden, z. B. über:

http://www.mydomain.de/fs4staging_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.

Basis Pfad auf Live-Server: Hier wird der Pfad zu dem Verzeichnis auf dem entfernten Server angegeben in welches die generierten Daten übertragen werden sollen. Das hier angegebene Verzeichnis kann potentiell gelöscht werden ("Optionen – Generierungsverzeichnis vorher leeren" – siehe Kapitel 7.5.9.2 Seite 368). 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 504), muss zusätzlich das Präfix für die Access-Control-Datenbank angegeben werden (siehe Kapitel 7.4.16 Seite 315).

HTTP-Proxy Einstellungen

Optionen – 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.

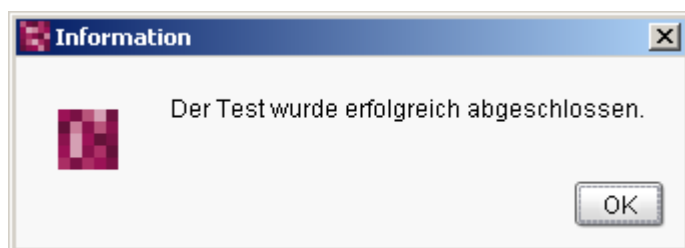
Port: Der Port zum oben angegebenen Proxy-Server. Diese Angabe ist zwingend erforderlich.

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.

Konfiguration testen

Per Klick auf diesen Button 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 anzeigt.

7.5.9.4 Skript ausführen

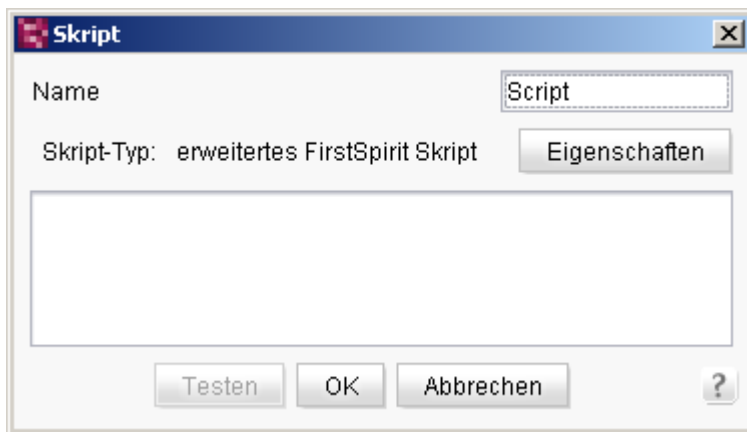


Abbildung 7-150: Aktion anlegen – Skript ausführen

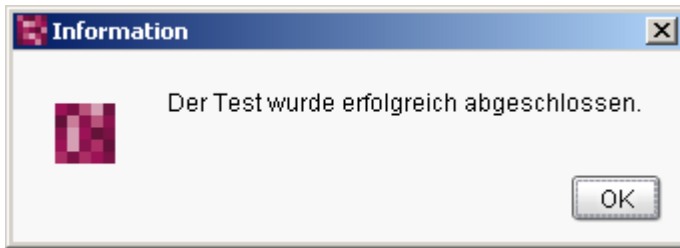
Name: In dieses Feld kann ein Name für das Skript vergeben werden.

Texteingabefeld: In diesem Feld ist der Skript-Code einzutragen, der in dieser Aktion ausgeführt werden soll.

Eigenschaften Per Klick auf den Button öffnet sich ein neuer Dialog, in welchem die Eigenschaften dieses Skripts editiert werden können (siehe Kapitel 7.5.9.4.1 Seite 382). Dies ist beispielsweise dann notwendig, wenn das Skript ändernde Operationen im Projekt durchführen soll (z. B. Objekte anlegen oder ändern).


Testen Per Klick auf diesen Button kann die eingestellte Konfiguration getestet werden. Erst wird überprüft, ob alle benötigten Parameter angegeben sind. Im Anschluss versucht der Server, den eingegebenen Skript-Code mit den konfigurierten Eigenschaften auszuführen. Ist der Test erfolgreich, erscheint folgende Meldung:





Im Fehlerfall erscheint diese Meldung:



Per Klick auf den Button  öffnet sich ein Dialog mit entsprechender Log-Datei, welche die aufgetretenen Fehler anzeigt.

7.5.9.4.1 Skript Eigenschaften

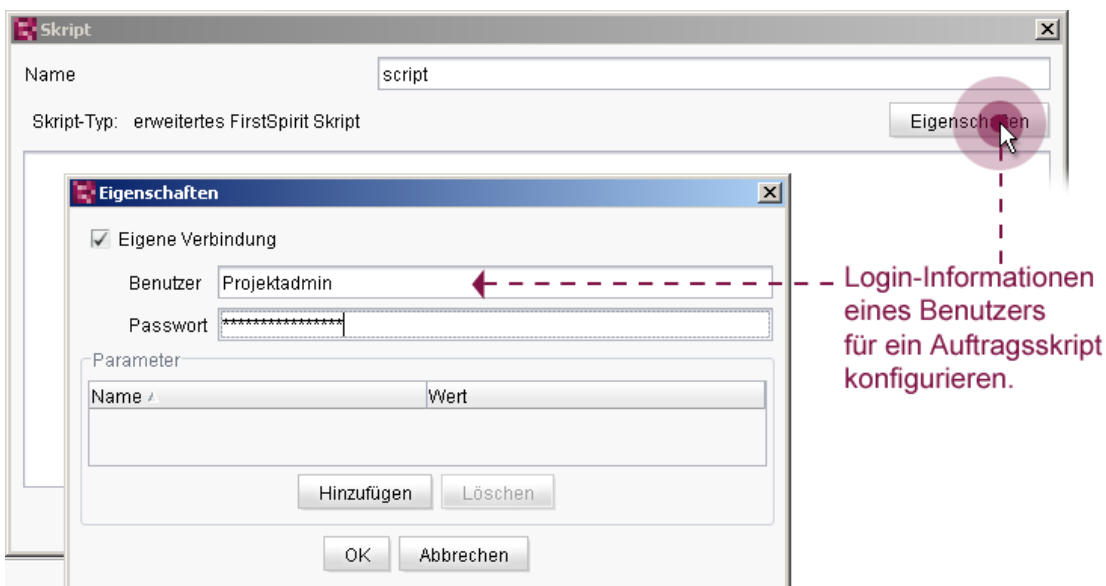


Abbildung 7-151: Skript editieren – Skript Eigenschaften

Über diesen Dialog kann – auf Wunsch – eine eigene Verbindung zum Server aufgebaut werden, anstatt die laufende Sitzung zur Ausführung des Skripts zu nutzen.



Optionen – Eigene Verbindung: Ist diese Option *aktiviert*, wird anhand folgender Daten zur Ausführung des Skripts eine neue Serververbindung aufgebaut.

Benutzer: Der Benutzername, welcher zur Anmeldung am Server verwendet werden soll.

Passwort: Das entsprechende Passwort zum oben angegebenen Benutzer.

Parameter: Hier werden alle Parameter eingegeben, welche 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.

In älteren FirstSpirit Versionen war es möglich, Auftragskripte über die System-Verbindung auszuführen, z. B.:

```
context.getUserService();
```

Darüber konnten verändernde Operation im Projekt ohne Rechteprüfung ausgeführt werden. Diese Möglichkeit erzeugte in FirstSpirit Version 4.0 bereits eine "Deprecation"-Warnung und ist in FirstSpirit Version 4.1 nicht mehr möglich.

Skripte, die ohne spezielle Benutzerinformationen konfiguriert wurden, können zwar auch in FirstSpirit Version 4.1 weiterhin ausgeführt werden. Dazu wird jedoch eine spezielle Verbindung genutzt, die zwar lesenden Zugriff besitzt (über die Rechte CAN_SEE, CAN_READ, CAN_META_SEE), aber keine verändernden Operationen mehr ausführen kann. In diesem Fall wird die Skriptausführung mit einer Security-Exception abgebrochen.

Innerhalb eines Auftragskripts stehen die beiden Variablen `context` und `connection` zur Verfügung. Ab FirstSpirit Version 4.1 stellt die Variable `context` eine spezielle Verbindung (ohne Benutzerinformationen) bereit, die nur noch lesenden Zugriff besitzt (s.o.). Für das Ausführen einer verändernden Operation über ein Auftragskript muss nun zunächst ein Benutzer über die Skripteigenschaften definiert werden, der die entsprechenden Rechte besitzt (z. B. über die Rechte CAN_CHANGE, APPEND, DELETE, etc).

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).



Änderung des Security-Modells in FirstSpirit Version 4.2: 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 Projektsicherung durchführen

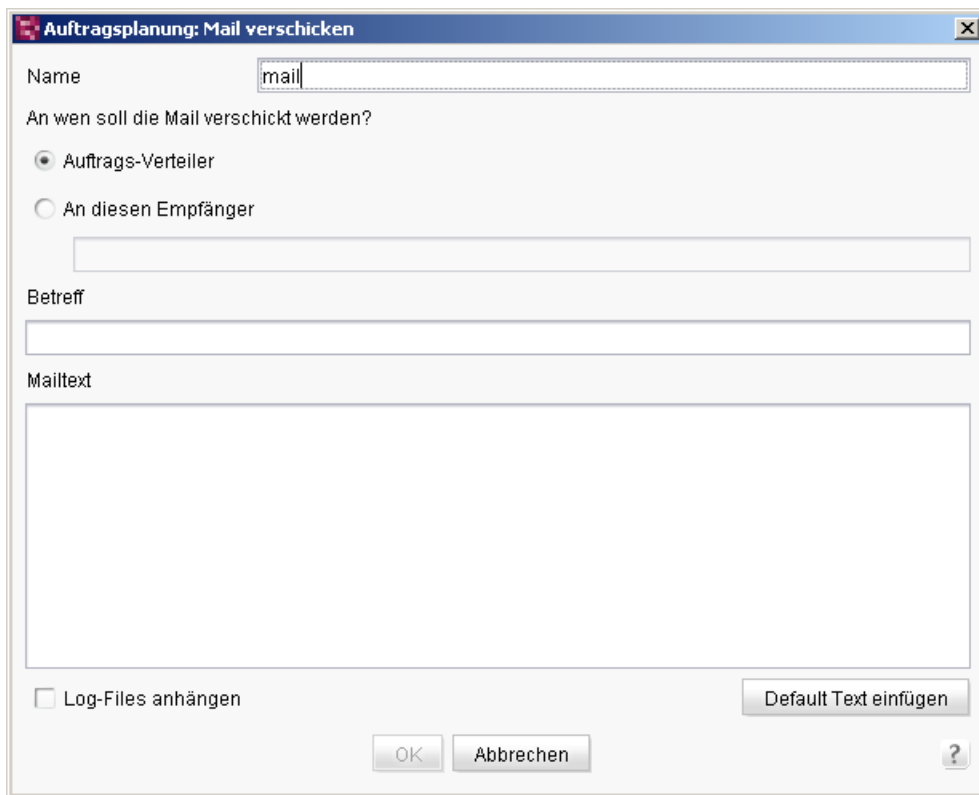
Die Projektsicherung dient dazu, einen aktuellen Stand des Projekts zu exportieren. Die Projektsicherung wird automatisch in dem Pfad angelegt, der in der Konfigurationsdatei `fs-server.conf` als Backup-Path definiert wurde. (Dabei ist es möglich, das Verzeichnis auf eine andere Festplatte auszulagern (siehe Kapitel 4.3.1.8 Seite 48).) Das Löschen nicht mehr für die Projektsicherung benötigter Exportdateien erfolgt über die Funktion "Server aufräumen" (siehe Kapitel 7.2.2.1 Seite 198).

Ab FirstSpirit Version 4.2 ermöglicht das lizenzabhängige Modul "FirstSpirit Enterprise Backup" 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 "FirstSpirit Release Notes zur Version 4.2" und "Dokumentation zum Modul FirstSpirit Enterprise Backup".



7.5.9.6 Mail verschicken

**Abbildung 7-152: 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.

Optionen – Auftrags-Verteiler: Ist diese Option *aktiviert*, wird die E-Mail an alle Empfänger versendet, die im Verteiler des zugehörigen Auftrags definiert sind.

Optionen – 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 werden³⁰.

Beispiele:

`$CMS_VALUE(#context.getStartTime())$`: gibt den Start-Zeitpunkt des Auftrags aus

`$CMS_VALUE(#task.getSubject())$`: gibt den Betreff der versendeten E-Mail aus

Optionen – 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.

Default Text einfügen

Bei Klick auf diesen Button wird dem vorhandenen Mailtext der Inhalt der Konfigurationsdatei für die Standardvorlage (%Serververzeichnis%/conf/server/DefaultMailText.txt) angehängt.

7.5.9.7 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 Serverbezogene Aktionen

Serverbezogene Aktionen werden innerhalb der Servereigenschaften angelegt und zu serverbezogenen Aufträgen hinzugefügt.

Folgende Aktionen stehen zur Verfügung:

- Skript ausführen: Analog zu Kapitel 7.5.9.4 (siehe Seite 381)
- Mail verschicken: Analog zu Kapitel 7.5.9.6 (siehe Seite 385)

³⁰ 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`).



7.6 Clustering: Lastverteilung bei der Generierung (ab V4.1)



Diese Funktionalität ist erst ab FirstSpirit-Version 4.1 freigegeben.

7.6.1 Einleitung (ab V4.1)

FirstSpirit unterstützt 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.

Mit FirstSpirit Version 4.1 kommt nun eine "horizontale Skalierbarkeit" hinzu. Es wird somit möglich, einzelne Funktionskomponenten des FirstSpirit-Servers auf unterschiedliche Computersysteme zu verteilen.

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.



7.6.2 Konzept (ab V4.1)

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:

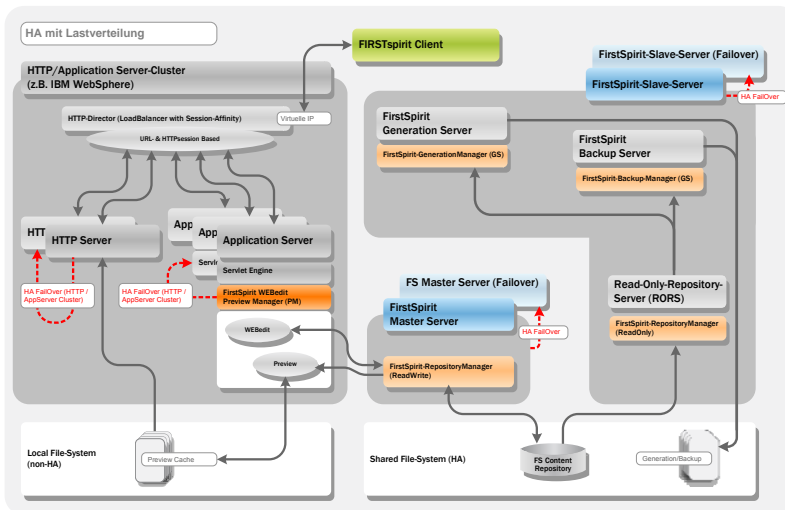


Abbildung 7-153: Hochverfügbarkeitscluster mit Lastverteilung

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.

7.6.3 Lizenzdatei prüfen (ab V4.1)

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 438). Der Parameter `license.CLUSTERING` muss für die Verwendung der Funktionalität auf den Wert `1` gesetzt sein (siehe Abbildung 7-154).

