FirstSpirit™

Your Content Integration Platform

Manual for Administrators

FirstSpirit Version 4.x

Version	**2.45**
State	**RELEASED**
Date	**2011-08-24**
Department	Techn. Documentation
Author/ Authors	B. Ehle, B.Gutknecht, H. Isenberg, D.Reinartz
Copyright	2011 e-Spirit AG
File name	ADMI4xEN_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

Table of contents

1 Introduction 12

1.1 Topic of this documentation 17

1.2 Structure of this documentation 18

1.3 The FirstSpirit user interface 19

2 Installation 21

3 FirstSpirit Server control 23

3.1 Unix 23

3.1.1 GNU/Linux and Solaris 9 23

3.1.2 Solaris 10 23

3.1.3 AIX 23

3.1.4 Via a normal user account 24

3.1.5 Generate a stack dump 24

3.2 Windows 25

3.2.1 Register / Deregister as system service via the start menu 25

3.2.2 Uninstall via the start menu 26

3.2.3 Start /Stop as a system service via the start menu 26

3.2.4 Start /Stop in console mode 26

3.2.5 Edit configuration files via the start menu 27

3.2.6 Further start menu functions 28

4 FirstSpirit Server configuration 30

4.1 File system organisation 30

4.1.1 File names 30

4.1.2 Web applications ... 30

4.1.3 Directory structures ... 31

4.2 General configuration information ... 31

4.3 Configuration files (FirstSpirit Server) ... 33

4.3.1 FirstSpirit Server configuration (fs-server.conf) ... 33

4.3.2 Configuration of the Java VM and the Java Wrapper (fs-wrapper.conf) ... 62

4.3.3 Database connection configuration (fs-database.conf) ... 71

4.3.4 Login process configuration (fs-jaas.conf) ... 72

4.3.5 Licence configuration (fs-license.conf) ... 87

4.3.6 Logging configuration (fs-logging.conf) ... 89

4.3.7 Web server configuration (fs-webapp.xml) ... 92

4.4 Connection to an LDAP server ... 96

4.4.1 Authenticate via LDAP ... 96

4.4.2 Bind LDAP attributes to a FirstSpirit user ... 97

4.4.3 Use TLS or SSL ... 98

4.5 Integration into an external web server ... 99

4.5.1 Apache HTTP server with Jetty servlet engine ... 99

4.5.2 Apache HTTP server with Tomcat servlet engine ... 103

4.5.3 External servlet engine (Tomcat) ... 107

4.5.4 Load balancing to several servlet engines ... 113

4.6 Integration into an external application server ... 115

4.6.1 Integration into WebSphere Application Server ... 116

4.6.2 External application servers requirements ... 119

4.7 HTTPS server configuration ... 120

4.7.1 Install a security certificate for a test server ... 120

4.7.2 Install a trusted security certificate ... 121

4.8 Database connection ... 124

4.8.1 Storing the JDBC driver files 124
4.8.2 Creating a JDBC driver module 125
4.8.3 Installation and configuration of the JDBC driver module 131
4.8.4 Data source configuration 137
4.8.5 Required permissions for database user accounts 143
4.8.6 Notifications and restrictions concerning the specific database systems 145
4.8.7 Examples for linking different database systems 148
4.8.8 Procedure for connecting external databases 152
4.9 Roll-out process for native applications (from V4.2) 153
4.9.1 Roll-out process (server) 153
4.9.2 Roll-out process (workstation computer) 153
4.9.3 Updating the native system components 155
4.9.4 Preventing the overwriting of files during the roll-out process 156

5 FirstSpirit web application configuration 157

5.1 FirstSpirit start page configuration (fs4root) 157
5.2 WebClient configuration 157
5.2.1 Project prerequisites when using WebEdit 158
5.2.2 WebEdit as a local project application (from V4.1) 159
5.2.3 Browser configuration when using WebEdit 160
5.2.4 Activate scripts in WebEdit 160
5.2.5 Prevent "directory browsing" 161
5.2.6 Configure WebEdit tree presentation 162
5.2.7 Configure workflows in WebEdit 163
5.3 Single Sign-On (SSO) configuration 164
5.3.1 SSO in connection with the JavaClient 164
5.3.2 WebEdit with portal authentication 165

5.3.3 Login with plain text password 165

5.3.4 Restrictions for using WebEdit 166

5.4 Pop-up blocker deactivation 166

6 FirstSpirit start page 168

6.1 Automatic login via Single Sign-On 168

6.2 Login with user name and password 168

6.3 FirstSpirit start page 170

6.3.1 Revised start page in FirstSpirit Version 4.1 171

6.3.2 Quick start 171

6.3.3 Client start 171

6.3.4 Administration 172

6.3.5 User 173

6.4 Application start 178

6.4.1 JavaClient 178

6.4.2 WebClient 179

6.4.3 Server Monitoring 179

6.4.4 Server and Project Configuration 179

6.5 Start FirstSpirit client as JAVA application 180

6.5.1 Socket mode 180

6.5.2 HTTP (Internet) mode 182

7 FirstSpirit Server and Project Configuration 184

7.1 Server and project administrators 184

7.2 Menu bar elements 185

7.2.1 File 185

7.2.2 Server 185

7.2.3 Project 189

7.2.4 User ... 204
7.2.5 Extras ... 208
7.2.6 Help ... 209
7.3 Server properties ... 210
7.3.1 Global server properties ... 210
7.3.2 Presentation channels ... 214
7.3.3 Preview configurations ... 215
7.3.4 Conversion rules ... 216
7.3.5 Installed fonts ... 218
7.3.6 Databases ... 220
7.3.7 Language templates ... 224
7.3.8 Webstart ... 226
7.3.9 Start page ... 227
7.3.10 Schedule entry overview ... 229
7.3.11 Schedule entry management ... 230
7.3.12 Action templates ... 230
7.3.13 JAAS configuration ... 231
7.3.14 Modules ... 232
7.3.15 Web server ... 238
7.3.16 Web applications ... 243
7.3.17 Clustering (from V4.1) ... 248
7.4 Project properties ... 255
7.4.1 Project ... 256
7.4.2 Options ... 257
7.4.3 Substitutions ... 264
7.4.4 Fonts ... 265
7.4.5 Languages ... 265
7.4.6 Resolutions ... 268

7.4.7 Users 271
7.4.8 Groups 273
7.4.9 Schedule entry overview 279
7.4.10 Schedule entry management 280
7.4.11 Action templates 280
7.4.12 Databases 281
7.4.13 Template sets 282
7.4.14 WebEdit settings 284
7.4.15 Quota 287
7.4.16 Permissions 287
7.4.17 Project components 289
7.4.18 Web components 291
7.4.19 Remote projects 296
7.4.20 Media constraints (from V4.1) 299
7.4.21 Client applications 302
7.5 Schedule entry planning 310
7.5.1 Schedule entry overview 311
7.5.2 Schedule entry management 314
7.5.3 Action templates 317
7.5.4 Add / Edit a schedule entry (properties tab) 318
7.5.5 Add / Edit a schedule entry (actions tab) 322
7.5.6 Add actions to a schedule entry 324
7.5.7 Copy actions from another schedule entry 325
7.5.8 Add actions via action templates 326
7.5.9 Project-related actions 328
7.5.10 Server-related actions 350
7.6 Clustering – load distribution on generation (from V4.1) 351
7.6.1 Introduction (from V4.1) 351