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=365
#FIRSTspirit license
#Mon Oct 22 12:16:48 CEST 2007
license.USER=e-spirit
license.EXPDATE=15.01.2008
license.MAXPROJECTS=0
license.MAXSESSIONS=0
license.MAXUSER=0
license.SOCKET_PORT=0
license.VERSION=4
license.MODULES=integration,personalisation,portal,search
license.WEBEDIT=1
license.WORKFLOW=1
license.REMOTEPROJECT=1
license.CLUSTERING=1
license.PACKAGEPOOL=1
license.DOCUMENTGROUP=1
```

Fügen Sie hier bitte den unveränderten Inhalt der
Lizenz Datei ein

Speichern

Abbrechen

Abbildung 7-154: Anzeige der Parameter der Lizenzdatei (Server-Monitoring)

7.6.4 Konfiguration der Clusterknoten (ab V4.1)

Die Konfiguration der Clusterknoten erfolgt über die FirstSpirit Server- und Projektkonfiguration im Bereich "Servereigenschaften" im Menü "Clustering" (siehe Kapitel 7.3.17 Seite 270).



7.6.5 Konfiguration des Generierungsauftrags (ab V4.1)

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 7.5.9).

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.

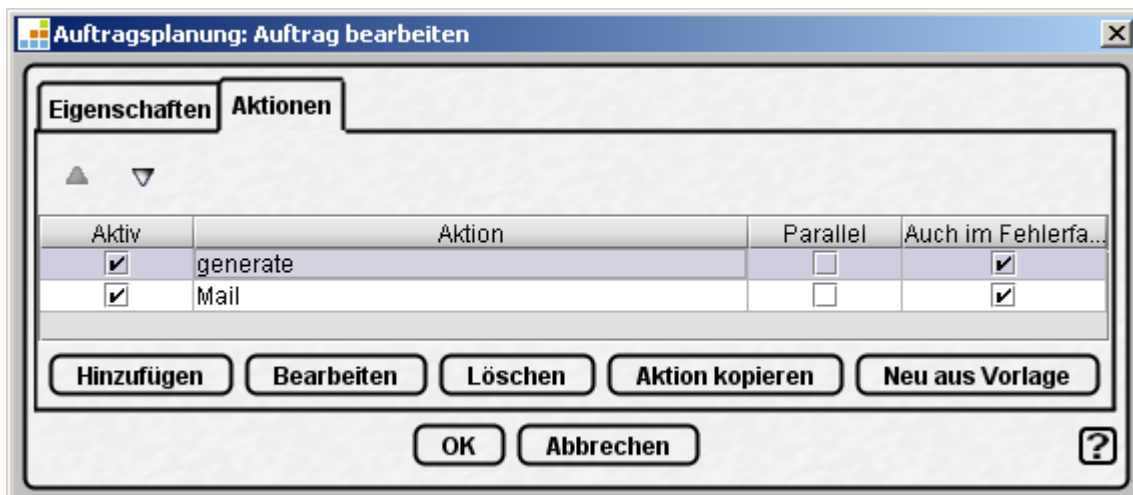


Abbildung 7-155: Projektbezogene Aktion "generate" eines Generierungsauftrags

Bearbeiten Die Konfigurationseinstellung für die Aktion können mit einem Klick auf den Button aufgerufen werden. Es öffnet sich der folgende Dialog:





Abbildung 7-156: Erweiterte Einstellungen zur Aktion "generate"

Neben den bekannten Einstellungen zur Generierung (siehe Kapitel 7.5.9) 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 458).



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 396) und projektspezifisch (siehe Kapitel 7.7.7 Seite 403) 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 7.3.14 Seite 251).

Ein rechteckiger Button mit der Aufschrift "Installieren".

Mit einem Klick auf den Button 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":



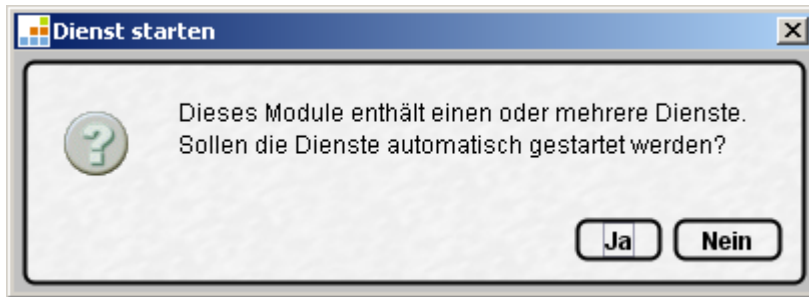


Abbildung 7-157: Dienste automatisch starten

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 401).

Die erfolgreiche Installation der fsm-Datei wird durch eine Meldung bestätigt. Das Modul "SpellService" wird anschließend im Bereich "Module" aufgelistet.



Abbildung 7-158: SpellService installiert

Im Anschluss an die Installation kann die Konfiguration der Projekt-lokalen Komponente erfolgen (siehe Kapitel 7.7.6 Seite 401).

7.7.1.2 Deinstallieren

Die Deinstallation des SpellService erfolgt über die Servereigenschaften im Bereich "Module" (siehe Kapitel 7.3.14 Seite 251).

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:



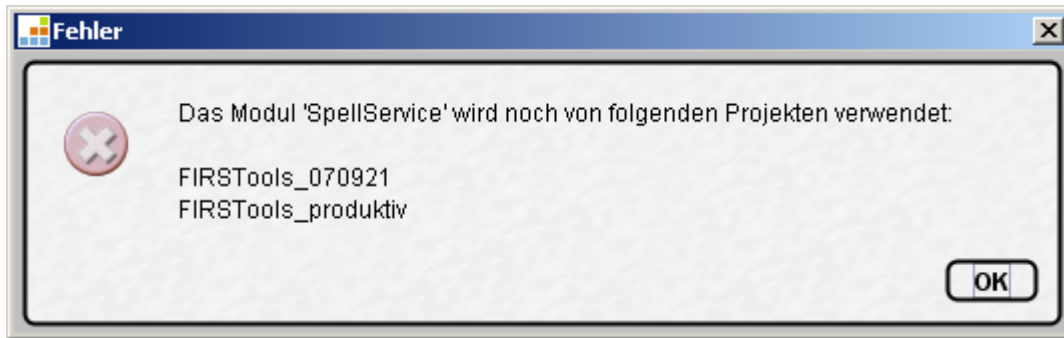


Abbildung 7-159: SpellService deinstallieren – Fehlermeldung

In diesem Fall müssen zunächst die Projekt-Komponenten innerhalb der Projekteigenschaften gelöscht werden (siehe Kapitel 7.7.6 Seite 401). Erst wenn keine Verwendungen mehr vorliegen, kann das Modul erfolgreich deinstalliert werden.



Mit einem Klick auf den Button wird das Modul deinstalliert.



Abbildung 7-160: 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.)




Mit einem Klick auf den Button öffnet sich ein Dialog zur Auswahl der fsm-Datei aus dem lokalen Dateisystem. Nach erfolgreicher Installation der neuen Version, kann die Version innerhalb der Projekteigenschaften ebenfalls aktualisiert werden.

Die bisherigen Konfigurationseinstellungen bleiben bei der Aktualisierung erhalten.



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 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.

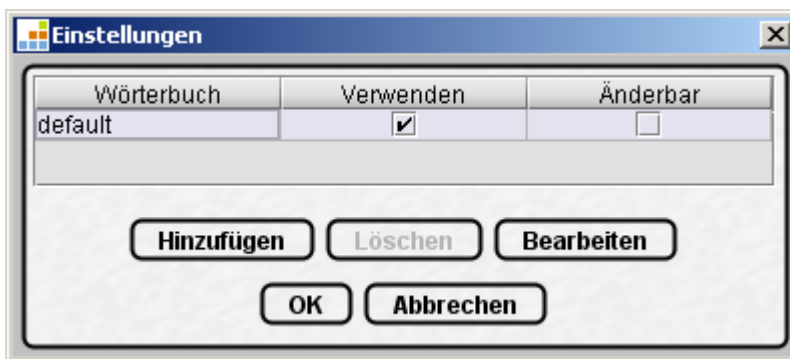


Abbildung 7-161: Globale Konfigurationseinstellungen "SpellService"

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.


Hinzufügen

Mit einem Klick auf den Button 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.

Löschen

Mit einem Klick auf den Button 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.

Bearbeiten

Mit einem Klick auf den Button kann die Konfiguration für das Wörterbuch bearbeitet werden. Es öffnet sich der Dialog zur Konfiguration (siehe Kapitel 7.7.4 Seite 398).

OK

Die Änderungen werden mit einem Klick auf den Button gespeichert.

Abbrechen

Die Änderungen werden mit einem Klick auf diesen Button nicht gespeichert, das Fenster wird geschlossen.



7.7.4 Globale Wörterbücher konfigurieren

 Mit einem Klick auf den Button öffnet sich der Dialog zur Konfiguration des Wörterbuchs:

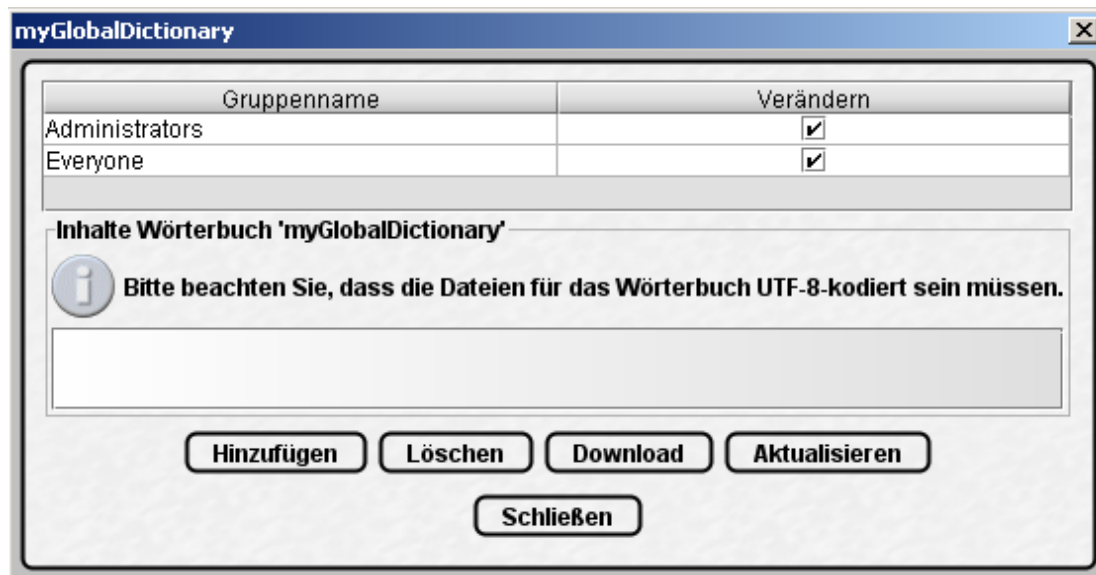


Abbildung 7-162: Wörterbuch konfigurieren

Der Dialog ist unterteilt in einen Bereich zur Rechtedefinition (siehe Kapitel 7.7.4.1 Seite 398) und einen Bereich zum optionalen Hinzufügen von sprachabhängigen Inhalten zum Wörterbuch (Dict-Dateien) (siehe Kapitel 7.7.4.2 Seite 399).

7.7.4.1 Berechtigungen für globale Wörterbücher konfigurieren

| Gruppenname | Verändern |
|----------------|-------------------------------------|
| Administrators | <input checked="" type="checkbox"/> |
| Everyone | <input type="checkbox"/> |

Abbildung 7-163: Berechtigungen konfigurieren

Im oberen Bereich des Konfigurationsdialogs (vgl. Abbildung 7-162) 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 405).

Die hier definierten globalen Rechte beeinflussen die Schreib-Leserechte innerhalb der Projektkonfiguration, können jedoch dort geändert werden (siehe Kapitel 7.7.8.1 Seite 405).





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 7-164: Inhalte hinzufügen

Im unteren Bereich des Konfigurationsdialogs (vgl. Abbildung 7-162) 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.

Hinzufügen

Mit einem Klick auf den Button 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:



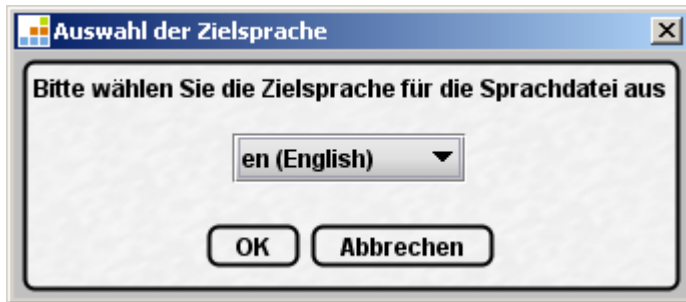


Abbildung 7-165: Zielsprache auswählen

Die Klappliste zeigt dabei alle Sprachen an, die im Bereich "Sprach-Vorlagen" innerhalb der Server-Eigenschaften definiert wurden (siehe Kapitel 7.3.7 Seite 242). 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`

Löschen

Mit einem Klick auf den Button kann eine zuvor hinzugefügte Wörterliste (Dict-Datei) wieder entfernt werden. Vor dem Löschen erfolgt eine Sicherheitsabfrage. Das Wörterbuch "default" kann nicht entfernt werden, es ist aber möglich, die Verwendung zu deaktivieren.

Download

Mit einem Klick auf den Button 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.

Aktualisieren

Mit einem Klick auf den Button 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 7.3.14 Seite 251)
- Über das Server-Monitoring im Bereich FirstSpirit – Konfiguration - Dienste (siehe Kapitel 8.6.1.7 Seite 442) und im Bereich FirstSpirit – Konfiguration - Steuerung (siehe Kapitel 8.6.2.4 Seite 445).

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 393). Die globale Konfiguration wird ebenfalls über die Servereigenschaften ausgeführt (siehe Kapitel 7.7.3 Seite 396). Dabei können globale Wörterbücher angelegt und konfiguriert werden, die anschließend innerhalb der Spell-Service-Projektkonfiguration verfügbar sind (siehe Kapitel 7.7.4 Seite 398).

Die projektspezifische Konfiguration des SpellServices erfolgt anschließend über die Projekteigenschaften im Bereich "Projekt-Komponenten" (siehe Kapitel 7.4.17 Seite 317).



Bevor eine Spell-Service-Projektkonfiguration erfolgen kann, muss der SpellService gestartet sein (siehe Kapitel 7.7.5 Seite 401).



Hinzufügen Mit einem Klick auf den Button kann die Spell-Service-Konfiguration einem Projekt hinzugefügt werden.

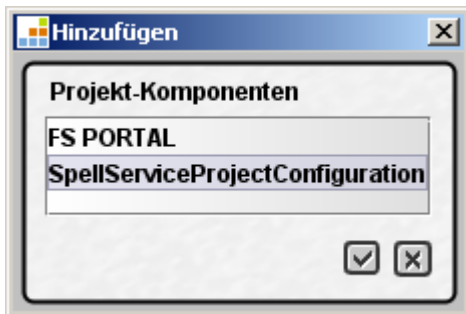


Abbildung 7-166: Servereigenschaften

Danach stehen die global konfigurierten Funktionalitäten der installierten Komponente im Projekt zur Verfügung.




Abbildung 7-167: Projekteigenschaften – Spell-Service-Projektconfiguration

Löschen Mit einem Klick auf den Button kann eine zuvor hinzugefügte Spell-Service-Projektconfiguration wieder entfernt werden. Dabei werden alle projektspezifischen Spell-Service-Konfigurationen (projektspezifische Wörterbücher, Berechtigungen) ebenfalls entfernt. Vor dem Löschen erfolgt eine Sicherheitsabfrage.

Aktualisieren Mit einem Klick auf den Button 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 395). 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 kann die projektspezifische Spell-Service-Konfiguration definiert werden (siehe Kapitel 7.7.7 Seite 403).

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.

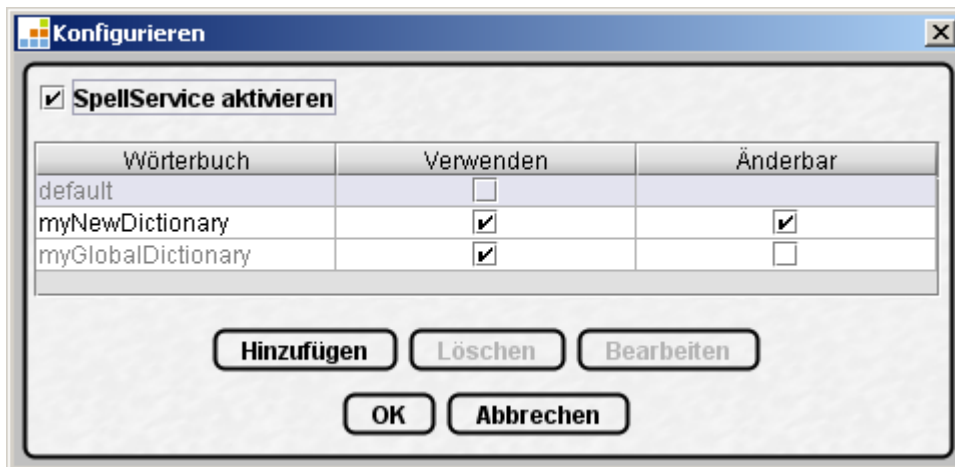


Abbildung 7-168: Konfiguration der Spell-Service-Projektconfiguration

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 Projekte 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.

SpellService aktivieren Wird die Checkbox 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.

Hinzufügen

Mit einem Klick auf diesen Button können weitere projektspezifische Wörterbücher hinzugefügt werden (siehe Kapitel 7.7.8 Seite 405). 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.

Löschen

Mit einem Klick 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.

Bearbeiten

Mit einem Klick auf den Button kann die Konfiguration für das Wörterbuch bearbeitet werden. Es öffnet sich der Dialog zur Konfiguration des Wörterbuchs (siehe Kapitel 7.7.8 Seite 405).

OK


Die Änderungen werden mit einem Klick auf den Button gespeichert.

Abbrechen

Die Änderungen werden mit einem Klick auf diesen Button 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 398).

 Mit einem Klick auf den Button öffnet sich der Dialog zur Konfiguration des Wörterbuchs (siehe Abbildung 7-162).

Der Dialog ist unterteilt in einen Bereich zur Rechtedefinition (siehe Kapitel 7.7.8.1 Seite 405) und einen Bereich zum optionalen Hinzufügen von sprachabhängigen Inhalten zum Wörterbuch (Dict-Dateien) (siehe Kapitel 7.7.4.2 Seite 399).

7.7.8.1 Berechtigungen für projektspezifische Wörterbücher konfigurieren

| Gruppenname | Verändern |
|----------------|-------------------------------------|
| Administrators | <input checked="" type="checkbox"/> |
| Everyone | <input type="checkbox"/> |
| Redakteure | <input type="checkbox"/> |
| Chefredakteure | <input type="checkbox"/> |
| Marketing | <input type="checkbox"/> |

Abbildung 7-169: Berechtigungen konfigurieren

Im oberen Bereich des Konfigurationsdialogs (vgl. Abbildung 7-162) 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 398) 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



299).

7.7.8.2 Sprachabhängige Inhalte zu globalen Wörterbüchern hinzufügen

Im unteren Bereich des Konfigurationsdialogs (vgl. Abbildung 7-162) 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 399).

Der Dialog zur Zuordnung einer Dict-Datei zu einer Projektsprache (Inhaltssprache) bietet nur die im Projekt bekannten Sprachen an (siehe 7.4.5 Seiten 290).

7.8 Unterstützung für Apache FOP (ab V4.1)

Mit FirstSpirit Version 4.1 wurde die Unterstützung für Apache FOP in ein Modul ausgelagert. Das Modul kann, wie gewohnt, über die Server- und Projektkonfiguration installiert werden (siehe Kapitel 7.3.14 Seite 251).

Mit der Freigabe von **FirstSpirit Version 4.1** sind zwei Module zur Installation verfügbar:

- **fs-fop.fsm:** neues Modul, das den aktuellen Stand von Apache FOP beinhaltet. Das Modul "fs-fop.fsm" befindet sich zum Zeitpunkt der Freigabe von FirstSpirit Version 4.1 auf dem Stand 0.95 von Apache FOP. Da mit dieser FOP-Version noch keine umfangreichen, praktischen Erfahrungen vorliegen, befindet sich das Modul bei der initialen Freigabe von FirstSpirit Version 4.1 noch im Status "Beta-Test". Eine Verwendung von Apache FOP 0.95 sollte nur nach umfangreichen Tests in Erwägung gezogen werden.
- **fs-fop_0_20_5.fsm:** herkömmliche FOP-Installation (Standard bis FirstSpirit Version 4.0). Diese Installation ist inkompatibel mit Apache FOP 0.95³¹.

Nach der Installation der Module stehen die jeweiligen "Präsentationskanäle" innerhalb der Servereigenschaften zur Verfügung (siehe auch Kapitel 7.3.2 Seite 231):

³¹ Weiterführende Informationen siehe <http://xmlgraphics.apache.org/fop/compliance.html>



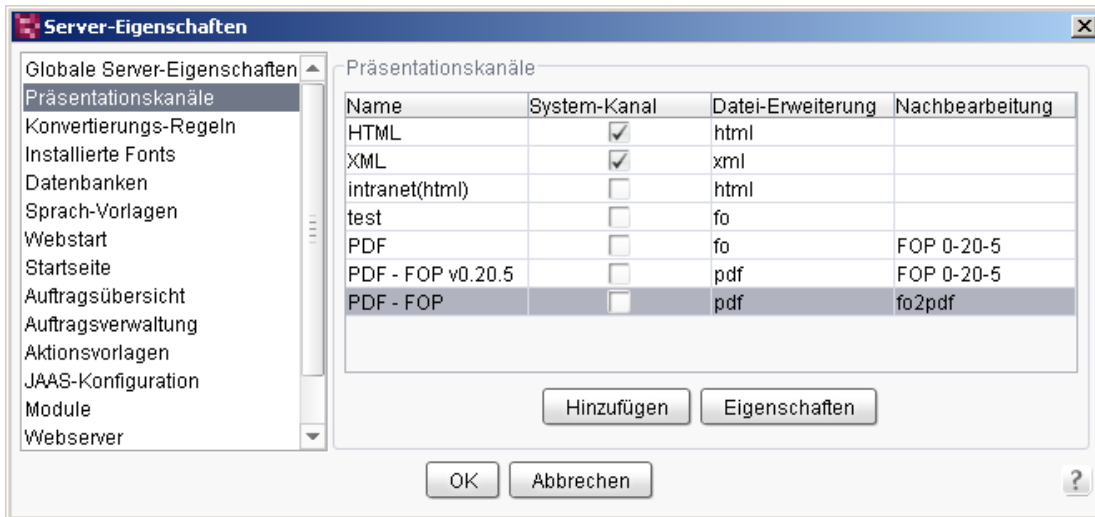


Abbildung 7-170: Präsentationskanäle (Servereigenschaften) nach der FOP-Installation

Und können anschließend in den Projekten als neuer Vorlagensatz hinzugefügt werden (siehe Kapitel 7.4.13 Seite 309):

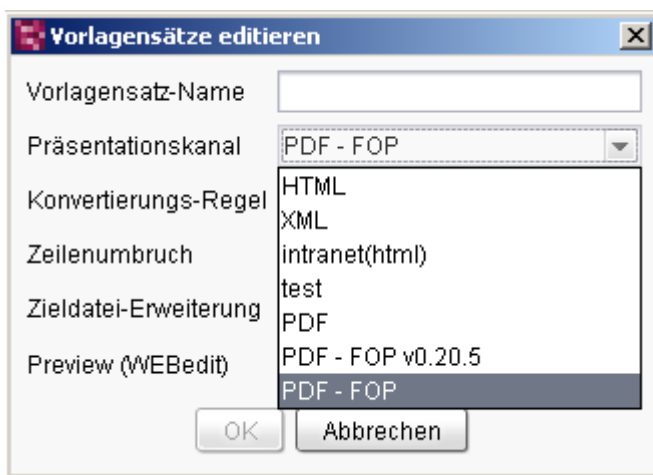


Abbildung 7-171: Vorlagensätze (Projekteigenschaften) nach der FOP-Installation



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 7.5.9.1 Seite 363) 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 welchen Personen 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 7.5.9.1 Seite 363) 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:

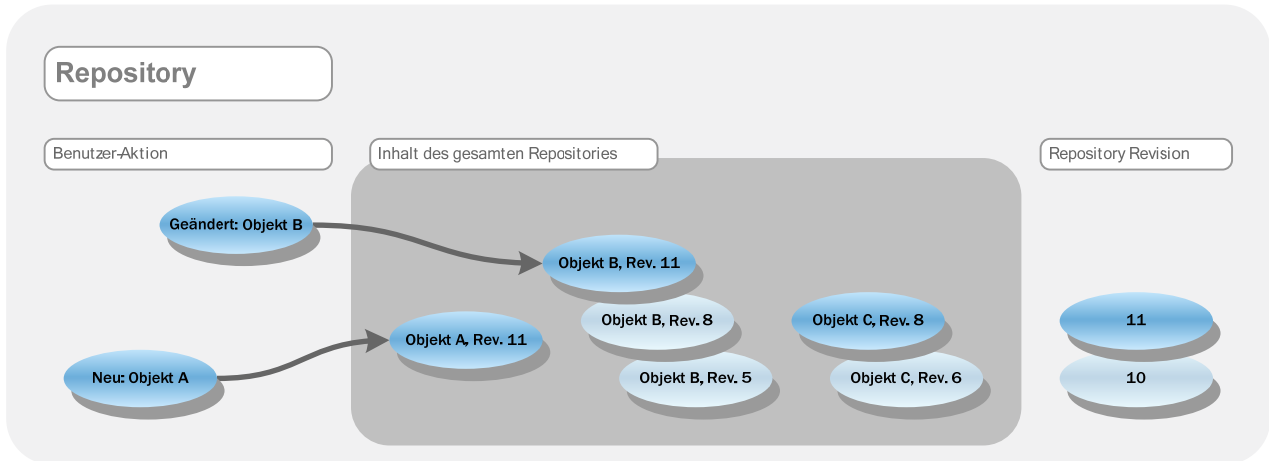


Abbildung 7-172: Beispiel Repository-Revisionen

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 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 7.5.9.1 Seite 363) 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 beispielsweise sogenannte "teilarchivierte" Revisionen.

Wird ein **Archivierungszeitraum** angegeben (partielle Archivierung, siehe Abbildung 7-142 Seite 366), 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:

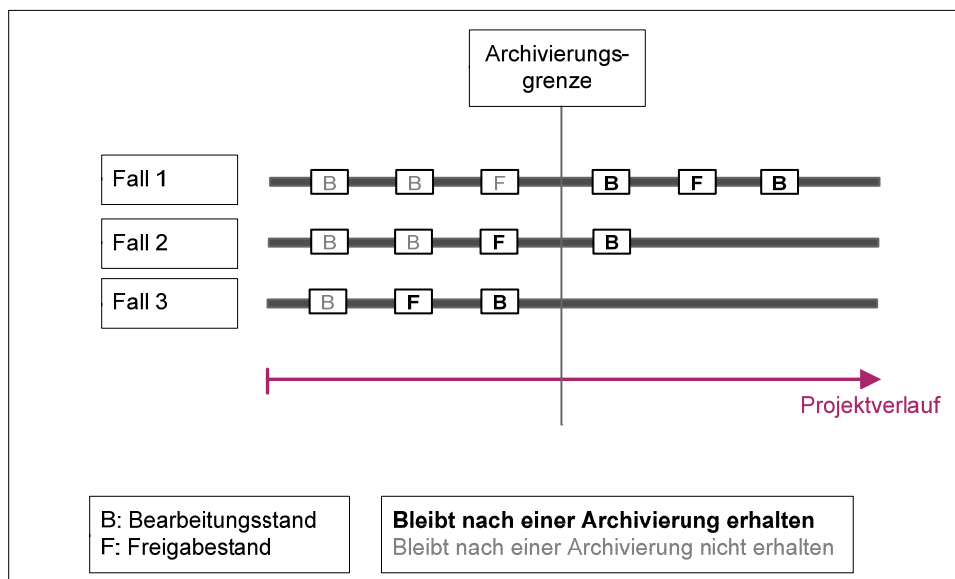


Abbildung 7-173: Mindestanforderung an die Projektarchivierung

- **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 gelöschte Objekte (Inhalte, Medien, Vorlagen, Systemdaten) aus dem Projekt ausgelagert, ist die jeweilige Revision, in der das Objekt vorhanden war, nicht mehr vollständig.

In der Versionshistorie eines Objektes wird dann nur noch die älteste vollständig vorhandene Revision angezeigt sowie alle jüngeren. "Teilarchivierte" Revisionen können über die Option "Versteckte Revisionen einblenden" angezeigt werden.

7.9.4.1 Wiederherstellen nach Archivierung

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:

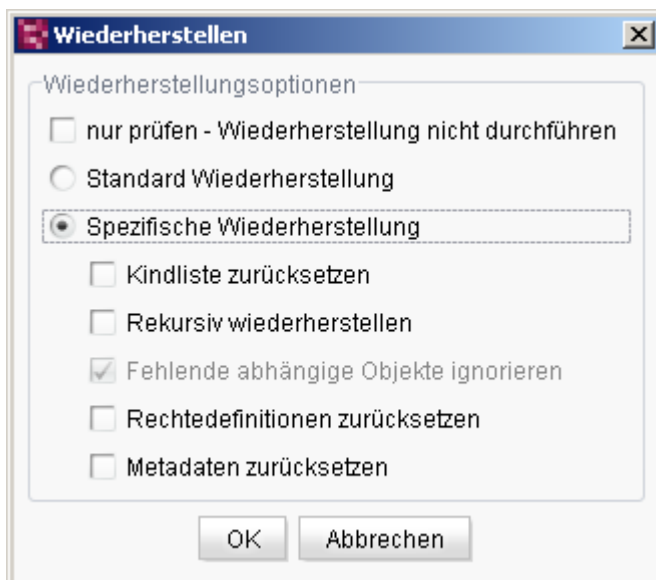


Abbildung 7-174: Wiederherstellung nach Archivierung



7.9.4.2 Vorschau von archivierten Revisionen

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:

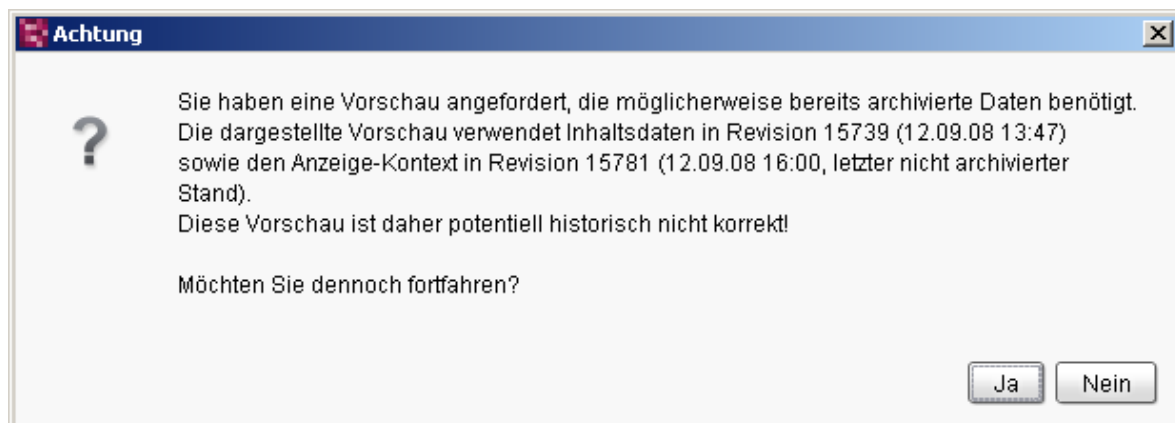


Abbildung 7-175: Meldung historisch nicht korrekte Vorschau

7.9.4.3 Vergleich von Revisionen nach Archivierung

Stehen nach einer Archivierung nicht mehr alle Teile der Objekte zur Verfügung, die für einen Vergleich notwendig sind, wird folgende Meldung angezeigt:



Abbildung 7-176: 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 Web-Browser bedient.



Die gleichzeitige Anmeldung über einen Web-Browser (z. B. in mehreren Fenstern oder Tabs) auf mehreren FirstSpirit-Servern mit demselben Hostnamen (z. B. myServer:8200 und myServer:8400) wird nicht unterstützt.

The screenshot displays the FirstSpirit Server-Monitoring web interface. At the top, there is a language selection dropdown labeled 'Sprache' and a user menu for 'Benutzer Admin' with a 'Monitoring beenden' link. The main content area is divided into several sections:

- Status:**
 - Grundinformationen (System):** Hostname des Servers: Metaxa, Server-Version: 4.2.24, Server-Verzeichnis: D:\FIRSTspirit4, IP-Adresse: 192.168.100.175, Ports: 4050/4051 (HTTP/SOCKET), Applikations-Server: intern, Lizenznehmer: e-spirit, Lizenzablauf: 15.01.2010, Server Uptime: 3h 48m 13s, Java-Version: 1.6.0_10, Java-Verzeichnis: C:\Programme\Java\jdk1.6.0_10\re, Anzahl Projekte: 34, Anzahl Benutzer: 960.
 - Grundinformationen (Load):** Aktive Projekte: 0, Aktive Benutzer: 1, Aktive Sitzungen: 1, Vorschauberechnungen: 0, Aktuell generierte Projekte: 0, Laufende Aufträge: 0, Auslastung der CPU: 1%, Anzahl laufender Threads: 56, Freier Festplattenspeicher: 8,455 GB, Aktueller Speicherverbrauch: 5%, Frei: 1,246 GB, Benutzt: 71,947 MB, Maximal: 1,317 GB.
- Speicherauslastung der letzten 24h:** A bar chart showing storage usage over a 24-hour period from 07.07.2009 13:10:25 to 08.07.2009 13:10:25. The y-axis ranges from 0M to 1.400M.
- Sessions der letzten 24h:** A bar chart showing the number of sessions over the same 24-hour period. The y-axis ranges from 0 to 50.

On the left side, there is a navigation menu with categories like 'Übersicht', 'Projekt', 'Logdateien', 'Benutzer', and 'FirstSpirit'. A red dashed line labeled 'Navigation' points to the menu items. At the top right, there is a link for 'Mailversand der angezeigten Informationen'.

Abbildung 8-1: FirstSpirit Server-Monitoring



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.

Ab FirstSpirit Version 4.1: Aktuelle Server-Status-Informationen können auch per (Text-)Mail versandt werden (siehe Kapitel 8.3.1.4 Seite 427).



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.

Ab FirstSpirit Version 4.1: Freigabe des neuen Look & Feels für das FirstSpirit Server-Monitoring (vgl. Kapitel 1.3 Seite 19).

8.1 Übersicht

8.1.1 Übersicht – Status



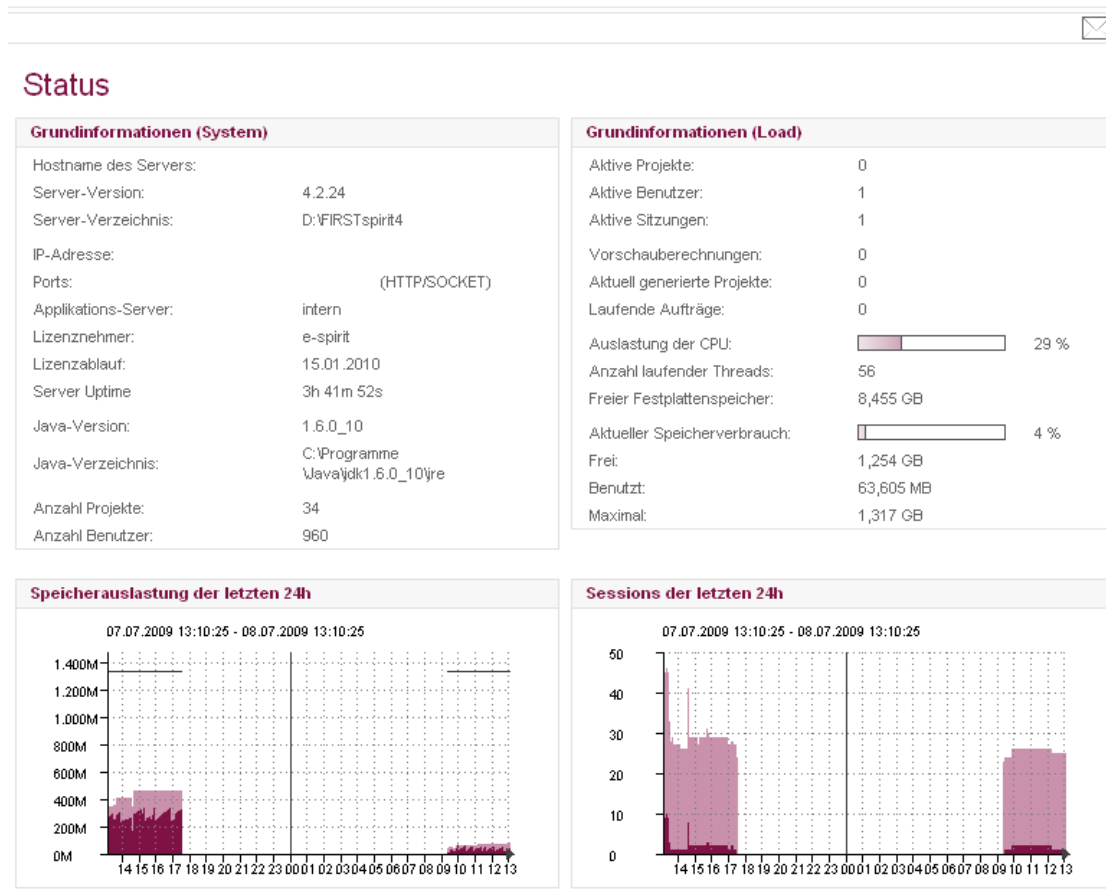


Abbildung 8-2: FirstSpirit Server-Monitoring – Übersicht – Status

Auf dieser Seite wird eine Übersicht über die wichtigsten Informationen dargestellt. 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. Durch einen Klick auf die Grafik gelangt man zum Kapitel "FirstSpirit – Monitoring – Speicher" (siehe Kapitel 8.6.5.1 ab Seite 449).

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 450).

8.1.2 Übersicht – Aktivitäten

Auf dieser Seite gibt es eine Übersicht über einige Aktivitäten auf dem Server.

Zurzeit aktive Projekte: In dieser Tabelle werden alle Projekte aufgelistet, die aktuell auf dem Server aktiv sind. Dabei werden folgende Informationen angezeigt:

Aktivitäten

| Zurzeit aktive Projekte | | | | |
|--------------------------------------|-----------|----------|--------------------|-----------------|
| Projekt | Sitzungen | Benutzer | Referenzberechnung | Suchindizierung |
| Exml-PerformanceTest | 3 | 2 | Nicht gestartet | Nicht gestartet |
| FIRSTools | 1 | 0 | Nicht gestartet | Nicht gestartet |
| FIRSTools Internet | 4 | 2 | Läuft | Nicht gestartet |

Abbildung 8-3: FirstSpirit Server-Monitoring – Überblick – Aktivitäten

Durch einen Klick auf ein aktives Projekt wird eine ausführliche Übersicht zu diesem Projekt angezeigt (siehe Kapitel 8.2.1 ab Seite 418).

- Sitzungen: Anzahl der zurzeit aktiven Sitzungen im Projekt
- Benutzer: Anzahl der zurzeit angemeldeten Benutzer im Projekt
- Referenzberechnung: Status der Referenzberechnung im Projekt (siehe Kapitel 9.12).
- Suchindizierung: Status der Indizierung für die Suche (siehe Kapitel 9.14).

Zurzeit laufende Aufträge: In dieser Tabelle werden alle Aufträge aufgelistet, die



aktuell auf dem Server aktiv sind (siehe Kapitel 7.4.21 Seite 333).

FirstSpirit Version 4.2R4: In diesem Bereich werden auch Informationen zur Neuberechnung und Konvertierung von Referenzen (die in einer alten Referenzgraph-Version vorliegen) angezeigt, die automatisch ausgeführt wird, wenn Projekte erstmalig nach einem Upgrade auf FirstSpirit 4.2R4 geöffnet werden (vgl. Kapitel 9.12 Seite 480).

Zurzeit geplante Aufträge: In dieser Tabelle werden alle zukünftig anstehenden, aktiven (nicht manuelle) Aufträge aufgelistet (siehe Kapitel 7.5.1.2 Seite 344).

Zeitplanung der letzten 24 Stunden (maximal 15 Einträge): In dieser Tabelle werden alle ausgeführten Aufträge der letzten 24 Stunden aufgelistet.

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:** Sitzung wird über den WebClient aufgebaut.
- **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 Dummy-Sessions anzeigen aktiviert wurde. Für die Auswertung der lizenzierten Sitzungen werden diese Sitzungen nicht mitgezählt (siehe Kapitel 4.3.5 Seite 92).
- **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 92).





Bei einer Sitzung vom Typ Child ist es möglich, dem Benutzer der Sitzung eine Nachricht zu schicken. Durch einen Klick auf den Sitzungstyp gelangt man zu der Seite "FirstSpirit – Nachricht" (siehe Kapitel 8.6.3 ab Seite 447). 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 436).

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 419).

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.

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 Status des Projektes (wird generiert, wird deployed...) und die ID des Projekts angezeigt.

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.

Seiten gibt an, wie viele Seiten bei der letzten vollständigen Generierung des Projektes erzeugt wurden.

Generierungen, Deployments gibt an, wie viele Generierungen (inkl. Teilgenerierungen) und Veröffentlichungen bisher durchgeführt wurden.

Max. Sessions gibt die maximale Anzahl an Sitzungen an.

Durchschn. Dauer Generierung

Dauer der letzten Generierung

Durchschn. Dauer Generierung einer Seite

Durchschn. Dauer Generierung eines Previews

Dauer Generierung des letzten Previews

Es wird jeweils die durchschnittliche bzw. letzte Dauer der Generierungen von Projekt, Seite und Vorschau angezeigt, so dass abgeschätzt werden kann, ob kürzlich vorgenommene Änderungen die Performance des Projekts beeinflusst haben.

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.

Benutzte Fonts gibt die verwendeten Font des Projektes an. Dabei werden nur die Fonts aufgelistet, die in der Funktion CMS_FONT verwendet werden.

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 420)
- Veröffentlichung: Statistik über die Veröffentlichungen (siehe Kapitel 8.2.2.2 ab Seite 421)
- Ressourcen: Statistik über die Ressourcen (siehe Kapitel 8.2.2.3 ab Seite 421)

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 419)

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. Neben dem Projektnamen gibt es Angaben über die Anzahl der Generierungen, die Anzahl der Ausführung, sowie die Dauer der letzten Veröffentlichung und die durchschnittliche Dauer einer Veröffentlichung.

Diese Liste kann nach allen Spalten der Tabelle 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 419)

8.2.2.3 Ressourcen

Auf dieser Seite befinden sich Informationen über die Ressourcen, welche die Projekte benötigen. In der Tabelle werden alle auf dem Server installierten Projekte aufgelistet. Neben dem Projektnamen gibt es Angaben über die Anzahl der Projektseiten und den benötigten Festplattenspeicher. Weiterhin werden die Anzahl der durchgeführten Vorschauen sowie die Dauer für die Berechnung einer Vorschauseite und der Generierung einer Seite dargestellt.

Diese Liste kann nach allen Spalten der Tabelle sortiert werden. Über die Option "**Nur aktive Projekte**" können alle deaktivierten Projekte ausgeblendet werden. Durch die Funktion "**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. (Siehe Kapitel 8.2.1.1 ab Seite 419)



8.3 Log-Dateien

8.3.1 Log-Dateien – kompletter Server



Diese Menüebene steht nur Serveradministratoren zur Verfügung.

8.3.1.1 Aktuelle Ansicht

Auf dieser Seite werden die aktuellsten Aktionen der Log-Datei ausgegeben.

Level Zeilen

| Jetzt | | <input type="button" value="Pause"/> | <input type="button" value="Details"/> | <input type="button" value="Ältere"/> |
|---|---|--------------------------------------|--|---------------------------------------|
| WARN | [+] 10:09:07 (fs.io.servlet.WebAuthentication): web authentication failed: javax.securi [...] | | | |
| DEBUG | [+] 10:09:07 (fs.server.authentication.FSUserLoginModule): aborting... | | | |
| DEBUG | [+] 10:09:07 (fs.server.authentication.AuthenticationManagerImpl): request for callback [...] | | | |
| DEBUG | [+] 10:09:07 (fs.server.authentication.FSUserLoginModule): try to login... | | | |
| DEBUG | [+] 10:09:07 (fs.server.authentication.AuthenticationManagerImpl): request for callback [...] | | | |
| DEBUG | [+] 10:09:07 (fs.server.authentication.LdapLoginModule): login... | | | |
| DEBUG | [+] 10:09:07 (fs.server.authentication.FSTicketLoginModule): login failed: ticket not f [...] | | | |

Abbildung 8-4: 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 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 können alle Einträge gleichzeitig in ihrer vollständigen Länge angezeigt werden.

Durch einen Klick auf die Schaltfläche ist es möglich, sich einen älteren Teil der Log-Datei ausgeben zu lassen. (siehe hierzu Kapitel 8.3.1.2 ab Seite 424)

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. Durch einen Klick auf die Schaltfläche **Anzeigen** wird die Ansicht aktualisiert.

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.

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.

| Suchergebnis (11) | |
|--------------------|---|
| DEBUG [+] | 10:56:37 (fs.server.authentication.FSTicketLoginModule): login failed: ticket not f [...] |
| DEBUG [+] | 10:56:37 (fs.server.authentication.FSTicketLoginModule): login failed: ticket not f [...] |
| DEBUG [+] | 10:56:22 (fs.server.authentication.FSTicketLoginModule): login failed: ticket not f [...] |
| DEBUG [+] | 10:56:22 (fs.server.authentication.FSTicketLoginModule): login failed: ticket not f [...] |
| DEBUG [+] | 10:56:07 (fs.server.authentication.FSTicketLoginModule): commit failed. |

Abbildung 8-5: 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.


| | | | | | | | | |
|---|---|---------------------------------------|---|--|------------------------------------|---|----------------------------------|---|
| Level | <input type="text" value="Debug"/> | Datum | <input type="text" value="01.03.2007"/> | Zeit | <input type="text" value="11:20"/> | Zeilen | <input type="text" value="100"/> | <input type="button" value="Anzeigen"/> |
| <input type="text"/> | | <input type="button" value="Suchen"/> | | | | | | |
| Thu, 01.03.2007 11:20 bis Thu, 01.03.2007 11:19 | | <input type="button" value="Pause"/> | | <input type="button" value="Details"/> | | <input type="button" value="<"/> <input type="button" value=">"/> | | |
| DEBUG [+] | 11:20:43 {pID=131818} (or.database.ConnectionPool): checked 1 connections, 0 busy [...] | | | | | | | |
| DEBUG [+] | 11:20:40 (fs.server.logging.LoggingManagerImpl): Searching within Log-Files: found [...] | | | | | | | |
| DEBUG [+] | 11:20:40 (fs.server.logging.LoggingManagerImpl): Searching within Log-Files: betwee [...] | | | | | | | |
| WARN [+] | 11:20:37 (fs.io.servlet.WebAuthentication): web authentication failed: javax.securi [...] | | | | | | | |
| WARN [+] | 11:20:37 (fs.io.servlet.WebAuthentication): web authentication failed: javax.securi [...] | | | | | | | |
| DEBUG [+] | 11:20:37 (fs.server.authentication.FSUserLoginModule): aborting... | | | | | | | |
| DEBUG [+] | 11:20:37 (fs.server.authentication.FSUserLoginModule): aborting... | | | | | | | |

Abbildung 8-6: Log-Datei – Historie

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.


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. Durch einen Klick auf die Schaltfläche  wird die Ansicht aktualisiert.

Datum: Durch einen Klick auf die Schaltfläche  öffnet sich ein Fenster für Auswahl eines Datums.

| << März 2007 >> | | | | | | |
|-----------------|----|----|----|----|----|----|
| Mo | Di | Mi | Do | Fr | Sa | So |
| | | 1 | 2 | 3 | 4 | |
| 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| 26 | 27 | 28 | 29 | 30 | 31 | |


Abbildung 8-7: Datumsauswahl

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. Durch einen Klick auf die Schaltfläche  wird die Ansicht aktualisiert.

Zeit: Durch einen Klick auf die Schaltfläche  öffnet sich ein Fenster für Auswahl einer Uhrzeit (mit einem erneuten Klick wird das Fenster wieder geschlossen).

| | | | | | | | |
|-------|-------|-------|-------|-------|-------|-------|----|
| 00:00 | 00:30 | 01:00 | 01:30 | 02:00 | 02:30 | 03:00 | |
| 03:30 | 04:00 | 04:30 | 05:00 | 05:30 | 06:00 | 06:30 | |
| 07:00 | 07:30 | 08:00 | 08:30 | 09:00 | 09:30 | 10:00 | |
| 10:30 | 11:00 | 11:30 | 12:00 | 12:30 | 13:00 | 13:30 | |
| 14:00 | 14:30 | 15:00 | 15:30 | 16:00 | 16:30 | 17:00 | |
| 17:30 | 18:00 | 18:30 | 19:00 | 19:30 | 20:00 | 20:30 | |
| 21:00 | 21:30 | 22:00 | 22:30 | 23:00 | 23:30 | | |
| | | 00:00 | | | | | >> |

Abbildung 8-8: Zeitauswahl

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  wird die Ansicht aktualisiert.



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 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.

▶ aktuelle Ansicht ▶ Historie ▶ Suche

Suche

Zeitraum Von Bis

Suchoptionen Suchwort Exceptions

Level Treffer pro Seite **Suchen**

Mon, 29.10.2007 16:07 bis Mon, 05.11.2007 16:07 **Details** < >

Logausriss Zeilen vorher Zeilen nachher **Historie anzeigen**

Abbildung 8-9: 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 (Siehe Abbildung 8-7: Datumsauswahl) und durch einen Klick auf die Schaltfläche öffnet sich ein Fenster für Auswahl einer Uhrzeit (siehe Abbildung 8-8: Zeitauswahl). 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. Durch einen Klick auf die Schaltfläche **Anzeigen** wird die Ansicht aktualisiert.

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 **Suchen** wird die Ansicht aktualisiert.

8.3.1.4 Mailversand (ab V4.1)



Abbildung 8-10: Mailversand einer Log-Datei

In FirstSpirit Version 4.1 besteht die Möglichkeit Log-Dateien, z. B. einer Veröffentlichung, innerhalb der Anwendung zum Server-Monitoring herunterzuladen und/oder als Mailanhang zu verschicken. Beim Öffnen des Mailversands über das Icon  (siehe Abbildung 8-10) werden die betreffenden Log-Dateien automatisch angehängt:



Mailversand

Empfänger

Betreff

Kommentar

Anhang

[fs-server_INFO_20080916154352_20080916154301.log \(229,562 KB\)](#)

Abbildung 8-11: Mailversand mit angehängter Log-Datei

Sie können innerhalb des Dialogs aber auch mit einem Klick auf dem Link heruntergeladen und gespeichert werden.

8.3.2 Log-Dateien – nach Projekt

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 419).





Für Projektadministratoren werden nur die Projekte aufgelistet, für die man als Administrator eingetragen ist.

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.

firstspirit.de > [Wechseln](#)

Abbildung 8-12: Protokolle – Projekt wechseln

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 423.

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 424.

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 426.



8.3.2.4 Mailversand (ab V4.1)

In FirstSpirit Version 4.1 besteht die Möglichkeit Log-Dateien, z. B. einer Veröffentlichung, innerhalb der Anwendung zum Server-Monitoring herunterzuladen und/oder als Mailanhang zu verschicken. (siehe Kapitel 8.3.1.4 Seite 427).

8.3.3 Log-Datei – nach Veröffentlichung

Auf dieser Seite befindet sich eine Tabelle mit allen auf dem Server installierten Projekten. Angezeigt werden hier der Projektname und die Projektbeschreibung. Durch den Verweis [Projekt auswählen](#) hinter jedem Projekteintrag wird die aktuelle Logansicht für das ausgewählte Projekt angezeigt.



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 419)

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 423.



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 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.

Historie

Zeitraum

Von

Bis

Max. Einträge pro Projekt

- [+] [\(QA\) FirstUnit 4.x \(PRODUKTIV\) \(5\)](#) - [07.07.2009 11:53:14 - 07.07.2009 11:27:59]
- [-] [FirstSpirit.de \(100\)](#) - [09.07.2009 20:08:23 - 06.07.2009 11:13:37]
- [Jüngere Einträge anzeigen](#) - [Ältere Einträge anzeigen](#)
- [+] Thursday, 09.07.2009
- [+] Wednesday, 08.07.2009
- [+] Tuesday, 07.07.2009
- [+] Monday, 06.07.2009
- [-] [Online-Dokumentation FirstSpirit 4.2 \(PRODUKTIV\) \(4\)](#) - [07.07.2009 09:03:18 - 01.07.2009 10:25:22]
- [Jüngere Einträge anzeigen](#) - [Ältere Einträge anzeigen](#)
- [-] Tuesday, 07.07.2009
- [+] generate full (1) - [09:03:18 - 09:05:27] - [Erfolgreich] (OF / OE / OW) - [Historie anzeigen](#) - [Download \(1.208 KB\)](#)
- [-] Friday, 03.07.2009
- [+] generate full (1) - [15:36:34 - 15:38:10] - [Erfolgreich] (OF / OE / OW) - [Historie anzeigen](#) - [Download \(1.208 KB\)](#)
- [-] Wednesday, 01.07.2009
- [+] generate full (1) - [10:57:59 - 10:59:12] - [Erfolgreich] (OF / OE / OW) - [Historie anzeigen](#) - [Download \(1.208 KB\)](#)
- [+] generate full (1) - [10:25:22 - 10:27:12] - [Erfolgreich] (OF / OE / OW) - [Historie anzeigen](#) - [Download \(1.207 KB\)](#)

Abbildung 8-13: 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 Logdatei heruntergeladen werden.



Historie

Online-Dokumentation FirstSpirit 4.2 (PRODUKTIV) > generate full [Jul 7, 2009] > [Wechseln](#)

Level Zeilen





| Tue, 07.07.2009 09:05 bis Tue, 07.07.2009 09:03 | | <input type="button" value="Pause"/> | <input type="button" value="Details"/> | <input type="button" value="<"/> | <input type="button" value=">"/> |
|---|--|--------------------------------------|--|-------------------------------------|-------------------------------------|
|  | INFO [+] 09:05:27 INFO 07.07.2009 09:05:27.334 {seID=82081} (fs.server.scheduler.ScheduleMa [...] | | | | |
|  | INFO [+] 09:05:27 INFO 07.07.2009 09:05:27.229 {seID=82081} (fs.server.sessionmanagement.Se [...] | | | | |
|  | INFO [+] 09:05:27 INFO 07.07.2009 09:05:27.229 {seID=82081} (fs.server.sessionmanagement.Se [...] | | | | |
|  | INFO [+] 09:03:18 INFO 07.07.2009 09:03:18.216 {seID=82081} (fs.server.scheduler.ScheduleMa [...] | | | | |

Abbildung 8-14: Log-Dateien – Historie Veröffentlichung

Über den Verweis [Wechseln](#) hinter dem Projektnamen (inklusive Zeitpunkt der Veröffentlichung) kann ein anderes Projekt bzw. eine andere Veröffentlichung zum gleichen Projekt zur Ansicht ausgewählt werden.

Eine ausführliche Dokumentation zum Aufbau einer historischen Log-Ansicht befindet sich in Kapitel 8.3.1.2 ab Seite 424.

8.3.3.3 Mailversand (ab V4.1)

In FirstSpirit Version 4.1 besteht die Möglichkeit Log-Dateien, z. B. einer Veröffentlichung, innerhalb der Anwendung zum Server-Monitoring herunterzuladen und/oder als Mailanhang zu verschicken. (siehe Kapitel 8.3.1.4 Seite 427).



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 |
|--|--|----------|---------|-------------------|--------------|------------------|
| generate partly | (QA) FirstUnit 4.x (PRODUKTIV) | fest | aktiv | | 0ms | unbekannt |
| Repair references | (QA) FirstUnit 4.x (PRODUKTIV) | fest | aktiv | | 0ms | unbekannt |
| generate full | (QA) FirstUnit 4.x (PRODUKTIV) | fest | aktiv | 07.07.2009 11:53 | 11s | unbekannt |
| Enterprise Backup (Snapshot) | FirstSpirit.de | interval | inaktiv | 16.05.2009 20:50 | 20s | 10.07.2009 20:00 |
| generate partly | FirstSpirit.de | fest | aktiv | 09.07.2009 10:12 | 19s | unbekannt |
| Website Deployment (partial) | FirstSpirit.de | fest | aktiv | 09.07.2009 10:36 | 3s | unbekannt |
| Website Deployment | FirstSpirit.de | täglich | aktiv | 09.07.2009 20:08 | 2m 56s | 10.07.2009 20:08 |
| generate full | FirstSpirit.de | fest | aktiv | 06.07.2009 16:30 | 1m 25s | unbekannt |

Abbildung 8-15: Auftragsübersicht des Servers

In einer Tabelle werden alle auf dem Server eingestellten Aufträge aufgelistet (siehe Kapitel 7.5.1 Seite 343). 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 419). 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 |
|--|--------------------------------|--------|---------------------|---------------------|--------|
| Website Deployment | FirstSpirit.de | aktiv | 09.07.2009 20:08:23 | 09.07.2009 20:11:19 | 2m 56s |
| Website Deployment (partial) | FirstSpirit.de | aktiv | 09.07.2009 10:36:57 | 09.07.2009 10:37:00 | 3s |
| generate partly | FirstSpirit.de | aktiv | 09.07.2009 10:12:54 | 09.07.2009 10:13:13 | 19s |

Abbildung 8-16: 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 430).

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 419)



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, 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 419)

Durch einen Klick auf den Verweis "Historie anzeigen" können die Logausgaben zum Zeitpunkt der Auftragsausführung angezeigt werden:

The screenshot displays the 'Historie' (History) view in the FirstSpirit administration interface. At the top, there are tabs for 'aktuelle Ansicht' and 'Historie'. Below the tabs, the page title is 'Historie'. The breadcrumb navigation shows 'Online-Dokumentation FIRSTspirit 4.0 (PRODUKTIV) > generate full [Jun 28, 2007] > Wechseln'. There are controls for 'Level' (set to 'Debug'), 'Zeilen' (set to '100'), and an 'Anzeigen' button. A search input field contains 'finished' and a 'Suchen' button. Below these controls is a table of log entries for the period 'Thu, 28.06.2007 16:10 bis Thu, 28.06.2007 16:09'. The table has columns for level, time, and message. The entries are:

| Level | Time | Message |
|-------|--------------|--|
| INFO | [+] 16:10:04 | (fs.server.scheduler.ScheduleManagerImpl) finished schedule entry 'generat [...] |
| INFO | [+] 16:09:14 | (fs.server.scheduler.ScheduleManagerImpl) starting task 'generate' - sched [...] |
| INFO | [+] 16:09:14 | (fs.server.sessionmanagement.SessionManagerImpl) dummy session created (ID [...] |
| INFO | [+] 16:09:14 | (fs.server.scheduler.ScheduleManagerImpl) starting schedule entry 'generat [...] |

Below the table, there is a section titled 'Suchergebnis (1)' which shows the search results for the query 'finished':

| Level | Time | Message |
|-------|--------------|--|
| INFO | [+] 16:10:04 | (fs.server.scheduler.ScheduleManagerImpl) finished schedule entry 'generat [...] |

Abbildung 8-17: Zeitplanung – Historie der ausgeführten Aufträge

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 96).

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 **Senden** 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 219), 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 419)

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 32).

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 5.3 Seite 176). Weitere Konfigurationseinstellungen können über die JMX-Konsole vorgenommen werden (siehe Kapitel 9 Seite 430).

Das FirstSpirit Server-Monitoring bietet folgende Konfigurationsmöglichkeiten:

- | | | |
|------------------------|--|---------------------------|
| • Server | (Datei: <code>fs-server.conf</code>) | Kapitel 8.6.1.1 Seite 437 |
| • Lizenz | (Datei: <code>fs-license.conf</code>) | Kapitel 8.6.1.2 Seite 438 |
| • Protokollierung | (Datei: <code>fs-logging.conf</code>) | Kapitel 8.6.1.3 Seite 439 |
| • System | (keine Konfiguration) | Kapitel 8.6.1.4 Seite 440 |
| • Startoptionen | (Datei: <code>fs-wrapper.conf</code>) | Kapitel 8.6.1.5 Seite 441 |
| • Webapplikationen | (Datei: <code>fs-webapps.xml</code>) | Kapitel 8.6.1.6 Seite 442 |
| • Dienste | (Konfig. Systemdienste) | Kapitel 8.6.1.7 Seite 442 |
| • Anmeldekonfiguration | (Datei <code>fs-jaas.conf</code>) | Kapitel 8.6.1.8 Seite 443 |



Diese Menüebene steht nur Serveradministratoren zur Verfügung

8.6.1.1 Server

Die Konfiguration des FirstSpirit-Servers erfolgt über die Konfigurationsdateien `fs-server.conf`, die sich im Installationsverzeichnis des FirstSpirit-Servers befindet (siehe Kapitel 4.3, Seite 33).

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. 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 92 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 96, erläutert.

Standardmäßig werden auftretende Fehler und Infomeldungen dem Loggingsystem "log4j"³² 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.

³² Weiterführende Informationen <http://logging.apache.org/log4j/docs/index.html>



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 11.0-b15 |
| Java Runtime | Java(TM) SE Runtime Environment 1.6.0_10-b33 |
| Betriebssystem | Windows XP (Service Pack 2) x86 5.1 |
| Prozessoren | 2 |
| Aktuelle Zeit | Mittwoch, 08.07.2009 09:59:42 |
| Arbeitsordner | D:\FIRSTspirit4 |

Systemeigenschaften

| | |
|---|--------------------------|
| awt.toolkit | sun.awt.windows.WToolkit |
| backup_files | 50 |
| cmsroot | D:\FIRSTspirit4 |
| com.sun.management.jmxremote | |
| com.sun.management.jmxremote.authenticate | false |
| com.sun.management.jmxremote.port | 50000 |

Abbildung 8-18: 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 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 64, 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.



Bei allen Änderungen der Konfigurationsdatei `fs-wrapper.conf` ist ein Neustart des Servers erforderlich. (Der Neustart des Servers kann in diesem speziellen Fall nicht über die GUI des Server-Monitorings gesteuert werden, sondern muss über die Kommandozeile erfolgen.)



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 99, 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 444). 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 7.3.14 Seite 251.)

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 aktivieren" innerhalb der Server- und Projektkonfiguration verwendet werden (siehe Kapitel 7.3.14 Seite 251).

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 Hier kann ein Verzeichnis zum jeweiligen Dienst angegeben werden. Das Verzeichnis wird im Dateisystem unterhalb des Modulverzeichnisses mit dem eingegebenen Namen angelegt.

Datei erstellen Hier 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 76, 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 Wartungs-Modus

Die Funktion dient zum Aktivieren bzw. Deaktivieren des Wartungs-Modus des FirstSpirit Servers. Wird der Wartungsmodus aktiviert, können sich keine neuen Benutzer am FirstSpirit-Server anmelden. Bei der Aktivierung wird automatisch die Nachricht an alle aktuell eingeloggten FirstSpirit Benutzer geschickt, dass der Server heruntergefahren wird. Alle Bearbeiter erhalten so die Gelegenheit, aktuelle Änderungen zu speichern und sich abzumelden.

Erst wenn alle Benutzer vom Server abgemeldet sind (siehe Kapitel 8.5 Seite 436), sollte der Server gestoppt werden (siehe Kapitel 3 Seite 22).



Die Funktion steht zurzeit noch nicht zur Verfügung und wird voraussichtlich mit einer späteren FirstSpirit-Version ausgeliefert.

8.6.2.2 Web-Applikationen

Mithilfe dieser Funktion können die FirstSpirit Web-Applikationen gesteuert werden.



Die Funktion steht zurzeit noch nicht zur Verfügung und wird voraussichtlich mit einer späteren FirstSpirit-Version ausgeliefert.

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 442).

8.6.2.3 Aktualisierung

Die Funktion dient zur Aktualisierung des FirstSpirit-Servers.



Die Funktion steht zurzeit noch nicht zur Verfügung und wird voraussichtlich mit einer späteren FirstSpirit-Version ausgeliefert.

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 7.3.14 Seite 251). 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 7.3.14 Seite 251).

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

Durch einen Klick auf die Schaltfläche **Server neustarten** wird der Server neu gestartet.



Es erfolgt keine Sicherheitsabfrage vor dem Serverneustart. Nach dem Neustart sollte unbedingt das Server-Monitoring beendet werden.

8.6.2.6 AppCenter Lizenzen

Hier befindet sich eine Liste aller verwendeten AppCenter Lizenzen

The screenshot shows the FirstSpirit Admin interface. At the top, there's a navigation bar with 'Sprache' (language dropdown), 'Benutzer Admin', and 'Monitoring beenden'. A status box on the right shows 'FirstSpirit 5.0_DEV.2.44624', 'Projekte 74', 'Sitzungen 2', 'Aktionen 0', and 'Vorschau 0'. Below the navigation bar, there are tabs for 'Wartungs-Modus', 'Web-Applikationen', 'Aktualisierung', 'Dienste', and 'Server Neustart'. The 'AppCenter Lizenzen' tab is active. The main content area is titled 'AppCenter Lizenzen' and contains a section 'Verwendete AppCenter-Lizenzen' with a list of three items:

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

At the bottom of this list, there is a button labeled 'Verwendungen zurücksetzen'.

Abbildung 8-19: Server-Monitoring – AppCenter-Lizenzen

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.

8.6.3 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.

Nachrichtentext:

Abbildung 8-20: Nachricht senden

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.



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 4.5.2, ab Seite 111.

8.6.4.1 Übersicht

Übersicht über alle aktuell konfigurierten Datenbankanbindungen auf dem FirstSpirit Server.

derby_project6135_3

```
jdbc.layerclass = de.espirit.ormapper.or.layer.DerbyLayer
jdbc.PASSWORD   = p39137813
jdbc.URL        = jdbc:derby:projects/project_6135/derby;create=true
jdbc.USER       = user3
jdbc.POOLMAX    = 1
jdbc.POOLMIN    = 1
jdbc.DRIVER     = org.apache.derby.jdbc.EmbeddedDriver
```

Abbildung 8-21: Überblick Datenbank



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 75. Änderungen an der Konfigurationsdatei `fs-database.conf` können über die FirstSpirit Server- und Projektkonfiguration vorgenommen werden (siehe Kapitel 7.3.6 Seite 238).



Die Parameter `PASSWORD` und `USER` werden nur Serveradministratoren angezeigt.

8.6.4.2 Status

Weitere Informationen zu einer angebundenen Datenbank zeigt der Menüeintrag "Status" an. Unterhalb der angezeigten Datenbankverbindung, werden die Projekte angezeigt, die auf diese Datenbank zugreifen.



Die Funktion steht zurzeit noch nicht zur Verfügung und wird voraussichtlich mit einer späteren FirstSpirit-Version ausgeliefert.

Über den Verweis auf das Projekt unterhalb des Datenbank-Layers kann die Projekt-Detail-Seite aufgerufen werden (siehe Kapitel 8.2.1.1 Seite 419).



8.6.5 FirstSpirit – Monitoring

8.6.5.1 Speicher

Auf dieser Seite wird eine Grafik über den Speicherverbrauch der letzten 24 Stunden angezeigt.

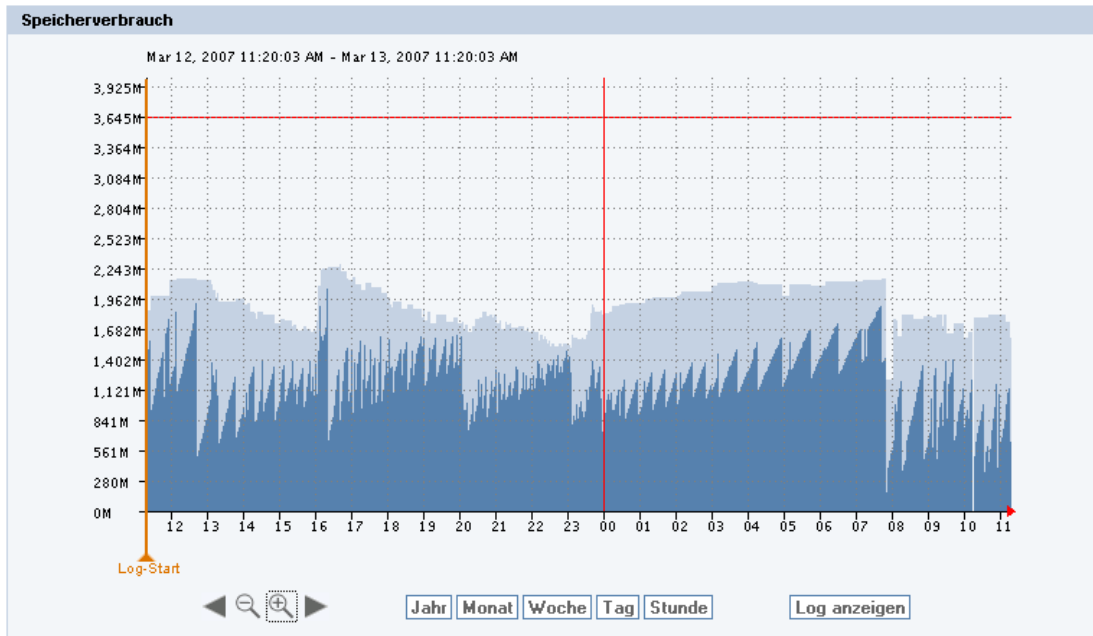






Abbildung 8-22: Monitoring – Speicherverbrauch

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 429)



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.

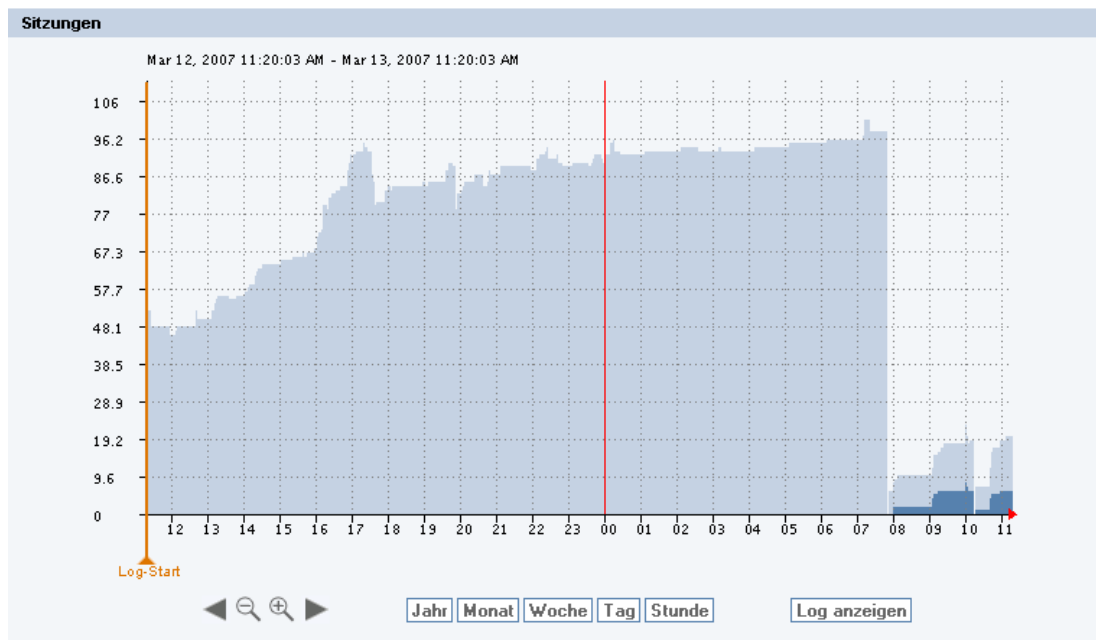


Abbildung 8-23: Monitoring – Sitzungen

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 429)



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.

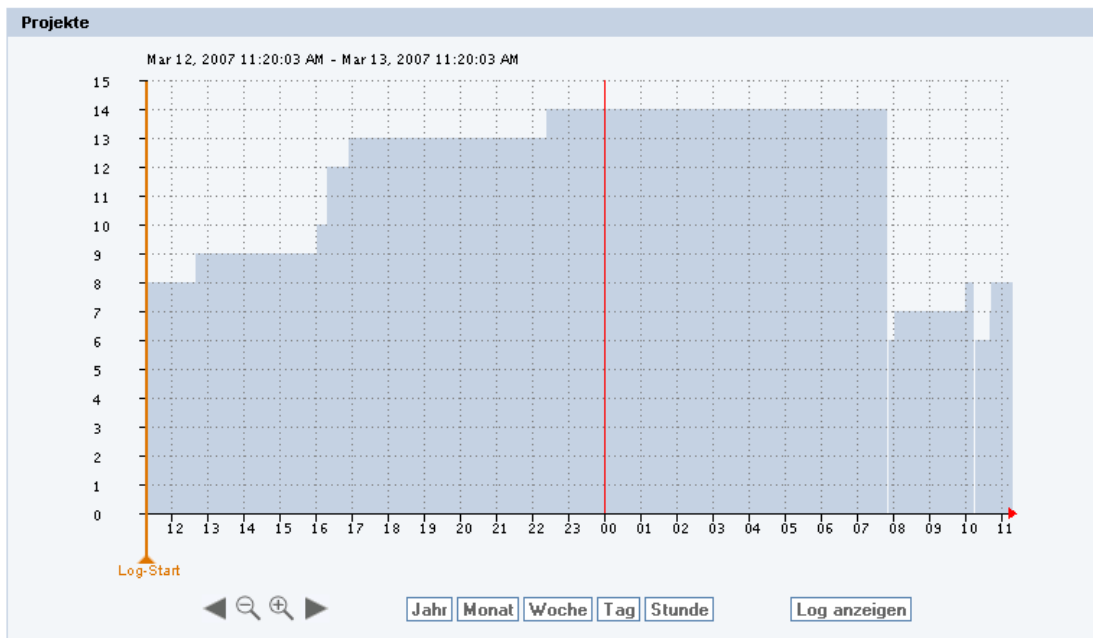






Abbildung 8-24: Monitoring – Projekte

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 429)



8.6.5.4 Ressourcen

Auf dieser Seite befindet sich eine Übersicht über die zur Verfügung stehenden und bereits verwendeten Speicherkapazitäten des Servers.

Ressourcen

Benutzer Speicher: 508 MB
 Allozierter Speicher: 795 MB
 Benutzter Festplattenspeicher: 1628 MB
 Freier Festplattenspeicher: 49521 MB
 Größe Export-Verzeichnis: 539 MB
 Größe Backup-Verzeichnis: 0 MB


| Projektname | ID | Benötigter Plattenplatz | Quota |
|-------------------------------------|-------|-------------------------|-------|
| Testprojekt | 9507 | 1 MB | -1 |
| FIRSTunit 4.0 (PRODUKTIV) | 9896 | 9 MB | -1 |
| FIRSTspirit 4.0 Quality Assuranc... | 4930 | 15 MB | -1 |
| ODFS-TEST | 11591 | 212 MB | -1 |
| Online-Dokumentation FIRSTspirit... | 5 | 274 MB | -1 |

Abbildung 8-25: Monitoring – Ressourcen

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 in 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 im folgenden Fenster 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.

Trace **Topline** **Lock** **Text**

| | Anzahl |
|--|--------|
| [-] "pool-2-thread-7033" prio=5 [] | 21 |
| Stacktrace des Threads | |
| <pre> java.lang.Thread.State: TIMED_WAITING - waiting on <0x1749c47> (a java.util.concurrent.locks.AbstractQueuedSynchronizer\$ConditionObject) at sun.misc.Unsafe.park(Native Method) at java.util.concurrent.locks.LockSupport.parkNanos(LockSupport.java:198) at java.util.concurrent.locks.AbstractQueuedSynchronizer\$ConditionObject.awaitNanos(AbstractQueuedSynchronizer\$ConditionObject.java:395) at java.util.concurrent.LinkedBlockingQueue.poll(LinkedBlockingQueue.java:395) at java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:944) at java.util.concurrent.ThreadPoolExecutor\$Worker.run(ThreadPoolExecutor.java:906) at java.lang.Thread.run(Thread.java:619) </pre> | |
| [+] "SessionScavenger" prio=5 [] | 3 |
| [+] "FSAppender.flushTimer" prio=5 [] | 3 |
| [+] "Attach Listener" prio=5 [] | 3 |
| [+] "Connection.PING" prio=5 [] | 2 |

Abbildung 8-26: 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.

Thread Dumps im Abstand von Minuten

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:

Threads