7.6.2 Concept (from V4.1) ... 352
7.6.3 Check licence file (from V4.1) ... 353
7.6.4 Configuration of the cluster nodes (from V4.1) ... 354
7.6.5 Configuration of the generation schedule (from V4.1) ... 354
7.7 Configuration of the spelling check ... 356
7.7.1 Install/uninstall SpellService (server properties) ... 357
7.7.2 Update SpellService (server properties) ... 359
7.7.3 Configure Global SpellService ... 359
7.7.4 Configure global dictionaries ... 361
7.7.5 Start and configure "SpellService" service ... 363
7.7.6 Add SpellService as project component ... 364
7.7.7 Project-specific Spell Service configuration ... 365
7.7.8 Add project-specific dictionaries ... 367
7.8 Support for Apache FOP (from V4.1) ... 368
7.9 Project archiving ... 370
7.9.1 Version history ... 370
7.9.2 Revisions ... 371
7.9.3 Minimum project archiving requirement ... 372
7.9.4 Version history following archiving ... 373
8 FirstSpirit Server Monitoring ... 376
8.1 Overview ... 377
8.1.1 Overview – State ... 377
8.1.2 Overview – Activities ... 378
8.1.3 Overview – Sessions ... 379
8.2 Projects ... 380
8.2.1 Projects – Overview ... 380
8.2.2 Projects – Statistic ... 381

8.3 Logfiles ... 383

8.3.1 Logfiles – Complete server ... 383

8.3.2 Logfiles – according to project ... 388

8.3.3 Logfiles – according to deployment ... 390

8.4 Scheduling ... 392

8.4.1 Scheduling – Overview ... 392

8.4.2 Scheduling – Planned schedule entries ... 393

8.4.3 Scheduling – Executed schedule entries ... 393

8.5 User ... 394

8.5.1 User – Search ... 394

8.6 FirstSpirit ... 395

8.6.1 FirstSpirit – Configuration ... 395

8.6.2 FirstSpirit – Control ... 401

8.6.3 FirstSpirit – Message ... 404

8.6.4 FirstSpirit – Databases ... 405

8.6.5 FirstSpirit – Monitoring ... 406

8.6.6 FirstSpirit – Clustering ... 415

9 FirstSpirit JMX Console ... 417

9.1 JMX Console start ... 417

9.2 Tabs ... 420

9.3 AuthenticationManager ... 420

9.4 BackendManager ... 421

9.4.1 Attributes ... 421

9.4.2 Operation ... 422

9.5 CacheSizeManager ... 423

9.5.1 Attributes (all) ... 423

9.5.2 Operation ... 424

9.5.3 Attributes (project-related) ... 425
9.6 ContentManager ... 426
9.7 EventManager ... 426
9.8 ExecutionManager ... 427
9.8.1 Thread queue classification ... 427
9.8.2 Processing in the ExecutionManager ... 428
9.8.3 Attributes ... 429
9.9 LockManager ... 431
9.10 MediaManager ... 432
9.11 PreviewManager ... 433
9.12 ReferenceManager ... 434
9.13 RepositoryManager ... 436
9.14 SearchManager ... 437
9.15 SessionManager ... 438

10 Secure deployment via rsync and ssh ... 439
10.1 Web server under Unix ... 439
10.2 Web server under Windows ... 440
10.3 FirstSpirit Server under Unix ... 441
10.4 FirstSpirit Server under Windows ... 442
10.5 FirstSpirit project configuration ... 443

11 User permission configuration ... 445
11.1 Introduction ... 445
11.1.1 Define user permissions ... 446
11.1.2 Check user permissions ... 447
11.1.3 Permission module ... 448
11.2 Architecture ... 449

11.2.1 Introduction 449
11.2.2 Overview 450
11.2.3 Permission check via FirstSpirit 454
11.3 Protection of personalised project content in FirstSpirit 455
11.3.1 Personalisation 455
11.3.2 Checking access rights via the Access Control Database 456
11.3.3 Secure Media concept 458
11.3.4 Scope 459
11.3.5 Permission definition 460
11.3.6 Definitions 462
11.4 Configuration 463
11.4.1 Introduction 463
11.4.2 Activate the permission service 463
11.4.3 Configure the server component 464
11.4.4 Configuration for the deployment 466
11.4.5 Permission configuration via the project properties 467
11.5 Application in the LIVE system 467
11.5.1 Servlet server configuration 467
11.6 Application in the project 468
11.6.1 Manual group definition and ID assignment 468
11.6.2 Configuration sematics 469
12 Appendix: Configuration files 471
12.1 fs-server.conf 471
12.2 fs-wrapper.conf 472
12.3 fs-jaas.conf 475
12.4 fs-webapp.xml 476
12.5 ssh/rsync deployment script 478

1 Introduction

FirstSpirit™ is a classic client/server application which is entirely based on Java and web technology. Access to the FirstSpirit™ Server occurs either via the FirstSpirit™ JavaClient or the FirstSpirit WebClient. Communication between the FirstSpirit™-Client and the FirstSpirit Server is, depending on the application scenario, either based on HTTP(S) using an application server, or on the direct application of TCP.

The live system actually provides contents to the end user. The FirstSpirit™ Server does not necessarily have to be connected to the live system all the time. If there is no permanent connection to the live system, all the data required by the live system is transferred during deployment, so that no direct interaction between the live system and the FirstSpirit™ Server is necessary.

The following diagram shows the total system architecture:

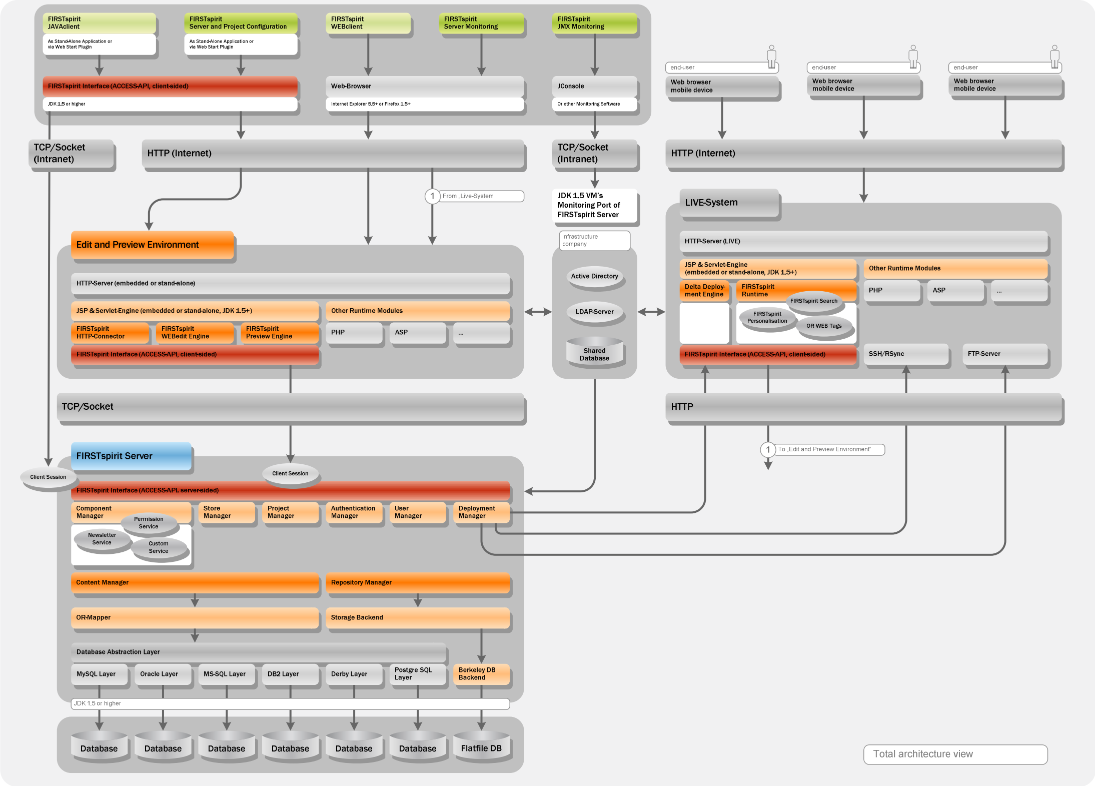

FIRSTspirit JAVAclient
As Stand-Alone Application or via Web Start Plugin
FIRSTspirit Server and Project Configuration
As Stand-Alone Application or via Web Start Plugin
FIRSTspirit WEBclient
FIRSTspirit Server Monitoring
FIRSTspirit JMX Monitoring
FIRSTspirit Interface (ACCESS-API, client-sided)
JDK 1.5 or higher
Web-Browser
Internet Explorer 5.5+ or Firefox 1.5+
JConsole
Or other Monitoring Software
end-user
Web browser mobile device
end-user
Web browser mobile device
end-user
Web browser mobile device
TCP/Socket (Intranet)
HTTP (Internet)
TCP/Socket (Intranet)
HTTP (Internet)
1
From „Live-System
JDK 1.5 VM's Monitoring Port of FIRSTspirit Server
Infrastructure company
LIVE-System
Edit and Preview Environment
HTTP-Server (embedded or stand-alone)
JSP & Servlet-Engine (embedded or stand-alone, JDK 1.5+)
Other Runtime Modules
FIRSTspirit HTTP-Connector
FIRSTspirit WEBedit Engine
FIRSTspirit Preview Engine
PHP
ASP
...
FIRSTspirit Interface (ACCESS-API, client-sided)
Active Directory
LDAP-Server
Shared Database
HTTP-Server (LIVE)
JSP & Servlet-Engine (embedded or stand-alone, JDK 1.5+)
Other Runtime Modules
Delta Deployment Engine
FIRSTspirit Runtime
FIRSTspirit Search
FIRSTspirit Personalisation
OR WEB Tags
PHP
ASP
...
FIRSTspirit Interface (ACCESS-API, client-sided)
SSH/RSync
FTP-Server
TCP/Socket
HTTP
1
To „Edit and Preview Environment"
FIRSTspirit Server
Client Session
Client Session
FIRSTspirit Interface (ACCESS-API, server-sided)
Component Manager
Permission Service
Newsletter Service
Custom Service
Store Manager
Project Manager
Authentication Manager
User Manager
Deployment Manager
Content Manager
Repository Manager
OR-Mapper
Storage Backend
Database Abstraction Layer
MySQL Layer
Oracle Layer
MS-SQL Layer
DB2 Layer
Derby Layer
Postgre SQL Layer
Berkeley DB Backend
JDK 1.5 or higher
Database
Database
Database
Database
Database
Database
Flatfile DB
Total architecture view