```

ID                3
Name              Finalizer
Status            WAITING
Lock              java.lang.ref.ReferenceQueue$Lock@ee75b7
Owner             -
In Native         false
Suspended         false
Blocked Count     397
Waited Count      292
    
```

Abbildung 8-27: 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.

VM Speicher

| Name | Typ | Max | Genutzt | Initial | Verpflichtet |
|-------------------|----------|------------|-----------|----------|--------------|
| Code Cache | NON_HEAP | 48 MB | 2,573 MB | 2,25 MB | 2,656 MB |
| PS Eden Space | HEAP | 24,125 MB | 1,339 MB | 3 MB | 9 MB |
| PS Survivor Space | HEAP | 2 MB | 1,912 MB | 512 KB | 2 MB |
| PS Old Gen | HEAP | 227,563 MB | 10,994 MB | 4 MB | 18,125 MB |
| PS Perm Gen | NON_HEAP | 64 MB | 28,692 MB | 16 MB | 31,375 MB |
| Total | TOTAL | 365,688 MB | 45,509 MB | 25,75 MB | 63,156 MB |

Abbildung 8-28: Monitoring – Speicher Java VM (Tabelle)



Diese Menüebene steht nur Serveradministratoren zur Verfügung.



Tabelle:

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.



Grafische Darstellung:



Abbildung 8-29: Monitoring – Speicher Java VM (Graph)

8.6.5.7 JMX-Werte

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 460).

Einige Informationen stehen als grafische Darstellung auch über das den Eintrag JMX-Werte 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 9-13)(siehe Kapitel 9.8 Seite 473).

TasksRunning: Anzahl der momentan laufenden Aktionen (beispielsweise ein Indizierungsjob oder ein Event) (siehe Kapitel 9.8 Seite 473).

TasksQueued: Wird intern eine Aktion angestoßen, beispielsweise eine Indizierung, dann wird der entsprechende Task zuerst in eine Warteschlange gestellt (Ausnahme: hochpriorisierte Tasks) (vgl. Abbildung 9-13). Der hier dargestellte Wert gibt die Anzahl der momentan zwischengespeicherten Aktionen wieder (siehe Kapitel 9.8 Seite 473).

ExecutionRate: Anzahl der ausgeführten Tasks, innerhalb der letzten 60 Sekunden.

AveragePreviewCount: Anzahl der Seiten bzw. Seitenreferenzen, für die innerhalb der letzten 60 Sekunden eine neue Vorschau berechnet wurde (siehe Kapitel 9.11 Seite 479).



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-Werte abweichen.

8.6.6 FirstSpirit – Clustering (ab V4.1)

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 7.3.17 Seite 270).

Es kann eine beliebige Anzahl an "Generation Slaves" für die verteilte Bearbeitung der Generierungsaufträge dargestellt werden (siehe Abbildung 8-30).



Clustering

| Server_1 (Server_1:80) | |
|------------------------|------------------------------------|
| Version | FirstSpirit 4.0.53 |
| Server Typ | SLAVE (Generation Slave) |
| Letzter Kontakt | 20.11.2007 11:47:55 |
| Aktueller Zustand | IDLE |
| Auslastung | <input type="text" value="0"/> 0 % |

| Server_2 (Server_2:5051) | |
|--------------------------|------------------------------------|
| Version | FirstSpirit 4.0.53 |
| Server Typ | SLAVE (Generation Slave) |
| Letzter Kontakt | 20.11.2007 11:47:55 |
| Aktueller Zustand | IDLE |
| Auslastung | <input type="text" value="0"/> 0 % |

Abbildung 8-30: Übersicht "Clustering"

Für jeden Generierungs-Server werden folgende Informationen abgebildet:

| Generation_Server (Generation_Server:5051) | |
|--|--|
| Version | FirstSpirit 4.0.53 |
| Server Typ | SLAVE (Generation Slave) |
| Letzter Kontakt | 20.11.2007 13:43:45 |
| Aktueller Zustand | BUSY |
| Auslastung | <input type="text" value="100"/> 100 % |

Abbildung 8-31: Übersicht eines Generierungs-Servers

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 413) 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 zu aktivieren, muss die folgende Systemeigenschaft für den Serverstart gesetzt werden:

```
-Dcom.sun.management.jmxremote
```

Wichtig ist außerdem der Parameter:

```
-Dsun.rmi.dgc.client.gcInterval=3600000
```

Hier wird das Intervall für die RMI-Garbage-Collection auf 1 Stunde gesetzt.

Für den Betrieb der JMX-Konsole in einer Produktionsumgebung sollten immer die



Parameter:

```
-Dcom.sun.management.jmxremote.authenticate=true
```

und gegebenenfalls (bei verschlüsselter Übertragung):

```
-Dcom.sun.management.jmxremote.ssl=true
```

gesetzt werden. Über `jmxremote.authenticate` wird ein Passwort für den JMX-Port vergeben und über `jmxremote.ssl` die SSL-Authentifizierung aktiviert.



Sind diese beiden Parameter deaktiviert (Standardeinstellung), ist der Zugriff auf den JMX-Port nicht geschützt und unberechtigte Benutzer könnten über den JMX-Port den Server beenden.

Beispielkonfiguration:

```
#wrapper.java.additional.12=-Dcom.sun.management.jmxremote  
#wrapper.java.additional.13=-Dcom.sun.management.jmxremote.ssl=true  
#wrapper.java.additional.14=-Dcom.sun.management.jmxremote.authenticate=true  
#wrapper.java.additional.15=-Dcom.sun.management.jmxremote.port=9000
```

Als Port kann jeder freie Port eingetragen werden, vorausgesetzt er wird nicht von einer lokalen Firewall blockiert.

Weiterführende Informationen zur Konfiguration des JMX-Ports mit gesichertem Zugriff befinden sich unter:

<http://java.sun.com/j2se/1.5.0/docs/guide/management/agent.html>

Zur Verbindung mit der Anwendung wird die JConsole über den Kommandozeilenauftrag "jconsole" gestartet:

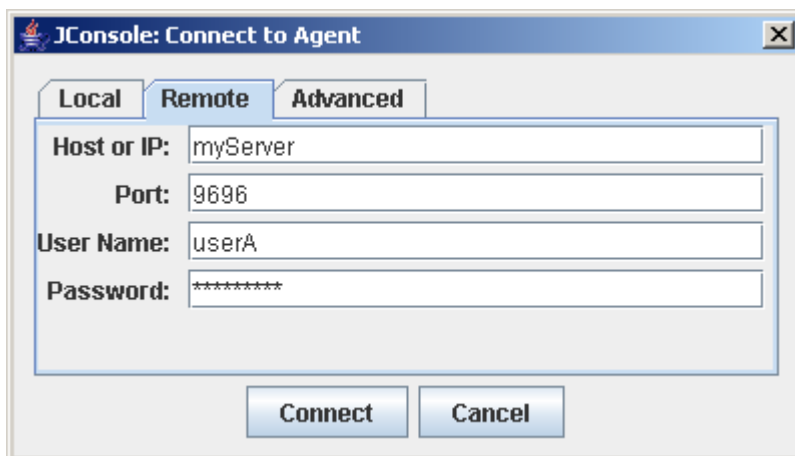


Abbildung 9-1: Starten der JConsole (Remote)

Der Remote-Zugriff wird über die Registerkarte "Remote" gestartet. Hier kann neben dem Hostname auch der JMX-Port der Zielanwendung eingetragen werden. Wenn



die entsprechenden Parameter konfiguriert wurden, 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://java.sun.com>.

Die folgenden Kapitel (Kapitel 9.3 ff.) beschreiben die Möglichkeit zur Überwachung des FirstSpirit Servers, über die durch MBeans zur Verfügung gestellten Informationen. Alle Informationen befinden sich unter der Registerkarte "MBean" (siehe Abbildung 9-2).

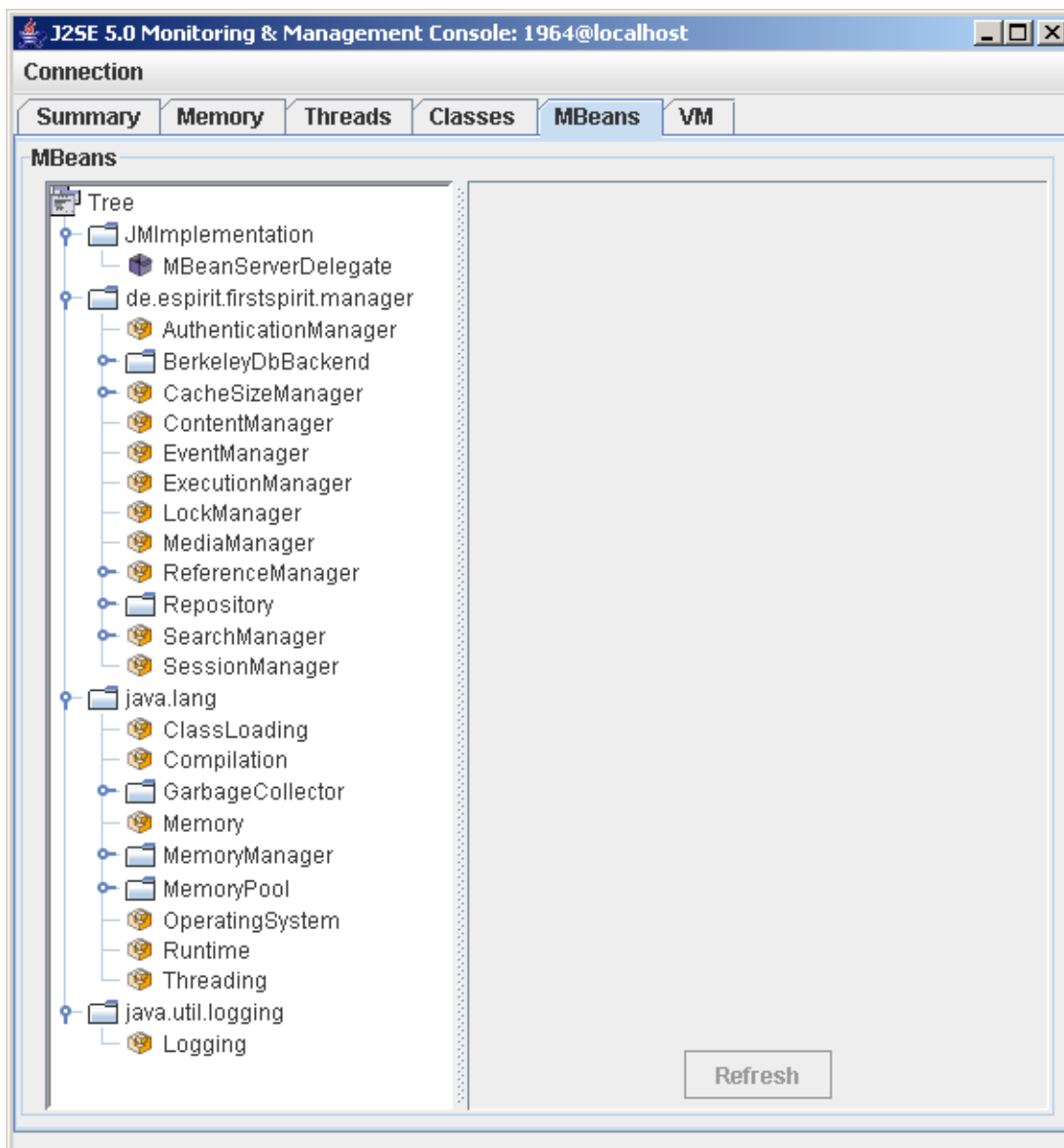


Abbildung 9-2: Registerkarte "MBean" der JMX-Konsole



9.2 Register

Die von FirstSpirit zur Verfügung gestellten MBeans stellen neben den Informationen im Register "Attribute" noch Bearbeitungsmöglichkeiten im Register "Operation" bereit (Beispiel siehe Kapitel 9.4.1 Seite 465 und Kapitel 9.4.2 Seite 466).



Abbildung 9-3: Register der JMX-Konsole

Attributes: Informationen und Statistikwerte, die sowohl tabellarisch (Standardanzeige) als auch graphisch (Umschaltung mit einem Doppelklick auf den entsprechenden Wert) dargestellt werden können (siehe Abbildung 9-12). Mit einem Klick auf den Button "Discard chart" wird die graphische Darstellung wieder geschlossen.

Operations: Operationen, die auf bestimmten Managern aufgerufen werden können, beispielsweise um weitere Statistikwerte anzuzeigen (siehe Abbildung 9-7) oder um bestimmte Aktionen anzustoßen (siehe Abbildung 9-6). Die Operationen werden mit einem Klick auf den entsprechenden Button gestartet.

Info: Allgemeine Informationen zum Manager (Name des Managers und Klasse, die den Manager implementiert).

Die entsprechenden Register werden nur in Managern angezeigt, in denen die entsprechenden Informationen oder Operationen, auch relevant sind.



9.3 AuthenticationManager

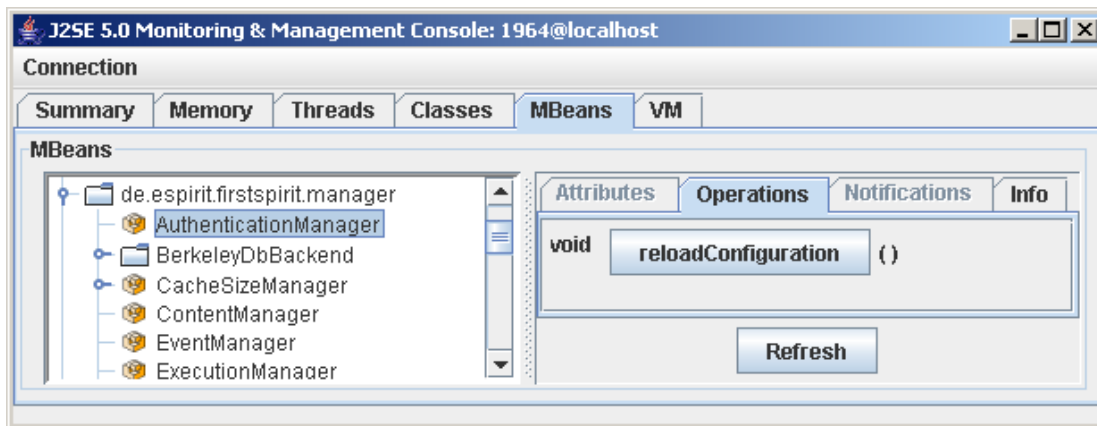


Abbildung 9-4: FirstSpirit JMX-Konsole (Darstellung AuthenticationManager)

Der AuthenticationManager verwaltet alle Anmelde- und Authentifizierungsmöglichkeiten über die Konfigurationsdatei "fs-jaas.conf" (siehe Kapitel 4.3.4 Seite 76). Die Einstellungen für den Anmeldevorgang können über die Server- und Projektkonfiguration bearbeitet werden (siehe Kapitel 7.3.13 Seite 250). In diesem Fall, wird die Konfigurationsdatei neu geschrieben und automatisch neu geladen.



Empfohlen wird immer die Änderung über die Server- und Projektkonfiguration!



9.4 BackendManager

Unter diesem Eintrag werden Informationen zu den angebotenen Backendsystemen dargestellt. Momentan wird als Backendsystem nur Berkeley DB unterstützt, daher wird nur der entsprechende Eintrag in der JMX-Konsole angezeigt. Der BackendManager ist, ebenso wie der Repository (siehe Kapitel 9.13) oder CacheSizeManager (siehe Kapitel 9.5), 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. Dabei besitzt jedes Projekt ein eigenes MBean, das unterhalb des Ordners "BerkeleyDBBackend" angezeigt wird.

9.4.1 Attribute

Der BackendManager enthält für jedes Projekt folgende Attribute:

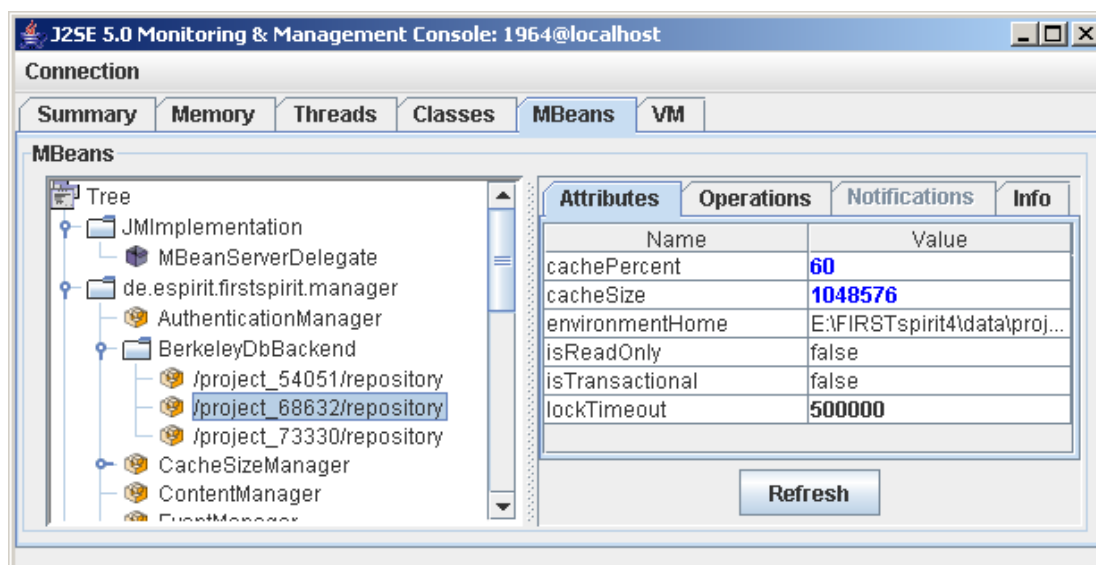


Abbildung 9-5: Backend-Information über die JMX-Konsole (Attribute)

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.

Alle weiteren Attribute werden in der Dokumentation zum entsprechenden Backendsystem genauer erläutert (siehe Dokumentation zur BerkeleyDB³³).

³³ <http://www.oracle.com/database/berkeley-db/index.html>



9.4.2 Operation

Neben den Attributen steht noch die Registerkarte "Operation" zur Auswahl. Über das Register können Operationen ausgeführt und eine Reihe von Statistikinformationen angefordert werden (Informationen siehe Dokumentation zur BerkeleyDB).

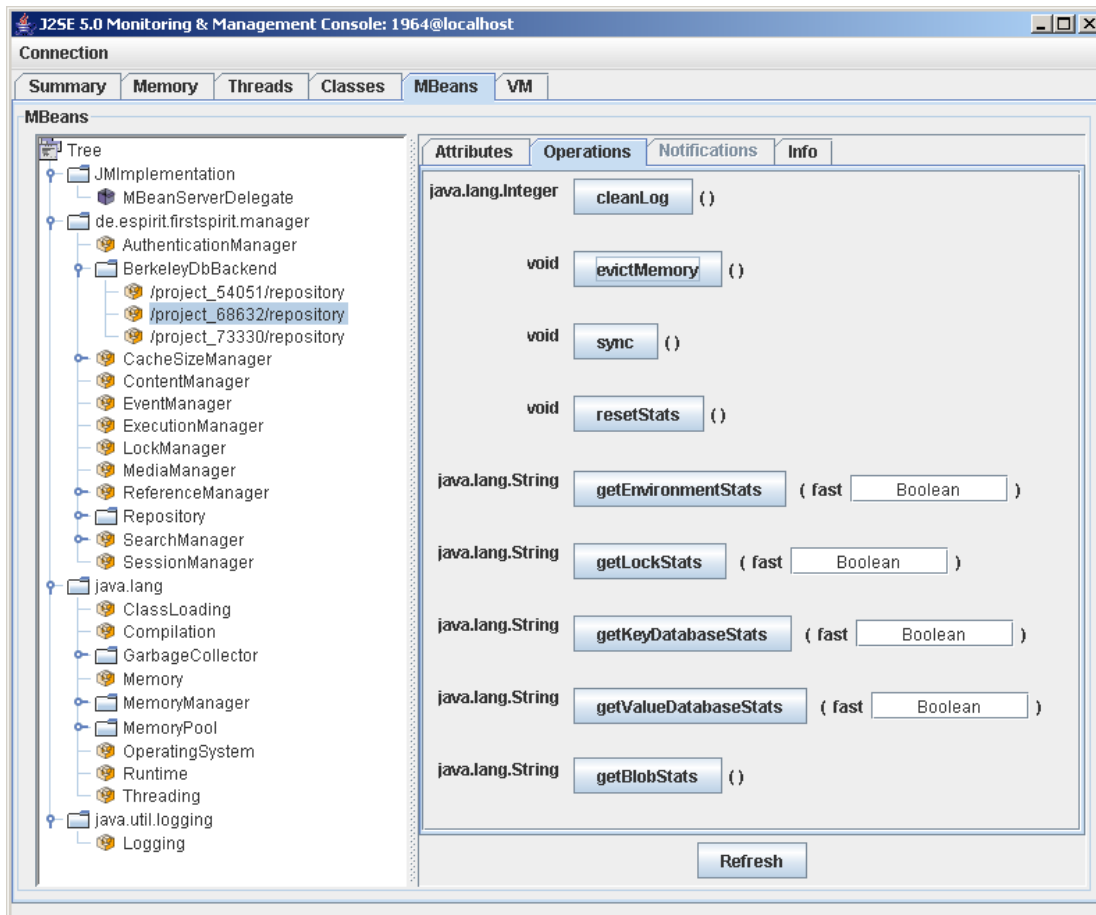


Abbildung 9-6: Backend-Information über die JMX-Konsole (Operations)

Im oberen Bereich des Dialogs werden die Operationen angezeigt, die mit einem Klick auf das entsprechende Feld, beispielsweise auf `cleanLog()` ausgeführt werden können.

Eine häufig benötigte Operation in diesem Bereich ist `sync()` (siehe Abbildung 9-6). Beim Ausführen von `sync()` 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 (siehe Abbildung 9-7):



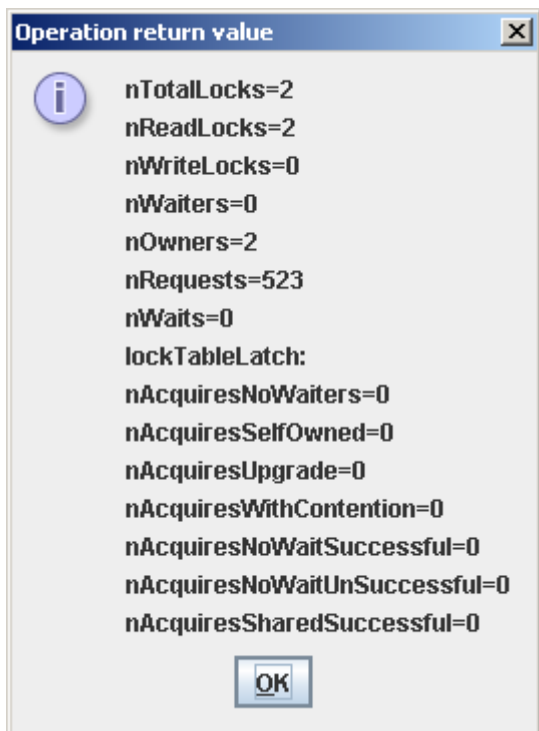


Abbildung 9-7: Statistik-Information über die JMX-Konsole (Operations)



Der Aufruf kann besonders bei vielen Daten einen Moment Zeit in Anspruch nehmen.



9.5 CacheSizeManager

Der CacheSizeManager teilt den zur Verfügung stehenden Hauptspeicher auf die verschiedenen Zwischenspeicher (Caches) auf. Unter diesem Eintrag werden Informationen zur Gesamt-Cache-Größe (CacheSizeManager – vgl. Abbildung 9-8) und Informationen zu den einzelnen Projekt-Caches (MBeans – vgl. Abbildung 9-10) dargestellt. Der CacheSizeManager ist, ebenso wie der RepositoryManager (siehe Kapitel 9.13) 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"). Dabei besitzt jedes Projekt ein eigenes MBean, das unterhalb des "CacheSizeManager" angezeigt wird.

9.5.1 Attribute (gesamt)

Auf Ebene des CacheSizeManagers stehen folgende Attribute zur Verfügung:

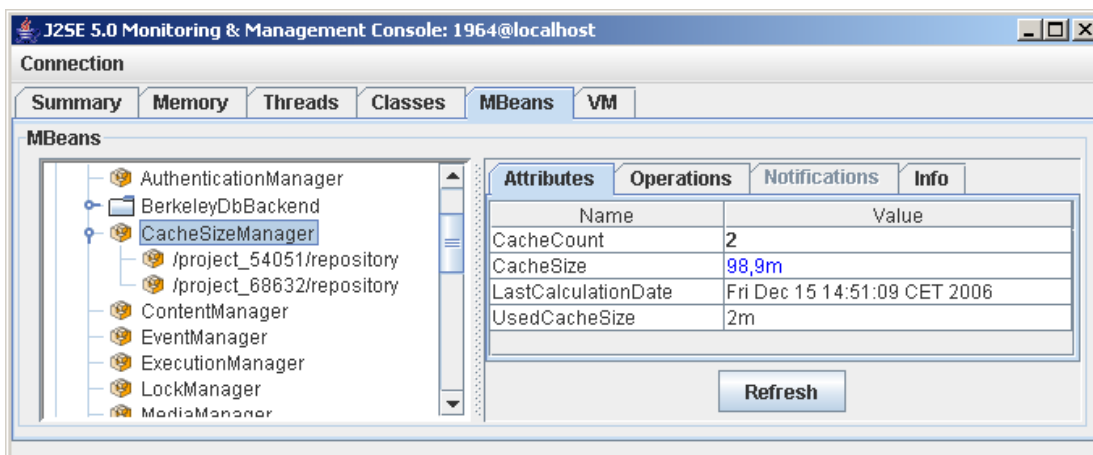


Abbildung 9-8: Gesamt Cache-Information über die JMX-Konsole (Attribute)

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 4.3.1.13 Seite 57) 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.

9.5.2 Operation

Auf Ebene des CacheSizeManagers steht eine Operation zur Verfügung:

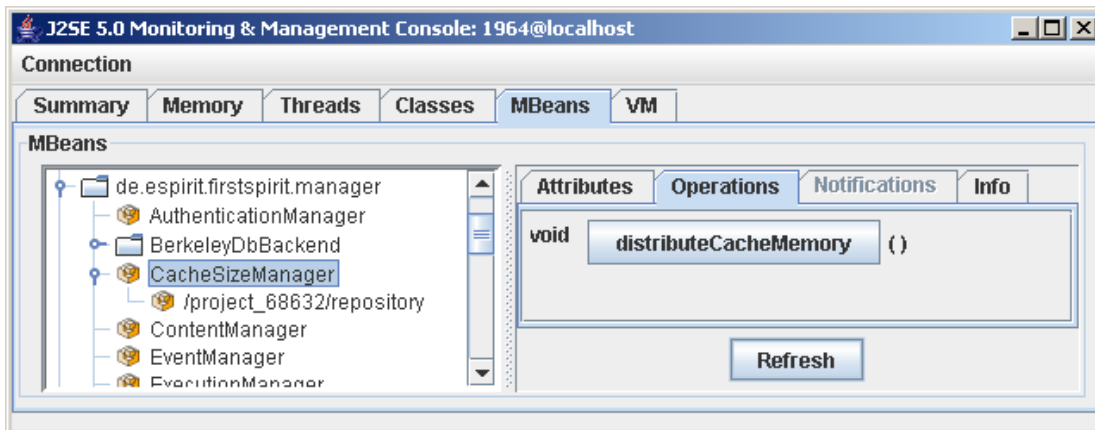


Abbildung 9-9: Gesamt Cache-Information über die JMX-Konsole (Operation)

Bei einem Klick auf `distributeCacheMemory()` wird der zur Verfügung stehende Speicher neu verteilt. Nach der Ausführung wird der unter "LastCalculationDate" gespeicherte Wert angepasst (vgl. 9.5.1).

9.5.3 Attribute (projektbezogen)

Neben den Gesamtinformationen für die CacheSize besitzt jedes Projekt ein eigenes CacheSizeMBean, mit folgenden Attributen:

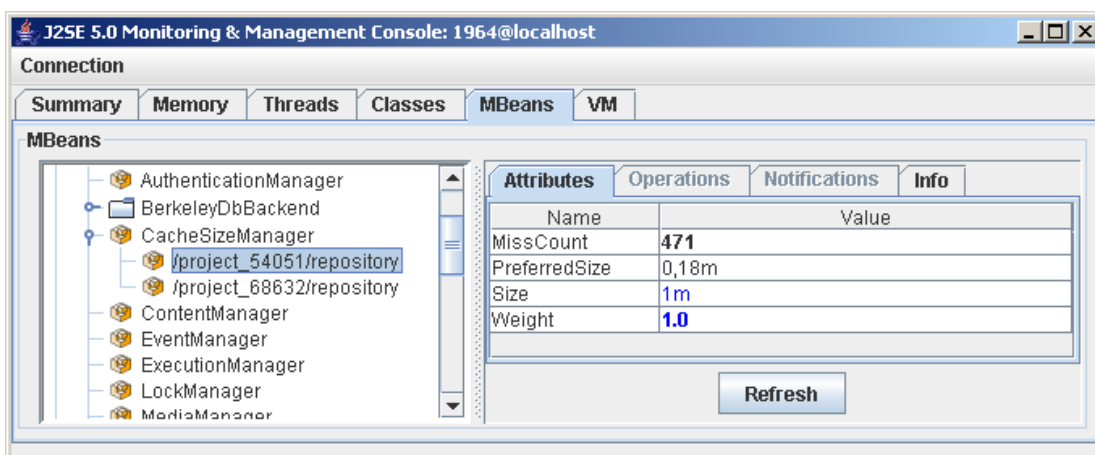


Abbildung 9-10: Cache-Information pro Projekt über die JMX-Konsole (Attribute)

MissCount: Der Zähler erfasst die Anzahl an Anfragen, 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.

9.6 Content Manager

Der ContentManager enthält Informationen zu FirstSpirit Datenbankverbindungen und -zugriffen.

Auf Ebene des ContentManagers stehen folgende Attribute zur Verfügung:

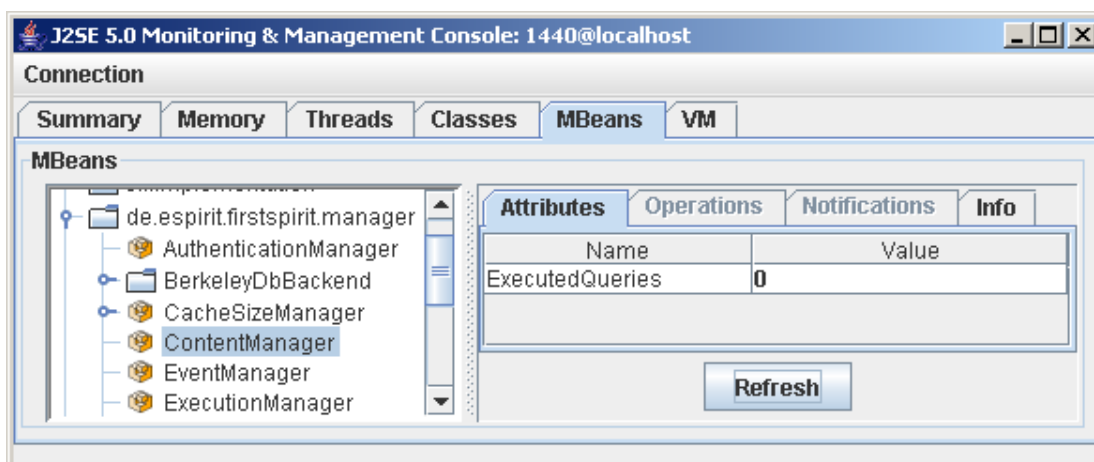


Abbildung 9-11: FirstSpirit JMX-Konsole (Darstellung ContentManager)



ExecutedQueries: Anzahl der ausgeführten Datenbank-SQL-Statements innerhalb der letzten Minute.

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.

Auf Ebene des EventManagers stehen folgende Attribute zur Verfügung:



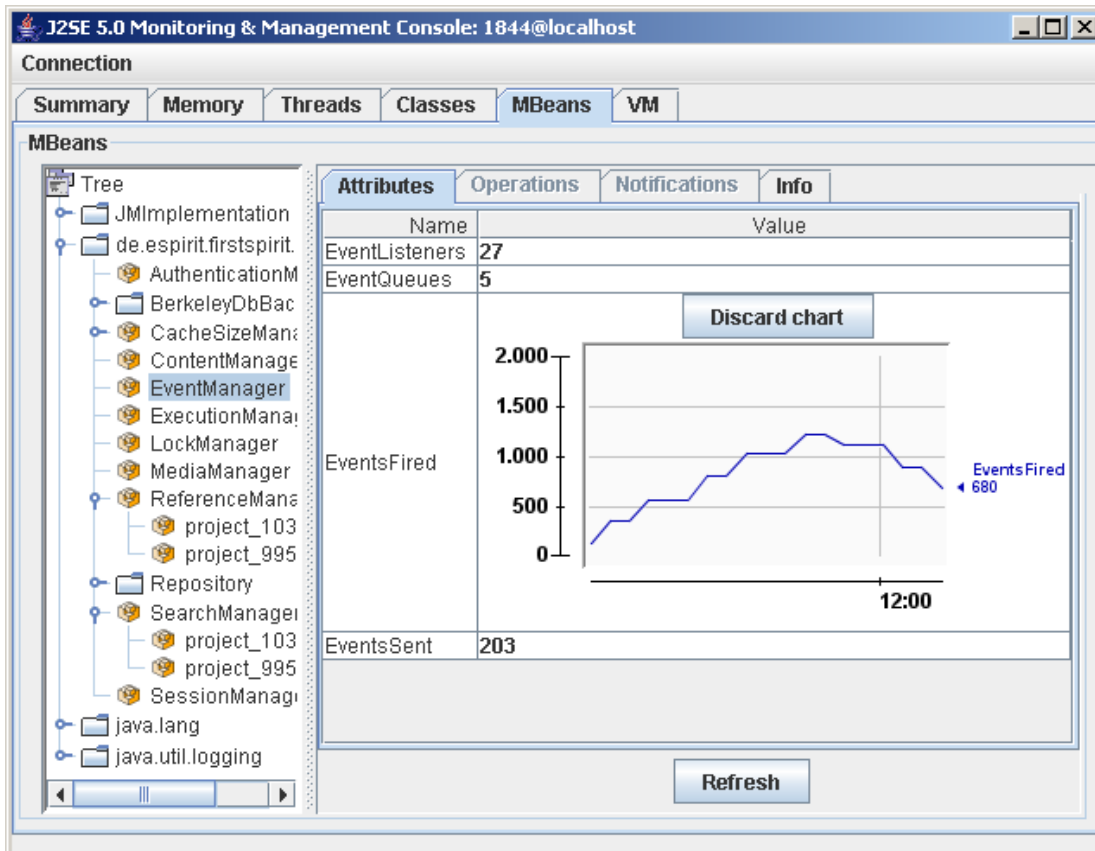


Abbildung 9-12: FirstSpirit JMX-Konsole (Darstellung EventManager)

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. Indirekt gibt dieser Wert Auskunft darüber, wie viele Clients mit dem Server verbunden sind.

EventsFired: Gefeuerte Events der letzten 60 Sekunden.

EventsSent: Tatsächlich gesendete Events der letzten 60 Sekunden.



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 42).

`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 42).

`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 42). Ist die maximale Aufnahmekapazität der Queue erreicht, werden weitere Client-Calls vom Server kurzfristig zurückgewiesen, bevor ein weiterer Versuch gestartet wird.

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 8.6.5.7 Seite 457) zur Verfügung:

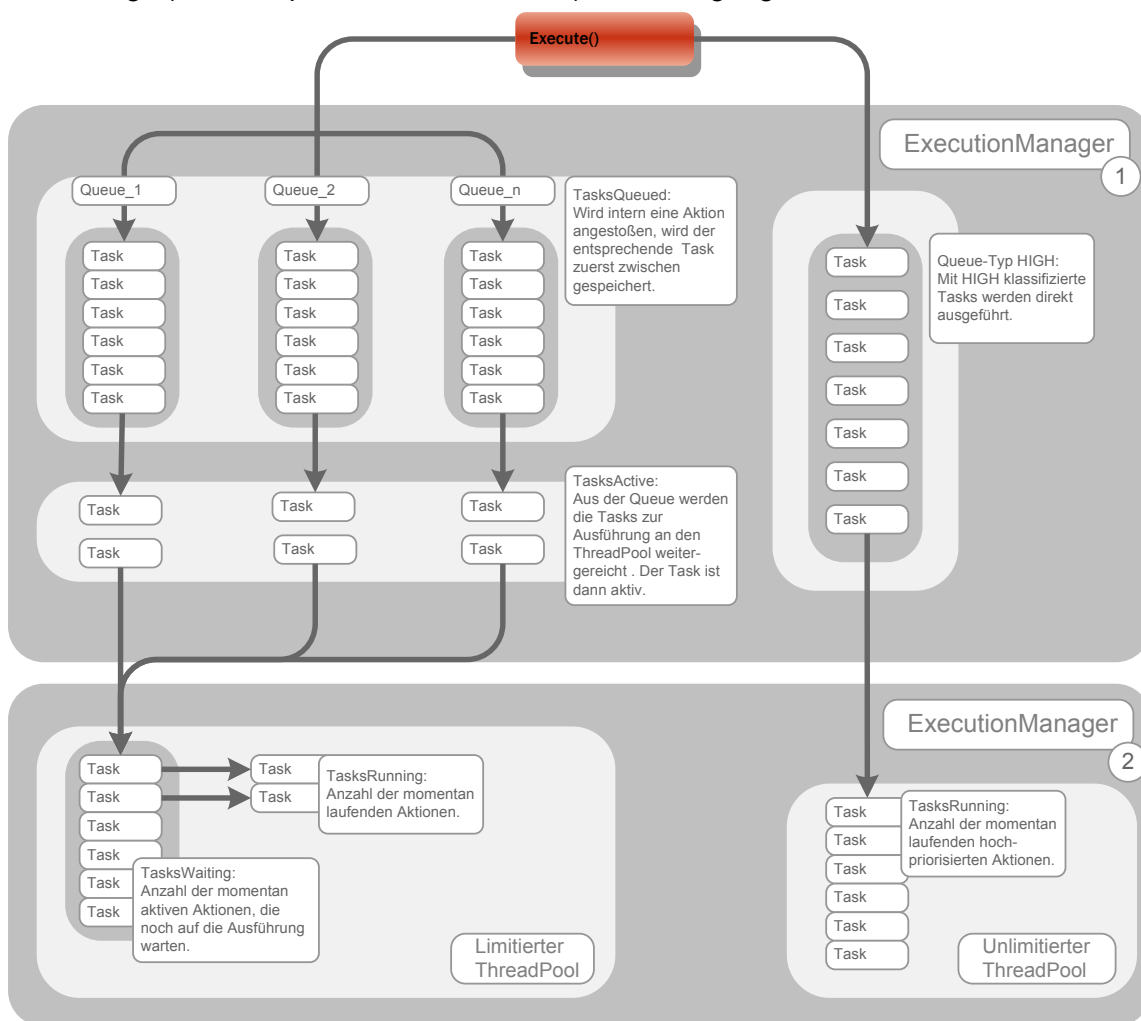


Abbildung 9-13: Ausführung der Tasks über den ExecutionManager



9.8.3 Attribute

Auf Ebene des ExecutionManagers stehen folgende Attribute zur Verfügung:

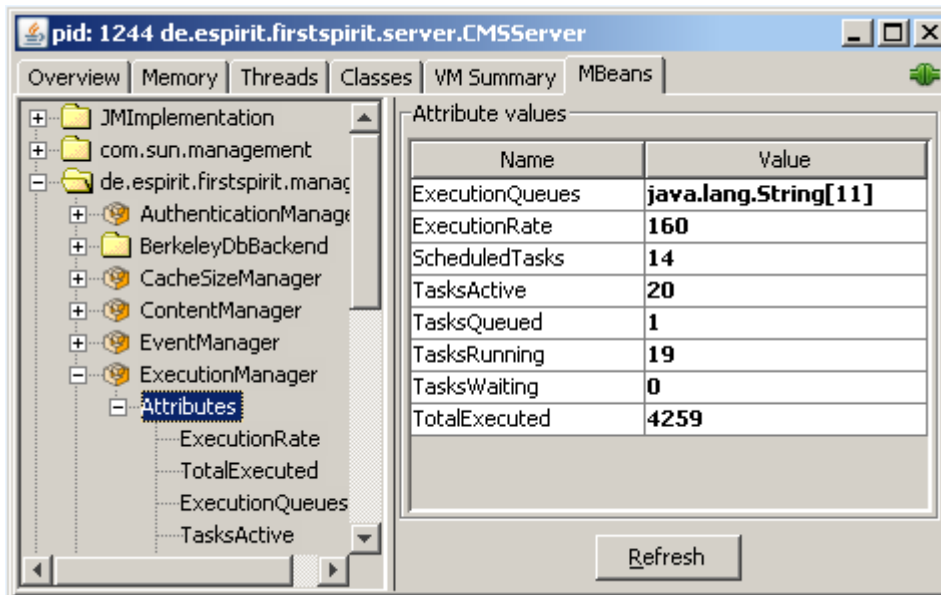


Abbildung 9-14: FirstSpirit JMX-Konsole (Darstellung ExecutionManager)

ExecutionRate: Anzahl der ausgeführten Tasks, innerhalb der letzten 60 Sekunden.

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 9-13). 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 42).

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 9-13).

TasksRunning: Anzahl der momentan laufenden Aktionen (beispielsweise ein



Indizierungsjob oder ein Event) innerhalb des limitierten und unlimitierten ThreadPools (vgl. Abbildung 9-13). Die Anzahl der parallel ausgeführten Tasks (über den limitierten ThreadPool) kann über den Parameter `maxSize` begrenzt werden (siehe Kapitel 4.3.1.4 Seite 41).

TasksWaiting: Anzahl der Aktionen, die bereits aus der entsprechenden Queue weitergeleitet wurden, aber noch auf die Ausführung innerhalb des ThreadPools warten (vgl. Abbildung 9-13).

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 9-15).