Figure 1-1: Total view

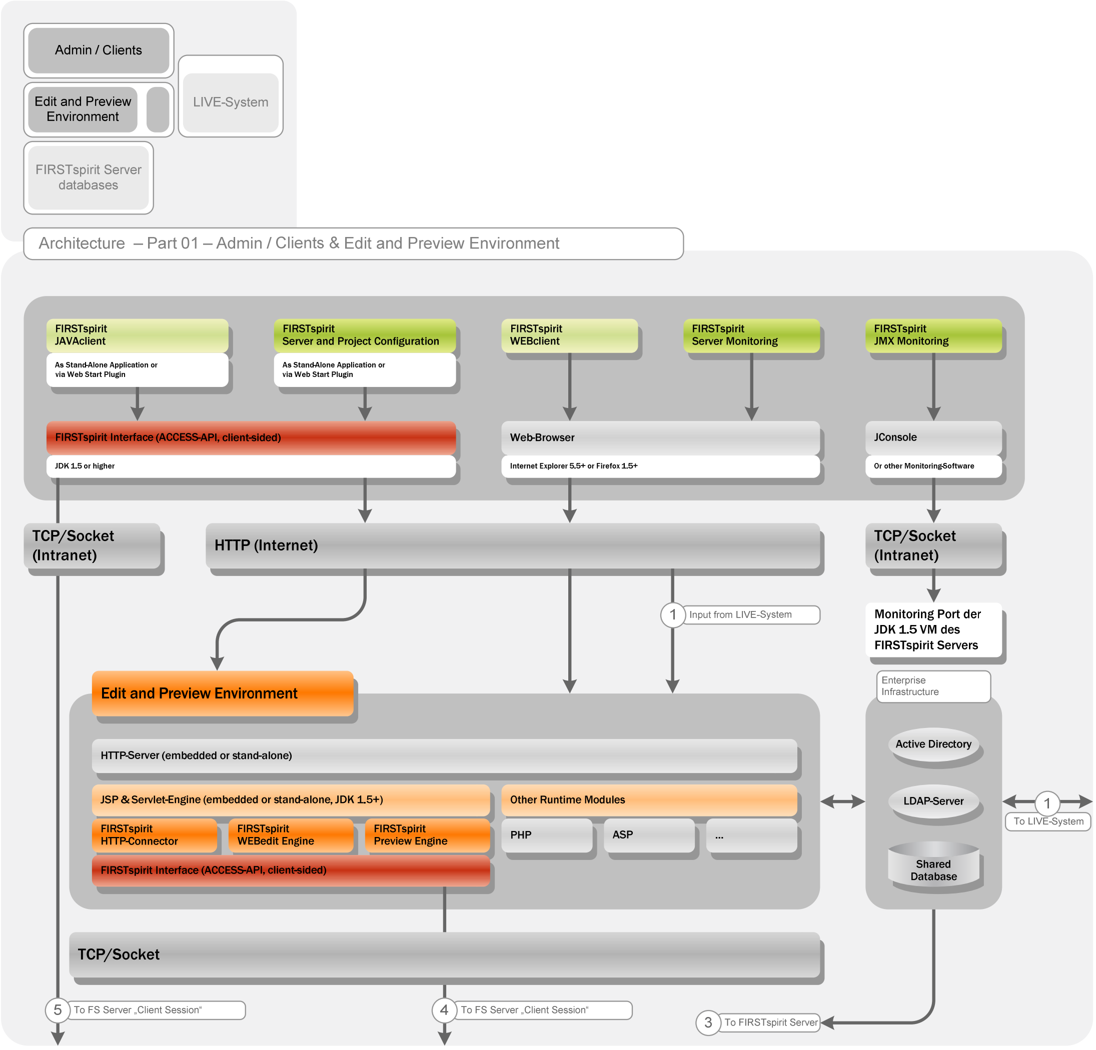

Admin / Clients
Edit and Preview Environment
LIVE-System
FIRSTspirit Server databases
Architecture – Part 01 – Admin / Clients & Edit and Preview Environment
FIRSTspirit JAVAclient
As Stand-Alone Application or via Web Start Plugin
FIRSTspirit Server and Project Configuration
As Stand-Alone Application or via Web Start Plugin
FIRSTspirit WEBclient
FIRSTspirit Server Monitoring
FIRSTspirit JMX Monitoring
FIRSTspirit Interface (ACCESS-API, client-sided)
JDK 1.5 or higher
Web-Browser
Internet Explorer 5.5+ or Firefox 1.5+
JConsole
Or other Monitoring-Software
TCP/Socket (Intranet)
HTTP (Internet)
TCP/Socket (Intranet)
1
Input from LIVE-System
Monitoring Port der JDK 1.5 VM des FIRSTspirit Servers
Enterprise Infrastructure
Edit and Preview Environment
HTTP-Server (embedded or stand-alone)
JSP & Servlet-Engine (embedded or stand-alone, JDK 1.5+)
Other Runtime Modules
FIRSTspirit HTTP-Connector
FIRSTspirit WEBedit Engine
FIRSTspirit Preview Engine
PHP
ASP
...
FIRSTspirit Interface (ACCESS-API, client-sided)
Active Directory
LDAP-Server
Shared Database
1
To LIVE-System
TCP/Socket
5
To FS Server „Client Session"
4
To FS Server „Client Session"
3
To FIRSTspirit Server