```
Queue[8:ReferenceManager:632107,SERIAL] - a:1 w:1
```

```
Queue[7:ReferenceManager:632028,SERIAL] - a:0 w:0
```

```
Queue[6:ReferenceManager:631983,SERIAL] - a:0 w:0
```

```
Queue[5:MediaManager,LOW] - a:0 w:0
```

```
Queue[HIGH]
```

Abbildung 9-15: ExecutionManager – Queues

Jeder Eintrag beinhaltet die folgenden Informationen:

Name der Queue: Der Name setzt sich zusammen aus (siehe Abbildung 9-15):

- 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 473). Einige Queue-Klassen können über die Konfigurationsdatei `fs-server.conf` parametrisiert werden (siehe Kapitel 4.3.1.5, ab Seite 42).

Anzahl der aktiven Threads: Anzahl der aktiven Threads [`a: Anzahl`], die aus der Queue an den ThreadPool weitergereicht werden (vgl. "TasksActive").

Anzahl der wartenden Threads: Anzahl der zwischen gespeicherten Threads [`w: Anzahl`], die aus der Queue an den ThreadPool weitergereicht werden (vgl.



"TasksQueued").

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.

Auf Ebene des LockManagers stehen folgende Attribute zur Verfügung:

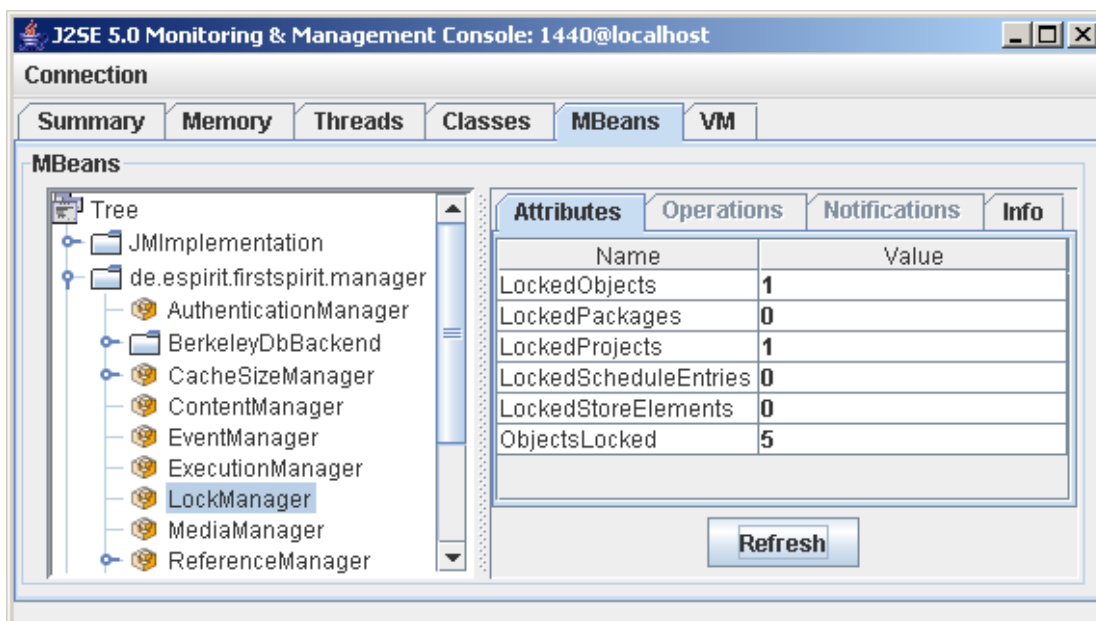


Abbildung 9-16: FirstSpirit JMX-Konsole (Darstellung LockManager)

LockedObjects: Anzahl aller Objekte (Projekte, Elemente, Pakete, Aufträge), die sich momentan in Bearbeitung befinden und 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 Paket-Verwaltung verwendet wird.

LockedProjects: Anzahl aller Projekte, die sich momentan in Bearbeitung befinden und durch einen Benutzer gesperrt wurden (Beispiel siehe Kapitel 7.3.16 Seite 264).

LockedScheduleEntries: Anzahl aller Aufträge, die sich momentan in Bearbeitung



befinden und durch einen Benutzer gesperrt wurden (Beispiel siehe Kapitel 7.5.4 Seite 351).

LockedStoreElements: Anzahl aller Elemente aus FirstSpirit Verwaltungsbereichen (z. B. Inhalte-Verwaltung), 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.

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.

Auf Ebene des MediaManagers stehen folgende Attribute zur Verfügung:

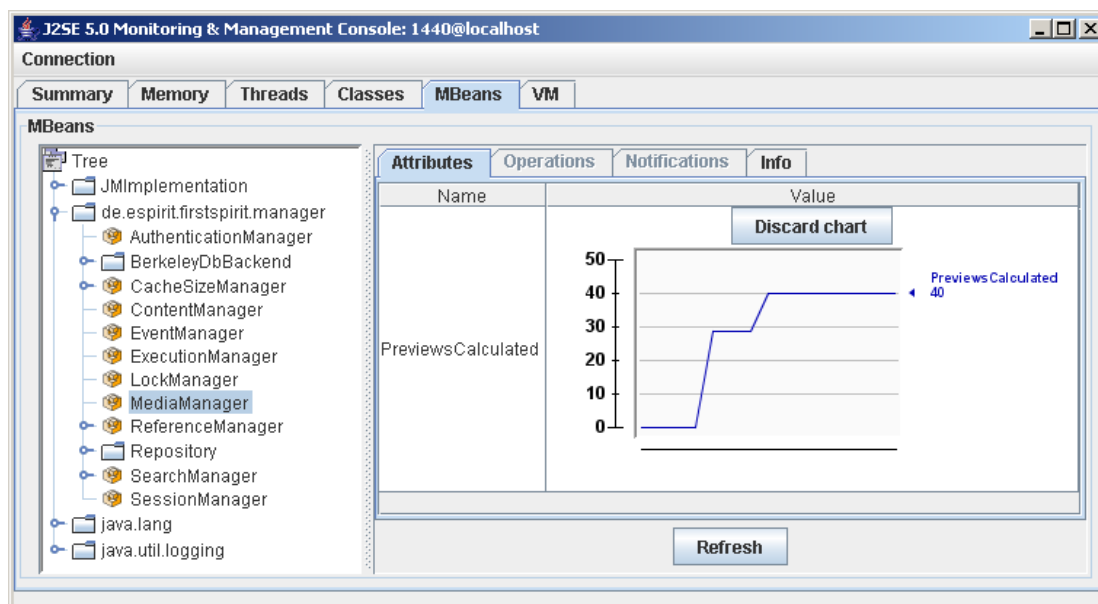


Abbildung 9-17: FirstSpirit JMX-Konsole (Darstellung MediaManager)

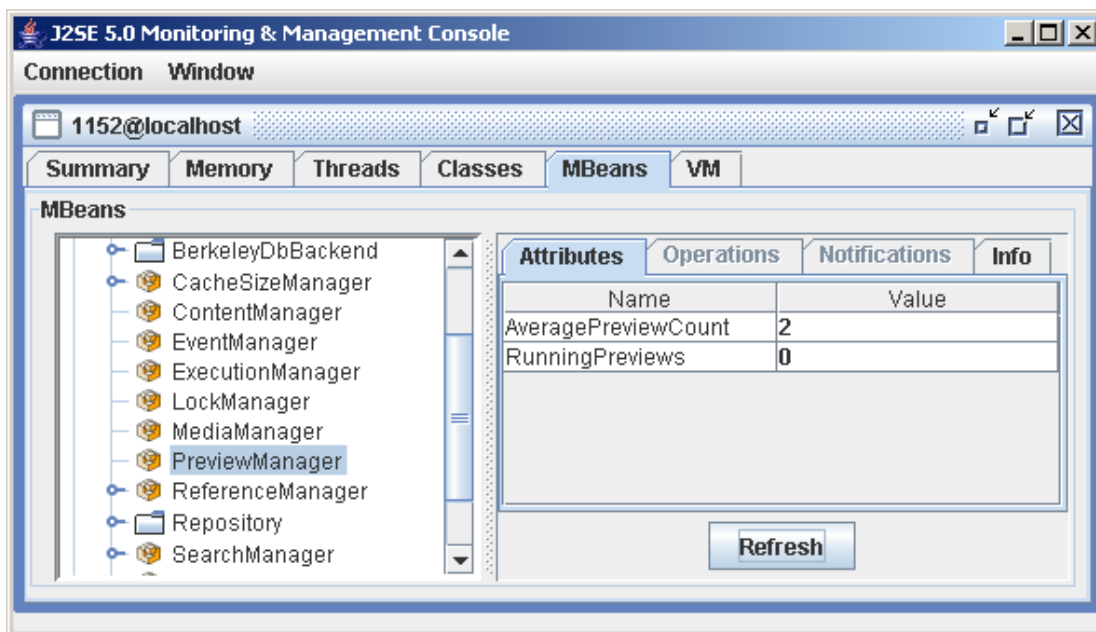
PreviewsCalculated: Anzahl der Medien, für die innerhalb der letzten 60 Sekunden ein neues Vorschaubild (Thumbnail) berechnet wurde.



9.11 PreviewManager

Über FirstSpirit kann jederzeit eine Vorschau der aktuell bearbeiten 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.

Auf Ebene des PreviewManagers stehen folgende Attribute zur Verfügung:



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.



9.12 ReferenceManager

Der ReferenceManager ist für das Berechnen und Ausliefern von Referenzen zuständig.



Neuberechnung und Konvertierung von Referenzen: Für Referenzen wird in Version 4.2R4 ein performanteres Datenformat verwendet. Bestehende Referenzen werden bei einem Upgrade auf 4.2R4 automatisch konvertiert und Neuberechnet. Dies wird beim ersten Öffnen von Projekten, die sich auf einem auf die Version 4.2R4 aktualisierten FirstSpirit-Server befinden, vorgenommen. Die Referenz-Neuberechnung verläuft asynchron. D. h., Server und Projekte können bereits verwendet werden, während die Referenzberechnung noch läuft. Die Konvertierung des Datenformates der Referenzen erfolgt während der Referenz-Neuberechnung.

Auf Ebene des ReferenceManagers stehen folgende Attribute zur Verfügung:

ActiveCalculated: Aktuelle Anzahl aller bereits neu berechneten Projektreferenzen auf dem FirstSpirit-Server (verfügbar ab FirstSpirit-Version 4.2).

ActiveDuration: Aktuelle Dauer der zurzeit laufenden Neuberechnung von Projektreferenzen auf dem FirstSpirit-Server (verfügbar ab FirstSpirit-Version 4.2). Diese Information wird auch im ServerMonitoring im Bereich „Übersicht – Aktivitäten“ eingeblendet (siehe Kapitel 8.1.2 Seite 416).

ActiveQueued: Aktuelle Anzahl der noch zu berechnenden Projektreferenzen auf dem FirstSpirit-Server (verfügbar ab FirstSpirit-Version 4.2).

ActiveRecalculations: Aktuelle Anzahl der Projektreferenzen, die in diesem Moment neu berechnet werden (verfügbar ab FirstSpirit-Version 4.2).

ActiveRepaired: Aktuelle Anzahl der reparierten Projektreferenzen auf dem FirstSpirit-Server (verfügbar ab FirstSpirit-Version 4.2). Diese Information wird auch im ServerMonitoring im Bereich „Übersicht – Aktivitäten“ eingeblendet (siehe Kapitel 8.1.2 Seite 416).

CalculationRate: Anzahl der Referenzen, die innerhalb der letzten 60 Sekunden neu berechnet wurden (Referenzen/Minute) (verfügbar ab FirstSpirit-Version 4.0).

Auf Ebene des ReferenceManagers stehen folgende Operations zur Verfügung:

updateProjectReferences: Die Neuberechnung und Konvertierung von Referenzen, die in einer alten Referenzgraph-Version vorliegen, erfolgt nach einem



Upgrade auf FirstSpirit-Version 4.2R4 automatisch beim ersten Zugriff auf ein Projekt. Ist eine manuelle Aktualisierung gewünscht, kann diese über die Operation `updateProjectReferences` gestartet werden. Diese Operation startet die Neuberechnung aller veralteten Projektreferenzen auf dem FirstSpirit-Server (verfügbar ab FirstSpirit-Version 4.2.433).

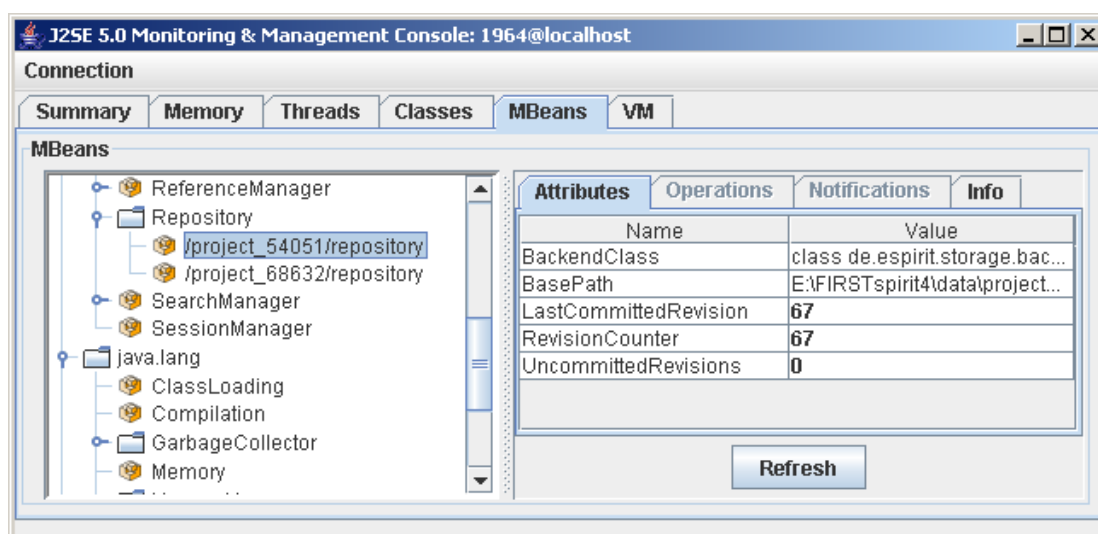
startRecalculation: Diese Operation startet die Neuberechnung der Projektreferenzen, in der jeweils aktuellsten, vorliegenden Revision, für ein bestimmtes Projekt. Über die JConsole kann hier die Projekt-ID des Projektes übergeben werden, für die eine Neuberechnung der Referenzen gestartet werden soll (verfügbar ab FirstSpirit-Version 4.2.433).

Die Neuberechnung der Projektreferenzen kann im ServerMonitoring im Bereich „Übersicht – Aktivitäten“ verfolgt werden (siehe Kapitel 8.1.2 Seite 416).

9.13 RepositoryManager

FirstSpirit setzt zur Archivierung und Versionierung der Projektdaten Repositories ein. Für jedes Projekt existiert ein Repository und für jedes Repository ein MBean im Ordner "Repository" der JMX-Konsole. Der RepositoryManager ist, ebenso wie der Backend (siehe Kapitel 9.4) oder CacheSizeManager (siehe Kapitel 9.5), 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").

Der RepositoryManager enthält für jedes Projekt folgende Attribute:



The screenshot shows the J2SE 5.0 Monitoring & Management Console. The 'MBeans' tab is selected, and the 'Repository' folder is expanded. The 'project_54051/repository' MBean is selected, and its attributes are displayed in a table:

| Name | Value |
|-----------------------|---------------------------------|
| BackendClass | class de.espirit.storage.bac... |
| BasePath | E:\FIRSTspirit4\data\project... |
| LastCommittedRevision | 67 |
| RevisionCounter | 67 |
| UncommittedRevisions | 0 |

A 'Refresh' button is visible at the bottom of the attributes table.

Abbildung 9-18: Repository-Information über die JMX-Konsole (Attribute)



BackendClass: Klasse, die das entsprechende Repository implementiert (momentan wird nur BerkeleyDB unterstützt).

BasePath: Vollständiger Pfad zum Repository des Projekts.

LastCommittedRevision: 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.

9.14 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 Search-Index geschrieben bzw. dort wieder gelöscht.

Auf Ebene des SearchManagers stehen folgende Attribute zur Verfügung:

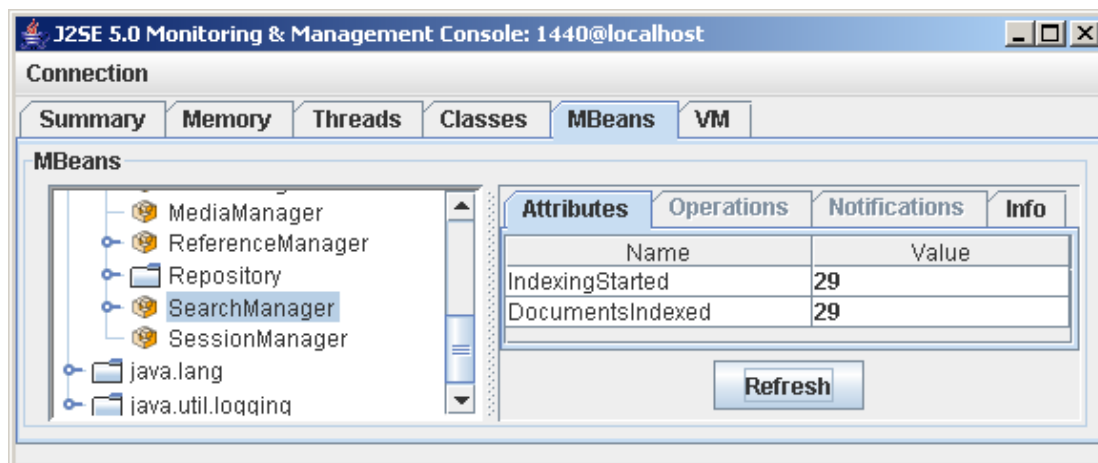


Abbildung 9-19: FirstSpirit JMX-Konsole (Darstellung SearchManager)

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).





Volltext-Indizierung der Medien-Verwaltung: Mit der Version 4.0 werden Dokumente aus der FirstSpirit Medien-Verwaltung indiziert, um eine Volltextsuche zu ermöglichen. Aus technischen Gründen werden dabei nicht alle Dokumenttypen unterstützt. Eine (vollständige) Indizierung ist daher nicht in allen Fällen möglich.



9.15 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 473).

Auf Ebene des SessionManagers stehen folgende Attribute zur Verfügung:

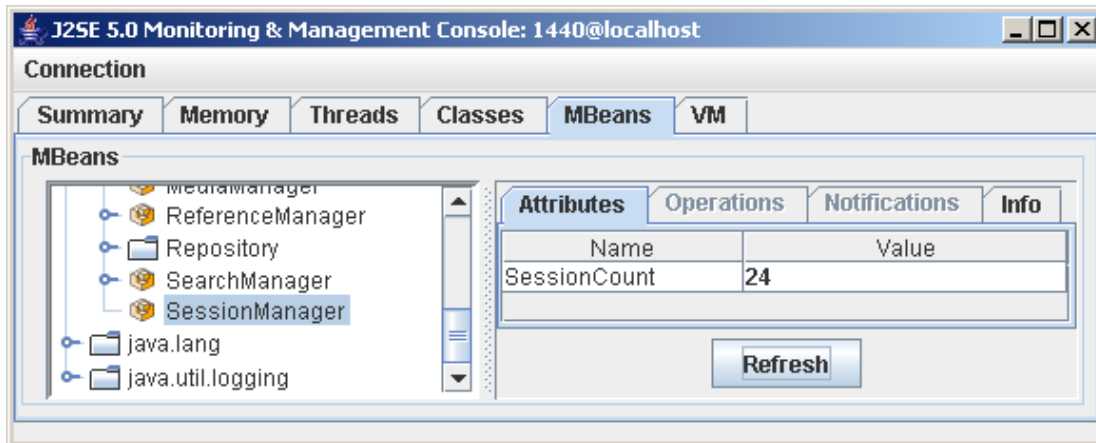


Abbildung 9-20: FirstSpirit JMX-Konsole (Darstellung SessionManager)

SessionCount: Anzahl der aktuell gültigen Sessions, innerhalb der letzten 60 Sekunden. Temporäre Sessions, das heißt vom Server angelegte Sessions, werden nicht angezeigt.



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`³⁴ und `ssh`³⁵.

`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

³⁴ `rsync`: <http://rsync.samba.org/>

³⁵ `ssh`: <http://www.openssh.com/>



darauf zu achten, dass der ssh-Client das ssh-Protokoll 2 nutzt. Unter Windows kann der ssh-Client `putty.exe` verwendet werden.

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 Web-Browser 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 Webservers 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 (fs4) anmelden bzw. als root "su - fs4" 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 Webservers installieren. Dazu über ssh als "web" anmelden und das Verzeichnis `.ssh` im Home-Directory erstellen: `mkdir .ssh`
Auf dem FirstSpirit-Server als Benutzer "fs4" 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 Webservers aufgelistet werden.
6. Zum Testen der rsync-Verbindung eingeben:
`rsync -n -vrtz -e "ssh -l web" www.mydomain.net:/var/www`
7. Ohne Eingabe eines Passworts sollte das Dokumentenverzeichnis des Webservers 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 Cygwin³⁶ 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`

³⁶ <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

- 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
```