Figure 1-2: Total architecture – Detail: Edit and Preview Environment

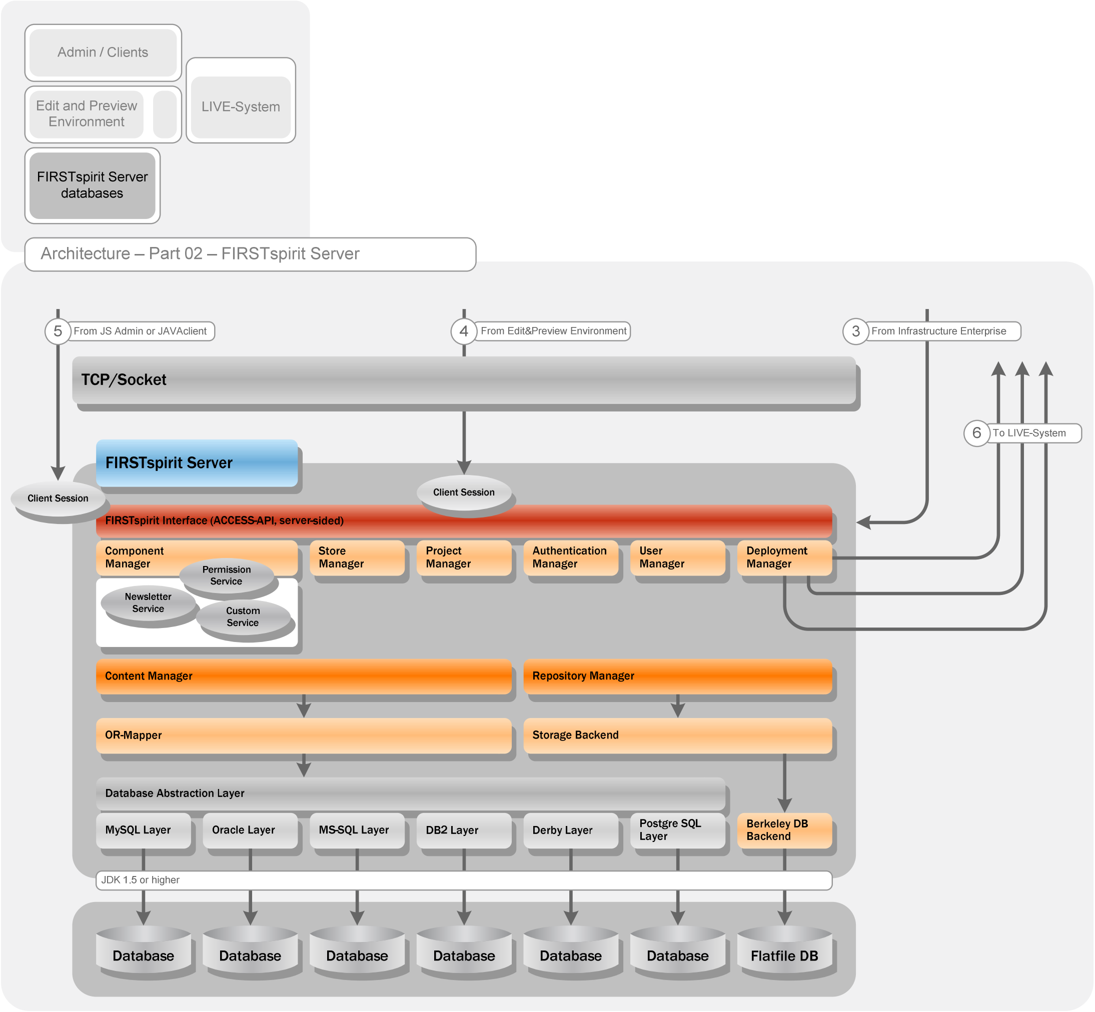


Figure 1-3: Total architecture – Detail: FirstSpirit Server

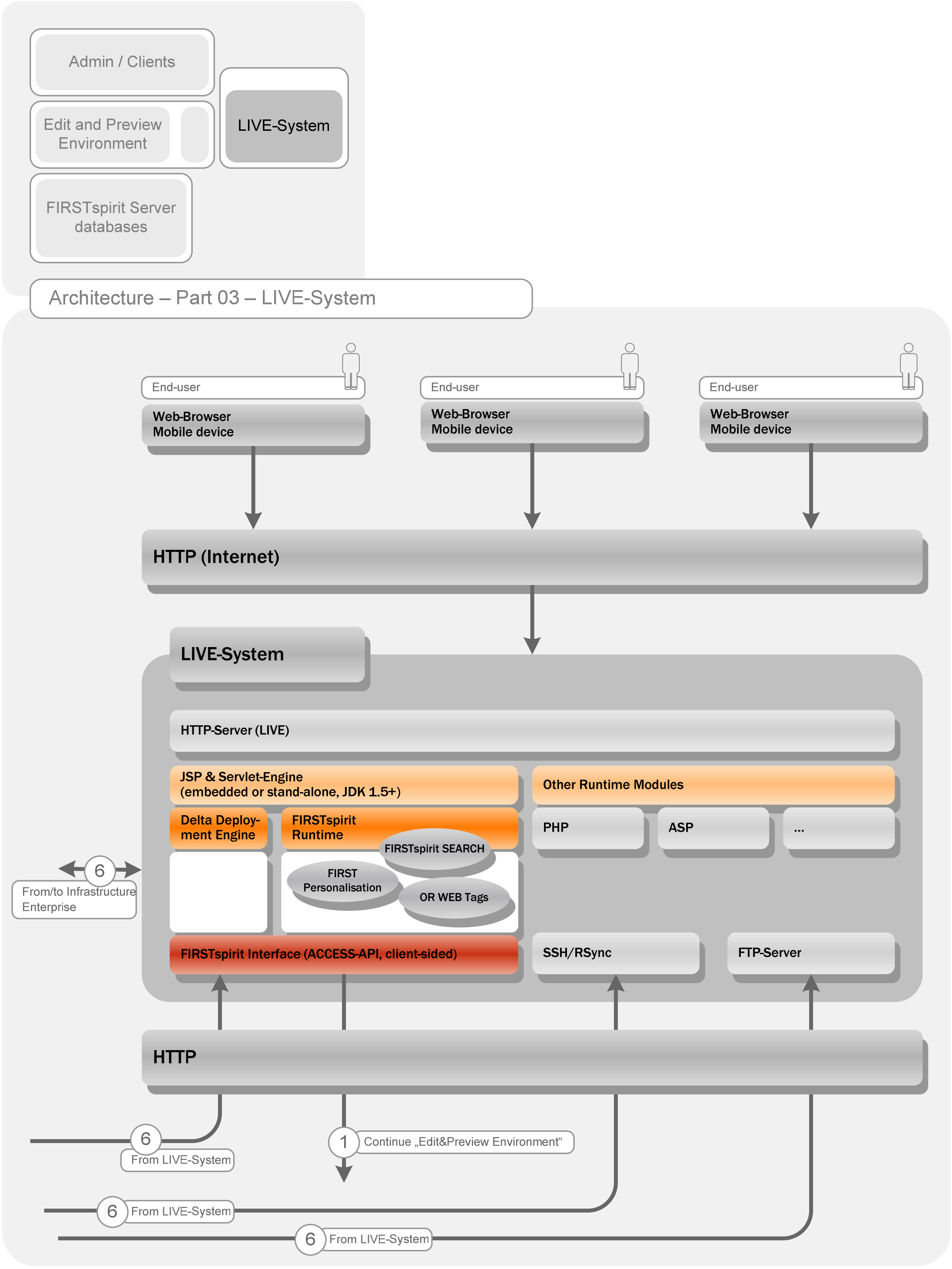

Admin / Clients
Edit and Preview Environment
LIVE-System
FIRSTspirit Server databases
Architecture – Part 03 – LIVE-System
End-user
Web-Browser Mobile device
End-user
Web-Browser Mobile device
End-user
Web-Browser Mobile device
HTTP (Internet)
LIVE-System
HTTP-Server (LIVE)
JSP & Servlet-Engine (embedded or stand-alone, JDK 1.5+)
Other Runtime Modules
Delta Deploy-ment Engine
FIRSTspirit Runtime
FIRSTspirit SEARCH
FIRST Personalisation
OR WEB Tags
PHP
ASP
...
6
From/to Infrastructure Enterprise
FIRSTspirit Interface (ACCESS-API, client-sided)
SSH/RSync
FTP-Server
HTTP
6
From LIVE-System
1
Continue „Edit&Preview Environment"
6
From LIVE-System
6
From LIVE-System

Figure 1-4: Total architecture – Detail: LIVE system

1.1 Topic of this documentation

FirstSpirit provides all users, depending on their tasks, with a client which has been precisely adapted to respective requirements. Therefore, FirstSpirit offers multiple clients for varying tasks and users. Generally speaking, a distinction is made between editorial environments and administration environments. While the FirstSpirit editorial environments support the work of editors and template developers, the administration environments have been primarily designed to monitor and configure FirstSpirit.

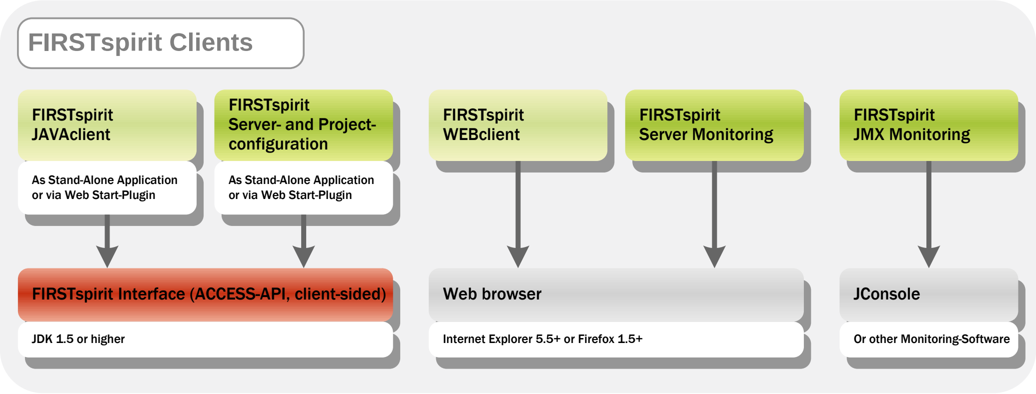

FIRSTspirit Clients
FIRSTspirit JAVAclient
As Stand-Alone Application or via Web Start-Plugin
FIRSTspirit Server- and Project-configuration
As Stand-Alone Application or via Web Start-Plugin
FIRSTspirit WEBclient
FIRSTspirit Server Monitoring
FIRSTspirit JMX Monitoring
FIRSTspirit Interface (ACCESS-API, client-sided)
JDK 1.5 or higher
Web browser
Internet Explorer 5.5+ or Firefox 1.5+
JConsole
Or other Monitoring-Software

Figure 1-5: FirstSpirit editorial environments

The documentation for administrators describes all aspects of FirstSpirit V4.x administration as well as the required administration environments and is, therefore, exclusively aimed at administrators. The FirstSpirit editorial environments (JavaClient and WebClient) are described in detail in the separate documentation for editors.

- **FirstSpirit Server and Project Configuration:** The FirstSpirit Server and Project Configuration is a Java application with a convenient, swing-based user interface which supports the FirstSpirit administrator for general, administrative FirstSpirit tasks. For example, the user interface can be used to create and configure new FirstSpirit projects. More extensive functions are possible in addition to the general tasks. The Server and Project Configuration can, e.g., be used to define users or integrate existing identity management systems, such as LDAP or Active Directory. Analogue to the JavaClient, the Server and Project Configuration is started and updated via Java Web Start (see chapter 7).

- **FirstSpirit Server Monitoring:** The browser-based FirstSpirit Server Monitoring is a web application for monitoring the FirstSpirit Server. Analogue to the WebClient, FirstSpirit Server Monitoring is operated via a web browser (see chapter 7.6).

- **FirstSpirit JMX Console:** Using Java Management Extensions (JMX) it is possible to monitor Java applications in a standardised manner. While the primary task of FirstSpirit Server Monitoring is to manually monitor a FirstSpirit Server, the JMX Console is used for automatic monitoring and can be perfectly integrated into an existing, company-wide monitoring system, if required. In principle, the JMX Console can also be used interactively. Compared to Server Monitoring, clearly finer granular information is provided (see chapter 9).

Note: FirstSpirit is not a universal "out-of-the-box" product, but continuously developed software. New functionalities and customer suggestions are constantly integrated and realised. These constant updates can only be reflected in the documentation to a limited degree. This may result in some figures in this documentation varying from the current FirstSpirit view. Please do not be confused by this, and just follow the instructions as normal.

! *This manual applies to FirstSpirit Version 4.2 Release 2 (4.2R2), unless otherwise noted.*

1.2 Structure of this documentation

The structure of this documentation is based on the functions provided by the administration environments.

Chapter 1: The introductory explanations in chapter 1 are followed by a brief overview of the FirstSpirit installation in this chapter. Detailed, separate documentation regarding this topic can be found in the “FirstSpirit Installation Instructions” (page 21 ff).

Chapter 3: This chapter describes the control of the FirstSpirit Server under Unix and Windows operating systems (page 23 ff).

Chapter 4: The FirstSpirit Server is configured via configuration files located in the installation directory of the FirstSpirit Server. These configuration files and their parameters are described in this chapter (page 30 ff).

Chapter 5: Configuration of the FirstSpirit web applications – especially project requirements, configuration and limitations of the WebClient – is described in this chapter (page 157 ff).

Chapter 6: The FirstSpirit start page is a web application for starting the FirstSpirit editorial and administration environments (page 168 ff).

Chapter 7: This chapter describes the project-wide functions which can be called via the Server and Project Configuration menu bar. The FirstSpirit Server and Project Configuration supports the general, administrative FirstSpirit tasks of the FirstSpirit administrator (page 184 ff).

Chapter 8: The browser-based FirstSpirit Server Monitoring is used to monitor the FirstSpirit Server and displays current operating parameters (e.g. number of users, memory load) (page 351 ff).

Chapter 9: Using Java Management Extensions (JMX) it is possible to automatically monitor Java applications. The JMX Console can be perfectly integrated into existing, company-wide monitoring, if required (page 417 ff).

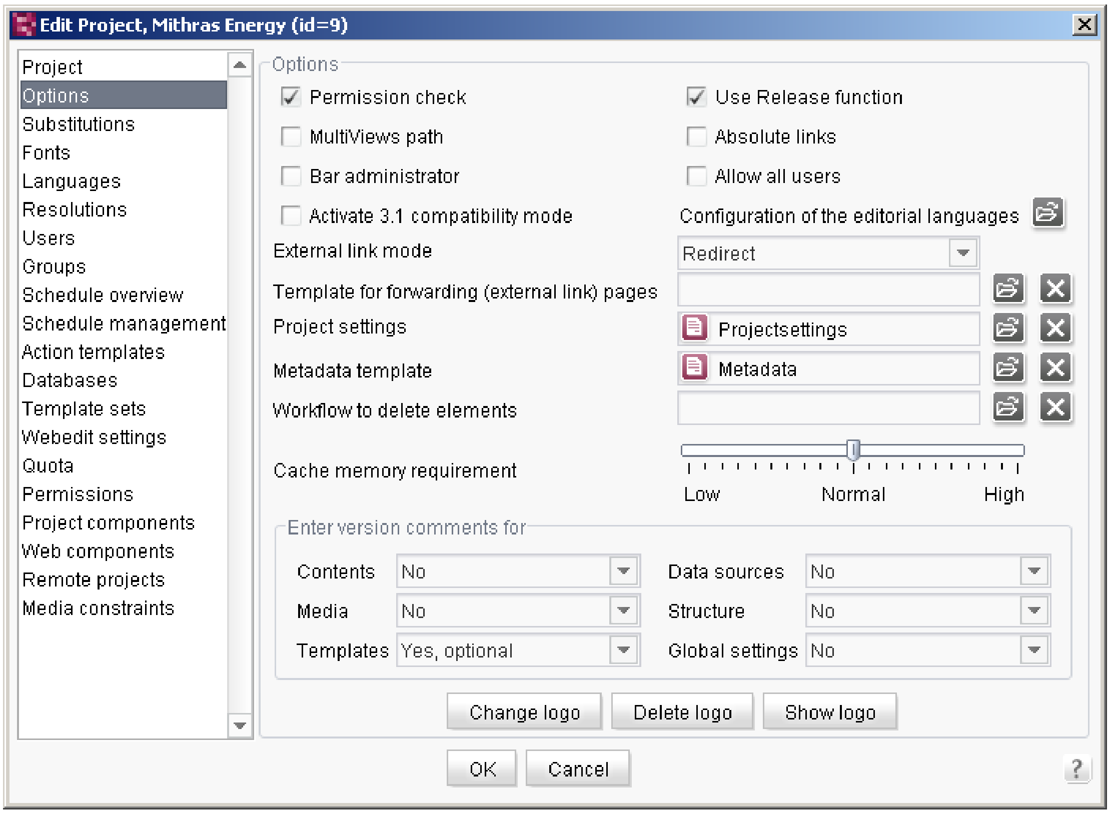
Chapter 10: Combined utilisation of the external service programs `rsync` and `ssh` is recommended for deployment via unsecured Internet connections or networks with low bandwidth (page 439 ff).

Chapter 11: This chapter outlines the mechanisms for user permission assignment and permission check provided by FirstSpirit and their precise application (page 445 ff).

1.3 The FirstSpirit user interface

FirstSpirit Version 4.1 is accompanied by the official release of the new "Look & Feel". The FirstSpirit server and project configuration application and JavaClient are then available with two display options: as Look & Feel "Classic" and the new Look & Feel "LightGray" (Default setting).

Customers who would rather work with the classic "look & feel" version in Version 4.1 too, can switch over to this on a project-specific or server-wide basis via the FirstSpirit server and project configuration. (It is only possible to switch over the "look & feel" if the applications are started from the FirstSpirit start page.)