- 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
```

- Ohne Eingabe eines Passworts sollte nun das Dokumentenverzeichnis des Webservers aufgelistet werden.

- Zum Testen der rsync-Verbindung nun eingeben:

```
c:\cygwin\bin\rsync -n -vcrtz -e "c:\cygwin\bin\ssh -l web" www.mydomain.net:/var/www
```

- Ohne Eingabe eines Passworts sollte das Dokumentenverzeichnis des Webservers 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.



10.5 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 351).
2. Zu dem Auftrag die Aktion "Projekt generieren" hinzufügen (siehe Kapitel 7.5.9.2 Seite 368).
3. Zu dem Auftrag die Aktion "Skript ausführen" unter der vorherigen Aktion hinzufügen (siehe Kapitel 7.5.9.4 Seite 381).
4. Das "ssh/rsync-Deployment-Skript" befindet sich im Anhang (siehe Kapitel 12.5 Seite 528).

5. Als Skript-Parameter werden die Parameter des Webserver 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 381). Erst wenn der Test fehlerfrei durchläuft, kann diese Veröffentlichung verwendet werden.



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), so gelten die Benutzerrechte für den "Besucher" der generierten Site und sind damit immer mit dem eingesetzten Personalisierungssystem verknüpft. Wird als Personalisierungssystem FirstSpirit Personalisation³⁷ eingesetzt (was nicht zwangsläufig der Fall sein muss), so kann ein sehr enger Zusammenhang hergestellt werden. (siehe Kapitel 11.2.3 Seite 501)

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. Ob ein Projekt mit Benutzerrechten arbeitet erkennt man in der Regel am Einsatz einer Login-Seite.

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

³⁷ Siehe Modul-Dokumentation zu FirstSpirit Personalisation



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-Site³⁸:
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 werden, das bedeutet, der Benutzer meldet sich als Redakteur an bzw. umgekehrt.

In FirstSpirit Version 4.0 wurden außerdem folgende Punkte realisiert³⁹:

- SSO: Bei Verwendung des SSO-Moduls erfolgt der Übergang zwischen Redaktion und Benutzer transparent ohne Passwort-Abfrage (Login-Modul: FS SSO)
- Wählbare Benutzer-Perspektive für den Redakteur: Um überprüfen zu können, ob eine Benutzerrechte-Konfiguration sich so verhält wie vom Redakteur gewünscht, ist der einfache Wechsel der Benutzer-Berechtigung wünschenswert. Dies ist durch die Konfiguration eines speziellen Login-Servlets für das Preview möglich, über das eine explizite Gruppenauswahl erlaubt wird.

³⁸ Siehe Modul-Dokumentation FirstSpirit Integration

³⁹ Siehe Modul-Dokumentation FirstSpirit Personalisation



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 Unterstützung, um diese Gruppenstrukturen entweder neu zu definieren oder aber aus einem bestehenden System (z. B. LDAP) zu übernehmen.

Unabhängig von der Herkunft wird die Gruppenhierarchie dem Redakteur in Form einer Baumansicht präsentiert, in der die Rechtekonfiguration vorgenommen werden kann. Bei der Rechtedefinitionskomponente handelt es sich also um eine spezielle Eingabekomponente, mit deren Hilfe man Rechte auf der Basis einer hierarchischen Gruppendifinition vergeben kann. Diese Rechtedefinition bezieht sich ausschließlich auf das Laufzeitsystem und nicht auf das Reaktionssystem selbst, es handelt sich also nicht um Redaktionsrechte. In der Regel wird daher die Rechtedefinitionskomponente im Rahmen der Metadaten verwendet, ein Einsatz in der Inhalteverwaltung 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 Personalisation 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-Personalisation-Tags⁴⁰) 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 504). 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" Konzeptes vorhanden (siehe Kapitel 11.3.3 Seite 506). Mehr zum Thema "Schutz personalisierter Inhalte" siehe Kapitel 11.3 Seite 502.

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 504). 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 505).

Im Detail werden diese Mechanismen in Kapitel 11.2.3 ab Seite 501 beschrieben.

⁴⁰ Siehe Modul-Dokumentation FirstSpirit Personalisation



11.1.3 Permission-Modul

Die Benutzerrechte-Vergabe und -Prüfung in FirstSpirit erfolgt über das Permission-Modul. Das Modul besteht aus zwei Komponenten:

- Editor (Rechtedefinitions-Komponente) – (siehe Kapitel 11.1.3.1 Seite 495)
- Service (PermissionService) – (siehe Kapitel 11.1.3.2 Seite 495)

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 "FirstSpirit Handbuch für Redakteure (JavaClient)".

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 512).



11.2 Architektur

11.2.1 Einleitung

Wie bereits aus der Einleitung ersichtlich sein dürfte, handelt es sich bei dem Thema der Berechtigungsprüfung um einen komplexen Aspekt, der viele Teile des FirstSpirit-Systems betrifft. In diesem Kapitel wird 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 (z. B.: ein CS4-Datenbank) verwendet.

Wesentlich für das Verständnis des Zusammenspiels der einzelnen Komponenten sind die Beziehungen und die Datenflüsse zwischen den Bestandteilen.



11.2.2 Übersicht

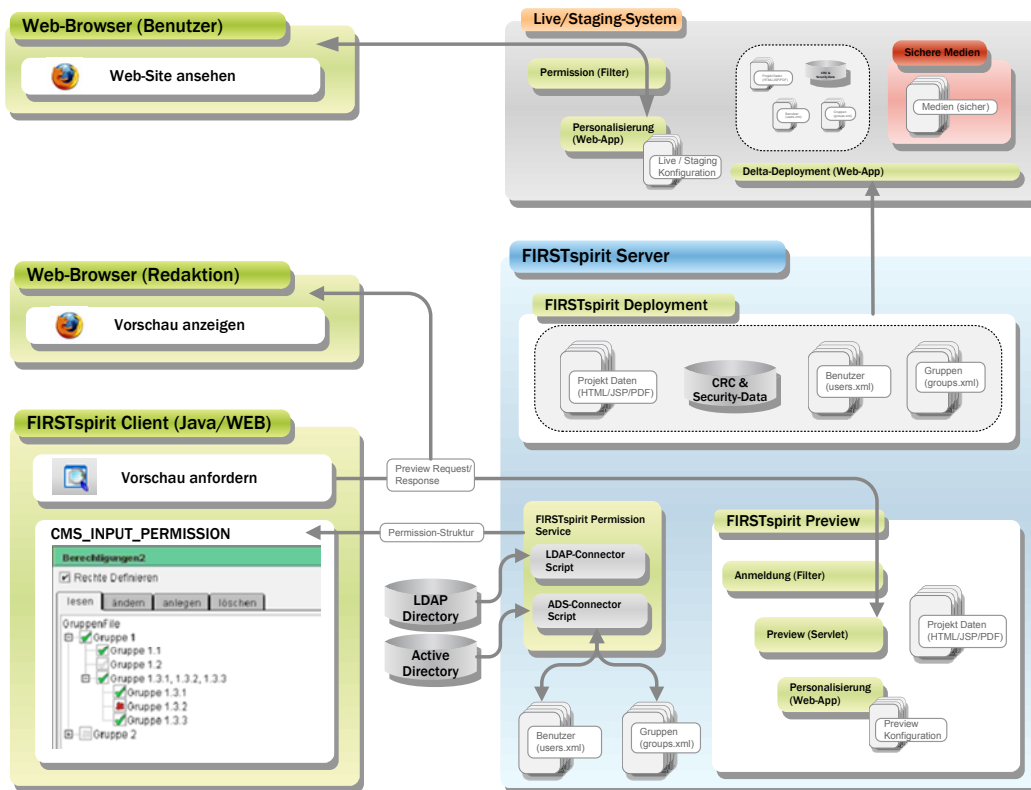


Abbildung 11-1: Übersicht Live-System

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 498).

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 498).

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 499).

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 501).



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 512).

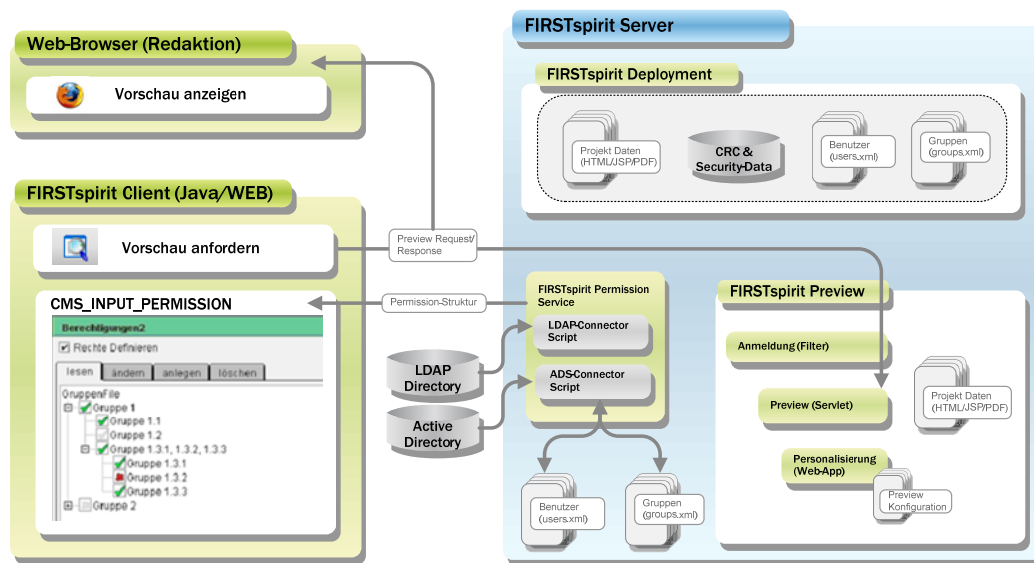


Abbildung 11-2: Ausschnitt Live-System

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 513). 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 442) 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, kann die Rechtedefinition entweder als "Soll-Berechtigung" für die Personalisierung (siehe Kapitel 11.3.1 Seite 502) dienen oder im Rahmen des Zugriffsschutzes über die Access-Control-Datenbank und den Multi-Access-Control-Filter (siehe Kapitel 11.3.2.2 Seite 505)



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 506). 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.

(Damit entfallen die für FirstSpirit V 3.1 notwendigen Mechanismen über den Security-Proxy.)

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 505). 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 498)



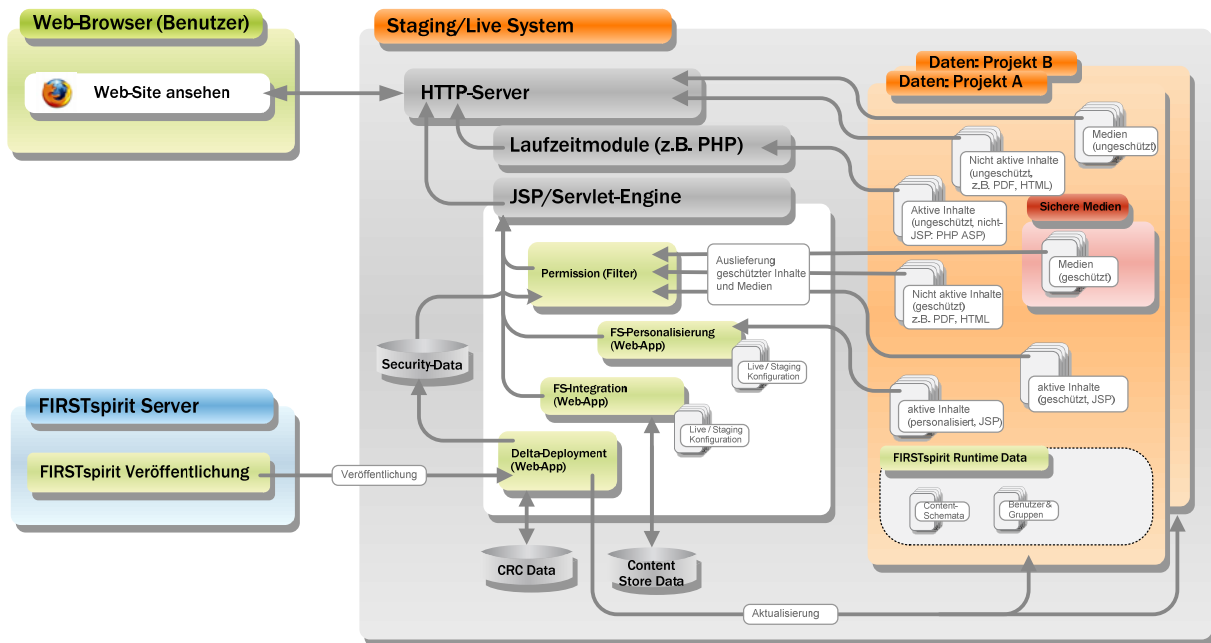


Abbildung 11-3: Live- und Staging-System

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-System und 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 (Abbildung 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 504).

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. Installation des Permission-Service auf dem FirstSpirit-Server (siehe Kapitel 7.3.14 Seite 251)
2. Permission-Service starten: Nach der Installation des Moduls erfolgt das Starten bzw. Stoppen der Dienste entweder über das FirstSpirit Server-Monitoring (siehe Kapitel 8.6.2.4 Seite 445) oder über die Server- und Projektkonfiguration (siehe Kapitel 7.3.14 Seite 251).
3. Systemkonfigurationsdateien (`service.ini`, `groups.xml` und `users.xml`) des Permission-Service anpassen (siehe Kapitel 11.4.3 Seite 512)
4. Metadatenvorlage erstellen (siehe FirstSpirit Handbuch für Entwickler)
`CMS_INPUT_PERMISSION` und `sichere Medien`
5. Web-Komponenten über die Server- und Projektkonfiguration konfigurieren (siehe Kapitel 7.4.18 Seite 319).



6. Berechtigungen über die Server- und Projektkonfiguration konfigurieren (siehe Kapitel 11.4.5 Seite 515)
 - Angabe der Eingabekomponente für sichere Medien
 - Angabe der Eingabekomponente für Rechteprüfung
7. Konfiguration des Multi-Access-Control-Filters (siehe Kapitel 11.3.2.2 Seite 505)

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 502)
- Prüfung der Zugriffsrechte auf ein Objekt über die Access-Control-Datenbank (siehe Kapitel 11.3.2 Seite 504)
- Zugriffsschutz für Medien innerhalb der FirstSpirit Vorschau ("Sichere Medien") (siehe Kapitel 11.3.3 Seite 506)

11.3.1 Personalisierung

Das Modul FirstSpirit Personalisation⁴¹ 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

⁴¹ Modul FirstSpirit Personalisation (siehe Dokumentation PERS40DE_FirstSpirit_Modules_Personalisation)



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 515).

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 504).



Beim Modul FirstSpirit Personalisation handelt es sich um eine kostenpflichtige Zusatzfunktionalität.

Weitere Informationen siehe Modul-Dokumentation FirstSpirit Personalisation.

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 aus FirstSpirit Version 3.1 bekannte Konzept der "Sicheren 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 werden⁴².

Die Konfiguration erfolgt über die FirstSpirit Server- und Projektkonfiguration (siehe Kapitel 7.4.16 Seite 315).

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 7.5.9.3.3 Seite 378), können über die Access-Control-Datenbank auch die Zugriffsrechte auf ein Objekt gespeichert werden (d.h. Rechte eines Besuchers

⁴² Modul FirstSpirit Security (siehe Dokumentation SECU40DE_FirstSpirit_Modules_Security)



der generierten Site – "Benutzerrechte"). Mithilfe von individuell konfigurierten 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.

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.

Weitere Informationen siehe Modul-Dokumentation FirstSpirit Security.



11.3.3 Konzept Sichere Medien

Ein bereits aus FirstSpirit Version 3.1 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 506)
- Innerhalb der Live-Systems über einen Filter (Multi-Access-Control-Filter) (siehe Kapitel 11.3.3.2 Seite 507)

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 315). 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 Medien (innerhalb des Ordners) 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 498). Dabei werden entweder 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 fs4preview 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



Durch die Einführung des Moduls FirstSpirit Security haben sich in FirstSpirit Version 4.0 (und 4.1) Änderungen für die Konfiguration (im Live-Stand) ergeben, da sich der Zugriffsschutz hier nicht länger nur auf Medien beschränkt, sondern alle Projektinhalte einschließt.

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 505) erforderlich⁴³. 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 504).

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 Personalisation

Im Einzelfall könnten 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 506) oder die "Prüfung der Zugriffsrechte über die Access-Control-Datenbank (siehe Kapitel 11.3.2 Seite 504) eingesetzt werden).

⁴³ siehe Dokumentation SECU40DE_FirstSpirit_Modules_Security.pdf



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 508).
2. Die Verwaltungen bilden ebenfalls eine Hierarchie, die häufig für die Vererbung von Rechtedefinitionen genutzt wird (siehe Kapitel 11.3.5.3 Seite 510).
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 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 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 gehen 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 ja der darüber liegende Einstiegspunkt fehlt.





Zusammenfassend kann man also sagen, 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 519).
3. Es werden im Gruppenbaum folgende drei Zustände definiert:
 -  Erlaubt = grünes Häkchen
 -  Verboten = rotes Kreuz
 -   wie Vater = 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 Gruppenshierarchie (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-Modul. Das Modul ist im Lieferumfang von FirstSpirit enthalten und muss zunächst auf dem FirstSpirit Server installiert werden (siehe Kapitel 7.3.14 Seite 251).

Nach der Installation des Moduls erfolgt das Starten bzw. Stoppen des Permission-Service entweder über das FirstSpirit Server-Monitoring (siehe Kapitel 8.6.2.4 Seite 445) oder über die Server- und Projektkonfiguration (siehe Kapitel 7.3.14 Seite 251).

11.4.3 Konfiguration der Server-Komponente

Die Konfiguration der Server-Komponente erfolgt durch eine Servicekonfigurationsdatei `service.ini`, in der die allgemeinen Einstellungen vorgenommen werden und optional einer Menge von Gruppensdefinitionen-Dateien im XML-Format (siehe Kapitel 11.4.3.1). Das Servicekonfigurationsverzeichnis für Module befindet sich unterhalb des FirstSpirit-Verzeichnisses im Unterverzeichnis "conf/modules". Die einzelnen Servicekonfigurationsdateien befinden sich in einem Unterverzeichnis unterhalb von "services", 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 Gruppenshierarchie 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 442) 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` = Liste der symbolischen Namen der verfügbaren Gruppenshierarchien.

Wenn eine Gruppenshierarchie aus einer XML-Datei erzeugt werden soll, so wird nur ein einziger Parameter benötigt:

`NAME.path` = Pfad zur Gruppen-XML-Datei

Soll die Gruppenshierarchie über ein Skript erzeugt werden, so sind eine Reihe von Parametern notwendig:

`NAME.path` = Pfad zur Gruppen-XML-Datei

`NAME.users` = Pfad zur User-XML-Datei

`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.

Standardkonfiguration der Datei service.ini:

```
## global params
# -----
# check each x seconds for changes
interval=20

# symbolic names for documents
documents=GruppenFile

#
# document specific params
# -----
GruppenFile.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
```



11.4.4 Konfiguration für die Veröffentlichung

Die Veröffentlichung der benötigten Dateien wird über ein Veröffentlichungs-Servlet realisiert (siehe Kapitel 7.5.9.3.3 Seite 378). Dazu wird in der Server- und Projektkonfiguration ein Veröffentlichungsauftrag für das Projekt angelegt (siehe Kapitel 7.5.4 Seite 351).

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.

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" die projektspezifischen Einstellungen für die Auswertung der Berechtigung vorgenommen werden (siehe Kapitel 7.4.16 Seite 315).



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-Server (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 7.5.9.3, Seite 372 ff.). Dann ist das "Zielverzeichnis" das Web-Applikationsverzeichnis, welches gemapped 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 "Gruppen.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 denen, 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 ist 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.

Im Gegensatz zur bisherigen Strategie wird in der "Gruppen.XML"-Datei zukünftig nur noch maximal eine ID pro Knoten erlaubt sein. Die ID kann aber weggelassen werden. Das bisher gelegentlich praktizierte Vorgehen, die Auflösung der Gruppenshierarchie mit "geschickter" ID-Vergabe implizit durchzuführen, wird nicht mehr unterstützt, sondern ist jetzt eine explizite Funktion der Rechtedefinitions-Komponente.



11.6.2 Konfigurationssemantik

Die Implikationen einer Berechtigungskonfiguration in Verbindung mit dem gewählten ID-Schema ist nicht in allen Aspekten trivial und wird FirstSpirit Handbuch für Redakteure anhand eines Beispiels ausführlicher erläutert werden:

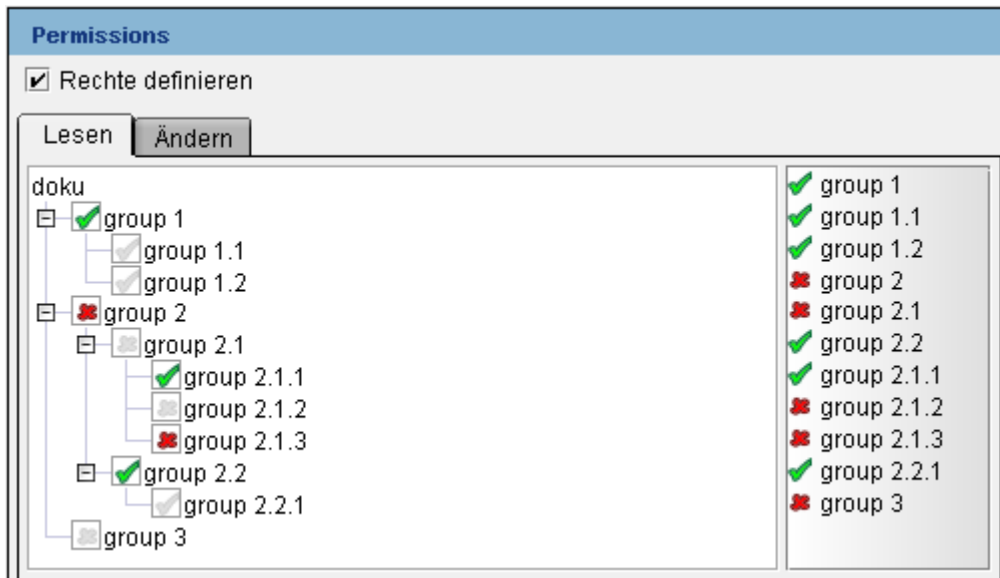


Abbildung 11-4: Beispiel Rechtedefinitions-Komponente

Die Rechtedefinitions-Komponente ist in der Lage, Rechte für unterschiedliche Operationen zu vergeben. Die Operationen sind grundsätzlich orthogonal, das heißt, es gibt kein "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: Konfigurationsdateien

Konfigurationsdateien, wie sie bei der Installation des FirstSpirit-Servers mit Standardwerten im Zielverzeichnis der Installation unter `conf/` angelegt werden.



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.

12.1 fs-server.conf

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

#####
# mail
#####
mail.smtp=
mail.default-recipient=
mail.sender=fs4server
```

12.2 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.org/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=java

# 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.
# Same syntax as Maxmem Java Heap Size.
# Set to value of about 75% of Maximum Java Heap Size.
wrapper.java.initmemory=525

# Java parameters.
# Continuous enumeration is needed!
# Empty values after the "=" sign are allowed.
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

# Select 64bit VM on Solaris with -d64:
wrapper.java.additional.6=#-d64

# Java parameters for garbage collection
# value of -Xmn should be set to 50% of wrapper.java.initmemory
wrapper.java.additional.7=-Xmn262M
wrapper.java.additional.8=-XX:PermSize=200m
wrapper.java.additional.9=-XX:MaxPermSize=200m
wrapper.java.additional.10=-XX:+DisableExplicitGC
wrapper.java.additional.11=-XX:SoftRefLRUPolicyMSPerMB=20
wrapper.java.additional.12=-XX:+UseParNewGC
wrapper.java.additional.13=-XX:+UseConcMarkSweepGC
wrapper.java.additional.14=-XX:+CMSIncrementalMode
wrapper.java.additional.15=-XX:+CMSParallelRemarkEnabled
wrapper.java.additional.16=-XX:+CMSClassUnloadingEnabled
wrapper.java.additional.17=-XX:SurvivorRatio=1
wrapper.java.additional.18=-XX:TargetSurvivorRatio=80
wrapper.java.additional.19=-XX:InitialTenuringThreshold=15
wrapper.java.additional.20=-XX:-UseLargePages
wrapper.java.additional.21=-Djava.rmi.dgc.leaseValue=3600000
wrapper.java.additional.22=-Dsun.rmi.dgc.server.gcInterval=3600000
wrapper.java.additional.23=-Dsun.rmi.dgc.client.gcInterval=3600000
wrapper.java.additional.24=
wrapper.java.additional.25=
wrapper.java.additional.26=
wrapper.java.additional.27=
wrapper.java.additional.28=
wrapper.java.additional.29=
wrapper.java.additional.30=
wrapper.java.additional.31=

# Monitoring: JMX-Connector for jconsole and other monitoring tools
# http://java.sun.com/j2se/1.5.0/docs/guide/management/agent.html
#wrapper.java.additional.32=-Dcom.sun.management.jmxremote
#wrapper.java.additional.33=-Dcom.sun.management.jmxremote.ssl=false
#wrapper.java.additional.34=-Dcom.sun.management.jmxremote.authenticate=false
#wrapper.java.additional.35=-Dcom.sun.management.jmxremote.port=9000
#wrapper.java.additional.36=
#wrapper.java.additional.37=
#wrapper.java.additional.38=
#wrapper.java.additional.39=

# Monitoring: Garbage Collector Log (Sun VM)
#wrapper.java.additional.40=-verbose:gc
#wrapper.java.additional.41=-XX:+PrintGCTimeStamps
#wrapper.java.additional.42=-XX:+PrintGCDetails
# If -Xloggc is not set, GC-Log will be written to fs-wrapper.log
# which is rotated regularly, see parameters wrapper.logfile.
#wrapper.java.additional.43=-XX:+PrintGCDateStamps
#wrapper.java.additional.44=-Xloggc:log/fs-gc.log

# Timeout parameters for controlling the Java process.
```



```
# Before changing read Wrapper documentation as all timeout parameters
# are related to each other.
wrapper.startup.timeout=30
wrapper.shutdown.timeout=180
wrapper.jvm_exit.timeout=30
wrapper.cpu.timeout=20
wrapper.ping.timeout=300
wrapper.timer_slow_threshold=15
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 sys/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=FIRSTspiritServer4

# Display name of the service
wrapper.ntservice.displayname=FirstSpirit 4.2

# 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

# Server update include file
#include ../server/update/fs-update.conf

#*****
# end of file
#*****
    
```

12.3 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.esprit.firstspirit.server.authentication.FSUserLoginModule sufficient
hash="true";
    de.esprit.firstspirit.server.authentication.FSTicketLoginModule
sufficient;
};

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

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

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

/* web authentication (for preview, webedit, webmonitor) with sso */
websso {
    de.esprit.firstspirit.server.authentication.FSTicketLoginModule
sufficient;
    // de.esprit.firstspirit.server.authentication.NTLMLoginModule optional
domains=":my-netbois-domain;:my-active-directory-domain";
    // de.esprit.firstspirit.server.authentication.LdapLoginModule optional
    
```



```

section="LDAP";
    de.espirit.firstspirit.server.authentication.FSUserLoginModule optional;
};

```

12.4 fs-webapp.xml

```

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE Configure PUBLIC "-//Mort Bay Consulting//DTD Configure//EN"
"http://jetty.mortbay.org/configure.dtd">

<!--
    Configuration of FIRSTspirit Web-Server.
    For parameter details see
    http://docs.codehaus.org/display/JETTY/Jetty+Documentation
    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.mortbay.jetty.Server">

    <!-- Server Thread Pool -->
    <!-- ===== -->
    <Set name="ThreadPool">
        <New class="org.mortbay.thread.concurrent.ThreadPool">
            <Set name="corePoolSize">5</Set>
            <Set name="maximumPoolSize">250</Set>
        </New>
    </Set>

    <!-- HTTP-Connector -->
    <!-- ===== -->
    <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>

    <!-- HTTPS-Connector -->
    <!-- ===== -->
    <!--
    <Call name="addConnector">
        <Arg>
            <New class="org.mortbay.jetty.security.SslSelectChannelConnector">
                <Set name="port">8443</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>
                <Set name="keystore"><SystemProperty name="cmsroot" />/conf/fs-
keystore.jks</Set>
                <Set name="password">changeit</Set>
                <Set name="keyPassword">changeit</Set>
            </New>
        </Arg>
    </Call>

```



```

-->

<!-- AJP-Connector for mod_proxy_ajp or mod_jk -->
<!-- ===== -->
<!--
<Call name="addConnector">
    <Arg>
        <New class="org.mortbay.jetty.ajp.Ajp13SocketConnector">
            <Set name="host">localhost</Set>
            <Set name="port">8009</Set>
        </New>
    </Arg>
</Call>
-->

<!-- Request Log -->
<!-- ===== -->
<!--
<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>
-->

<!-- FIRSTspirit Web Applications -->
<!-- ===== -->
<New
class="de.espirit.firstspirit.server.jetty.JettyManagerImpl$FailSafeWebAppContext
">
    <Arg><Ref id="Server" /></Arg>
    <Arg><SystemProperty name="WEBAPP_ROOT_PATH" /></Arg>
    <Arg><SystemProperty name="WEBAPP_ROOT_URL" /></Arg>
</New>
<New
class="de.espirit.firstspirit.server.jetty.JettyManagerImpl$FailSafeWebAppContext
">
    <Arg><Ref id="Server" /></Arg>
    <Arg><SystemProperty name="WEBAPP_WEBMON_PATH" /></Arg>
    <Arg><SystemProperty name="WEBAPP_WEBMON_URL" /></Arg>
</New>
<New
class="de.espirit.firstspirit.server.jetty.JettyManagerImpl$FailSafeWebAppContext
">
    <Arg><Ref id="Server" /></Arg>
    <Arg><SystemProperty name="WEBAPP_WEBEDIT_PATH" /></Arg>
    <Arg><SystemProperty name="WEBAPP_WEBEDIT_URL" /></Arg>
</New>
<New
class="de.espirit.firstspirit.server.jetty.JettyManagerImpl$FailSafeWebAppContext
">
    <Arg><Ref id="Server" /></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="Server" /></Arg>
    <Arg><SystemProperty name="WEBAPP_PREVIEW_PATH" /></Arg>
    <Arg><SystemProperty name="WEBAPP_PREVIEW_URL" /></Arg>
</New>

<New
class="de.espirit.firstspirit.server.jetty.JettyManagerImpl$FailSafeWebAppContext
">
    <Arg><Ref id="Server" /></Arg>
    <Arg><SystemProperty name="WEBAPP_WEBEDIT5_PATH" /></Arg>
    <Arg><SystemProperty name="WEBAPP_WEBEDIT5_URL" /></Arg>
</New>

</Configure>

```

12.5 ssh/rsync-Deployment-Skript

Weiterführende Erläuterungen zur Veröffentlichung über rsync / ssh siehe Kapitel 10 Seite 485.

```

#!/ Bashshell

// FIRSTspirit4 script for scheduled deployment via rsync and ssh.
// $Revision: 40272 $
//
// The script should be added via the project settings "Schedule management"
// as an "Extended FIRSTspirit script" after a "Generation Task".
// It copies all changed generated files via rsync/ssh from the generation task
// to the given path on the web server host.
//
// On first start, an error message is shown like
// "Warning: Permanently added 'myhost' (RSA) to the list of known hosts.".
// That message can be ignored and further runs does not show it.
//
// The script accepts the following parameters which can be changed
// via "Script Properties" in FIRSTspirit:

webuser="";
webhost="";
webpath="";
webinf="";
privkey="";
ssh="ssh";
rsync="rsync";
rsyncopts="-vrtz";
rsyncopts2=null;
rsyncopts3=null;
rsyncopts4=null;

import java.util.*;

tasklist = context.getTasks();
t = context.getTask();
s = t.getScheduleEntry();
p = s.getProject();
log = "rsync-ssh-Deployment for project \"" + p.getName() + "\": ";

context.logInfo(log + "script started");

params = t.getParameters();
if(params == null) {

```



```
context.logError(log + " failed. Missing parameters. "
    + "Required: host, user, path. "
    + "Optional: webinf, privkey, ssh, rsync.");
return;
}

for(pa: params) {
    switch(pa.getKey()) {
        case "webuser":
            webuser = pa.getValue();
            break;
        case "webhost":
            webhost = pa.getValue();
            break;
        case "webpath":
            webpath = pa.getValue();
            break;
        case "webinf":
            webinf = pa.getValue();
            break;
        case "privkey":
            privkey = pa.getValue();
            break;
        case "ssh":
            ssh = pa.getValue();
            break;
        case "rsync":
            rsync = pa.getValue();
            break;
        case "rsyncopts":
            rsyncopts = pa.getValue();
            break;
        case "rsyncopts2":
            rsyncopts2 = pa.getValue();
            break;
        case "rsyncopts3":
            rsyncopts3 = pa.getValue();
            break;
        case "rsyncopts4":
            rsyncopts4 = pa.getValue();
            break;
    }
}

if(webuser.equals("")) {
    context.logError(log + "failed. Parameter \"user\" missing.");
    return;
}
if(webhost.equals("")) {
    context.logError(log + "failed. Parameter \"host\" missing.");
    return;
}
if(webpath.equals("")) {
    context.logError(log + "failed. Parameter \"path\" missing.");
    return;
}
if(privkey.equals("")) {
    keyopt = "";
} else {
    keyopt = "-i " + privkey;
}

cmd = new Vector();

cmd1=new Vector();
cmd1.add(ssh);
if(keyopt.length() > 0) {
    cmd1.add("-i");
}
```



```

        cmd1.add(privkey);
    }
    cmd1.addAll(Arrays.asList(new String[]{"-oBatchMode=yes",
        "-oStrictHostKeyChecking=no",
        "-l", webuser, webhost, "echo", "success"}));
    cmd.add(cmd1);

    cmd2 = new Vector();
    cmd2.add(rsync);

    if(tasklist.size() <= 1) {
        // script testing mode
        // as at least one previous task for generating the files is needed
        cmd2.add("-n");
        srcpath = System.getProperty("cmsroot") + "/server/lib";
    } else {
        // normal scheduled mode
        i = context.getTaskIndex();
        if( i <= 0 ) {
            context.logError(log
                + "failed. No previous generate task found "
                + "in tasklist!");
            return;
        } else {
            srcpath = context.getPath();
        }
    }
}

if(srcpath == null || srcpath.length() == 0) {
    context.logError(log + "failed. empty srcPath variable!");
    return;
}

// Replace something like c:\path with /cygdrive/c/path as rsync from
// Cygwin does need this.
if(srcpath.indexOf(":") >= 0) {
    srcpath = "/cygdrive/" + srcpath.charAt(0) + srcpath.substring(2);
}

if(rsyncopts!=null && rsyncopts.length() > 0) {
    cmd2.add(rsyncopts);
}
if(rsyncopts2!=null && rsyncopts2.length() > 0) {
    cmd2.add(rsyncopts2);
}
if(rsyncopts3!=null && rsyncopts3.length() > 0) {
    cmd2.add(rsyncopts3);
}
if(rsyncopts4!=null && rsyncopts4.length() > 0) {
    cmd2.add(rsyncopts4);
}
cmd2.addAll(Arrays.asList(new String[]{"-e", ssh + " " + keyopt
    + " -oBatchMode=yes -oStrictHostKeyChecking=no -l " + webuser,
    srcpath, webhost + ":" + webpath}));
cmd.add(cmd2);

if(webinf != null && webinf.length()>0) {
    cmd3 = new Vector();
    cmd3.add(rsync);
    if(tasklist.size() <= 1) {
        // script testing mode
        // as at least one previous task for generating the files is needed
        cmd3.add("-n");
    }
    if(rsyncopts!=null && rsyncopts.length() > 0) {
        cmd3.add(rsyncopts);
    }
}
if(rsyncopts2!=null && rsyncopts2.length() > 0) {

```



```

        cmd3.add(rsyncopts2);
    }
    if(rsyncopts3!=null && rsyncopts3.length() > 0) {
        cmd3.add(rsyncopts3);
    }
    if(rsyncopts4!=null && rsyncopts4.length() > 0) {
        cmd3.add(rsyncopts4);
    }
    cmd3.addAll(Arrays.asList(new String[]{"-e", ssh + " " + keyopt
        + " -oBatchMode=yes -oStrictHostKeyChecking=no -l " + webuser,
        srcpath + "../WEB-INF", webhost + ":" + webpath + "/WEB-INF"}));
    cmd.add(cmd3);
}

class StreamCapture extends Thread {
    InputStream i;
    boolean useErrlog;

    StreamCapture(InputStream inputstream, boolean useErrorlog) {
        i = inputstream;
        useErrlog = useErrorlog;
    }

    public void run() {
        try {
            InputStreamReader reader = new InputStreamReader(i);
            BufferedReader buffr = new BufferedReader(reader);
            String line = null;
            while ((line = buffr.readLine()) != null) {
                if(useErrlog) {
                    context.logError(log + line);
                } else {
                    context.logInfo(log + line);
                }
            }
        } catch (IOException ex) {
            context.logError(log + "failed - " + ex);
        } finally {
            try {
                i.close();
            } catch (IOException exClose) {
                // ignore
            }
        }
    }
}

for(c: cmd) {
    cmdstr = "";
    for (String s : c) {
        cmdstr += " " + s;
    }
    cmdstr = "ProcessBuilder(" + cmdstr.substring(1) + ")";
    context.logInfo(log + cmdstr);

    Process p = null;
    try {
        p = new ProcessBuilder(c).start();
        StreamCapture stdout = new StreamCapture(p.getInputStream(),
false);
        StreamCapture stderr = new StreamCapture(p.getErrorStream(), true);
        stdout.start();
        stderr.start();
        result = p.waitFor();
        stdout.join(3000);
        stderr.join(3000);
        if ( result == 0 ) {

```



```
        context.logInfo(log + "command completed");
    } else {
        context.logError(log + "failed with exitcode " + result + "
for " + cmdstr);
    }
    } catch (InterruptedException e) {
        context.logError(log + "timeout for " + cmdstr + " - " + e);
    } catch (Exception e) {
        context.logError(log + "failed for " + cmdstr + " - " + e);
    } finally {
        if (p != null) {
            try {
                p.getOutputStream().close();
            } catch (IOException ignore) {
                // ignore
            }
        }
    }
}
}

context.logInfo(log + "script completed");
```