Options

☑ Permission check ☑ Use Release function
□ MultiViews path □ Absolute links
□ Bar administrator □ Allow all users
□ Activate 3.1 compatibility mode Configuration of the editorial languages

External link mode	Redirect
Template for forwarding (external link) pages	
Project settings	Projectsettings
Metadata template	Metadata
Workflow to delete elements	
Cache memory requirement	Low Normal High

Enter version comments for

Contents	No	Data sources	No
Media	No	Structure	No
Templates	Yes, optional	Global settings	No

Figure 1-6: Look & Feel "LightGray" for server and project configuration

Project-specific changeover is possible via the additional `lookAndFeel=classic` or `LookAndFeel=lightgrey` parameters during configuration of the quick-start entries by means of the server and project configuration (see Chapter 7.3.9 page 227).

Server-wide changeover is possible via the additional `lookAndFeel=classic` and `lookAndFeel=lightgrey` parameters of the Webstart settings for JavaClient or the server and project configuration ("Webstart" menu entry) (see Chapter 7.3.8 page 226).

In addition, the `-DlookAndFeel=classic` parameter can be used when the application is started.

> ! *New functions released from FirstSpirit Version 4.1 are displayed in this document with the new look & feel; for the time being, existing functions continue to be displayed with the old look & feel.*

The classic "look & feel" is no longer available **from FirstSpirit Version 4.2**.

2 Installation

Due to the application of Java, FirstSpirit is a widely platform-independent client/server system. Generally speaking, installation only relates to the FirstSpirit Server, since the client is either managed via Java Web Start (JavaClient) or operated as a web application via a web browser (WebClient). An installed FirstSpirit Server has a uniform design on all operating systems due to its platform independency. Discrepancies only occur during the installation process. In the installation process all FirstSpirit Server files are installed in the target directory, exept a few files required for the system start.

The FirstSpirit Server Version 4.0 is available for the following operating systems:

- Red Hat Enterprise Linux
- Debian/GNU Linux
- SUSE Linux Enterprise
- SUSE Linux
- LSB-compliant Linux distributions
- Solaris
- AIX
- Windows

All operating systems are supported in 32 and 64-bit architecture.

Precisely 1 FirstSpirit server only can ever be installed under an operating system instance using the package systems: RPM, Debian Package or Solaris Package. In the event of renewed installation, the existing FirstSpirit server is updated, although all usage and configuration data is retained, – i.e. if a FirstSpirit installation, Version 4.1 already exists, when FirstSpirit Version 4.2 is installed the existing server is updated from Version 4.1 to Version 4.2. If several FirstSpirit servers are to be installed under a single operating system instance, they must be installed under a normal user account using the tar archive.

By contrast, if it is installed under Windows operating systems, an explicit differentiation is made between an update and a new installation:

- If FirstSpirit Version 4.2 is installed as a new installation or re-installation, a new FirstSpirit server is installed (with separate service, Start menu, etc.).
- If an update is installed, the existing installation is updated.

For further information about installation and updating FirstSpirit servers, please refer to "FirstSpirit Installation Instructions Version 4.2".

! *From FirstSpirit Version 4.1, it is possible to install one (or several) FirstSpirit Version 4.0 servers and one (or several) FirstSpirit Version 4.1 servers in parallel* ***under Windows operating systems*** *(see "FirstSpirit Installation Instructions" for Version 4.1).*

3 FirstSpirit Server control

To Control via the Server Monitoring see section 8.6.2.

> *Prior to starting and stopping the FirstSpirit Server, activate the maintenance mode (see section 8.6.2.1 page 401).*

3.1 Unix

3.1.1 GNU/Linux and Solaris 9

Start as `root`:

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

Stopp as `root`:

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

3.1.2 Solaris 10

In Solaris 10 FirstSpirit uses the Service Management Facitlity, if available.

Start as `root`:

```
svcadm enable fs4
```

Stopp as `root`:

```
svcadm disable –s –t fs4
```

List processes as `root`:

```
svcs –p fs4
```

3.1.3 AIX

An entry in the file /etc/inittab (which occurs during installation with the identifier „fs4“) can be used to start the FirstSpirit Server under AIX.

Use the following calls to manually start and stopp as root.

Start as `root`:

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

Stopp as `root`:

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

3.1.4 Via a normal user account

The FirstSpirit Server can also be controlled from a normal user account. The default installation creates the user account `fs4`, but deactivates the login for this user account. To activate possible login as `fs4` via SSH or Telnet, just enter the user account `fs4` password. Call as `root`:

```
 passwd fs4
```

After logging in with the user account fs4, use the following calls to control the FirstSpirit Server:

Start as user `fs4`:

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

Stop as user `fs4`:

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

See section 8.6.2 for control via FirstSpirit Server Monitoring.

3.1.5 Generate a stack dump

Via the command:

```
fs4 dump
```

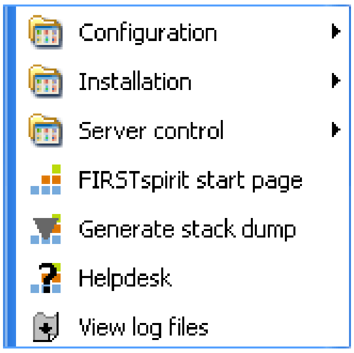
it is possible to create a current thread dump and write it into log/fs-dump-DATE-TIME.log.

FirstSpirit Server Monitoring provides an option for analysis. The thread dumps created via the “Threads” function can be analysed and displayed in a formatted view (see section 8.6.5.5 page 409).

3.2 Windows

The FirstSpirit Server can be controlled via the start menu under Windows operating systems. After successful installation of FirstSpirit, the following menu can be called under:

Start / Programs / FIRSTspirit4:



Configuration:	See section 3.2.5
Installation:	See section 3.2.1 und 3.2.2
Server control:	See section 3.2.3 and 3.2.4
FirstSpirit start page:	See section 3.2.6.3
Generate stack dump:	See section 3.2.6.4
Helpdesk:	See section 3.2.6.1
View log file:	See section 3.2.6.2

3.2.1 Register / Deregister as system service via the start menu

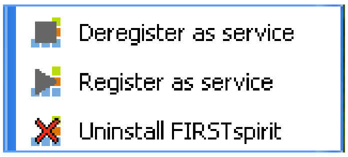


Figure 3-1: Register / Deregister the service

Administrators can configure the FirstSpirit Server as a system service. However, the “System services” component must be installed during FirstSpirit installation (see FirstSpirit Installation Instructions for further installation information). Without system service the server must always be started manually first (see section 3.2.4 page 26).

The “Register service” function is required if the FirstSpirit Server is to be started or stopped via the service (see section 3.2.3).

Only administrators are allowed to register or deregister as a system service via the start menu.

3.2.2 Uninstall via the start menu

FirstSpirit can also be uninstalled via the menu entry “Installation” in the start menu (see Figure 3-1) (see the FirstSpirit Installation Instructions for further information on FirstSpirit uninstallation).

3.2.3 Start /Stop as a system service via the start menu

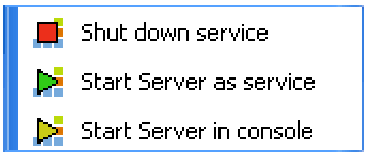


Figure 3-2: Start FirstSpirit Server

It is possible to start the FirstSpirit Server via the Windows start menu.

Administrators can configure the FirstSpirit Server as a system service (see section 3.2.1) and “Start Server as service” or “Shut down service”. Without a registered system service the server can be started in console mode (see section 3.2.4 page 26).

Only administrators can start or stop a system service via the start menu.

Prior to starting and stopping the FirstSpirit Server, activate the maintenance mode (see section 8.6.2.1 page 401).

3.2.4 Start /Stop in console mode

Click on the “Start Server in console” to open a console window in which the FirstSpirit Server is started (see Figure 3-2).

Administrator permissions are not required to start the FirstSpirit Server in console mode.

The server can be stopped via the console window using the shortcut CTRL + C.

Prior to starting and stopping the FirstSpirit Server, activate the maintenance mode (see section 8.6.2.1 page 401).

3.2.5 Edit configuration files via the start menu

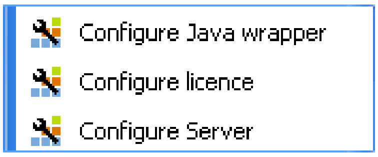


Figure 3-3: Configure via the start menu

Certain FirstSpirit Server configuration files can be displayed and edited via the start menu.

These files should ALWAYS be configured via the FirstSpirit Server Monitoring. Manual configuration should only occur if configuration via the Server Monitoring is no longer possible.

Click on the desired entry to open a text editor for configuration file editing. If the configuration files are changed manually, the server has to be restarted.

3.2.5.1 Configure the Java Wrapper

Click on the entry to open the configuration file `fs-wrapper.conf`, which contains important configuration settings for the server start and the FirstSpirit Server Java system. The configuration file is responsible for starting and stopping the Java process and contains parameters for optimum utilisation of the main memory of the host operating system (see section 4.3.2 page 62). The file should, if possible, be configured via FirstSpirit Server Monitoring (see section 8.6.1.5 page 398).

3.2.5.2 Configure licence

Click on the entry to open the configuration file `fs-license.conf`. See section 4.3.5 (page 87) for a description of the parameters. When inserting a new configuration file fs-license.conf, it is not necessary to restart the server. The file is automatically updated on the server. The file should, if possible, be configured via FirstSpirit Server Monitoring (also see section 8.6.1.2 page 396).

> *If the licence is invalid, the FirstSpirit Server is shut down after 30 minutes. If a valid licence has not been installed, a message is sent to all logged-in FirstSpirit users before the time period elapses.*
>
> *Manipulations to `fs-license.conf` result in an invalid license. If changes are necessary (e.g. IP address change), please contact https://helpdesk.e-spirit.de.*

3.2.5.3 Configure the server

Click on the entry to open the configuration file `fs-server.conf`, which contains important configuration settings for the FirstSpirit Server. See section 4.3.1 (page 33) for a description of the parameters. The file should, if possible, be configured via FirstSpirit Server Monitoring (see section 8.6.1.1 page 396). Certain changes to the configuration file `fs-server.conf` require a server restart.

3.2.6 Further start menu functions

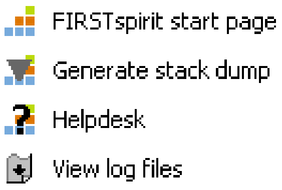


Figure 3-4: Further start menu functions

The start menu offers further functions in addition to controlling and configuring the FirstSpirit Server.

3.2.6.1 Helpdesk

Click on the entry to open a browser window with the login dialog for the FirstSpirit Trouble Ticket System[1]. Login occurs (only after prior registration) via an e-mail address and valid password.

3.2.6.2 View log files

Click on the entry to open the log files:

- fs-wrapper.log: Log file for the output messages of the Java Wrapper (see section 4.3.2.4 page 69 for the configuration).
- fs-server.log: Log file for the output messages of the FirstSpirit Server (see section 4.3.6 page 89 for the configuration).

3.2.6.3 FirstSpirit start page

Click on the entry to open the FirstSpirit start page for starting the FirstSpirit-Clients (see chapter 6 Page 168).

If errors occur while displaying the login window or the FirstSpirit start page, check whether the HTTP port on the server side is already occupied. The same applies when starting in socket mode. The port configuration on the server side should be checked first (see 3.2.5.3 page 28).

3.2.6.4 Generate stack dump

Click on the entry to generate a stack dump for monitoring the current system state of the FirstSpirit Server. These stack dumps can also be generated via FirstSpirit Server Monitoring. Server Monitoring additionally provides functions for analysing the stack dumps (see section 8.6.5.5 page 409).

[1] https://helpdesk.e-spirit.de/

4 FirstSpirit Server configuration

4.1 File system organisation

Introduction of FirstSpirit Version 4.0 has resulted in the restructuring of the directory structures of the server installation, the configuration file names and the web application structure as well as their adaptation to existing standards.

4.1.1 File names

Some of the most important changes to the file names are:

- Renaming of the configuration files:
  - 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
- Renaming of the program files:
  - CMSServer.jar → fs-server.jar
  - CMSClient.jar → fs-client.jar
  - CMSAccess.jar → fs-access.jar
  - OR.jar → fs-or.jar
- Renaming of the log files:
  - CMS_Server.log → fs-server.log
  - Wrapper.log → fs-wrapper.log
  - Deployment.*.log → fs-schedule.*.log

4.1.2 Web applications

The following changes and innovations have taken place for the FirstSpirit web applications:

- Start page and JavaClient (dynamic preview): fs4root
- Staging (local preview system, static preview): fs4staging
- WebEdit and WebEdit preview: fs4webedit
- Server Monitoring: fs4webmon
- Preview (not WebEdit): fs4preview

4.1.3 Directory structures

Reorganisation of the directory structures:

- System areas which are completely overwritten during server update:

~fs4\bin	Start environment incl. required system binaries
~fs4\server	Java environment incl. all libraries
~fs4\web	FirstSpirit web applications

- Configuration areas which can be changed by the user for the system configuration:

~fs4\conf	FirstSpirit configuration files (see section 4.3)

- Data areas:

~fs4\data	All project data incl. modules, etc.

- Temporay file area & exports:

~fs4\backup	Project exports for data backup
~fs4\export	Project exports for interactive export/import processes
~fs4\log	Log files
~fs4\work	Temporary FirstSpirit files
~fs4\web\fs4staging\projects	Staging with web apps
~fs4\web\fs4preview\preview_cache	Preview cache
~fs4\web\fs4webedit\preview_cache	Preview cache WebEdit

4.2 General configuration information

FirstSpirit is configured via configuration files located in the installation directory of the FirstSpirit Server (see section 4.3 page 33).

This chapter describes the structure of the FirstSpirit configuration files and the respective parameters.

There are various possibilities for editing the configuration files:

1) **Via FirstSpirit Server and Project Configuration:** The FirstSpirit Server and Project Configuration facilitates editing of the configuration settings for database connection configuration (fs-database.conf) and login configuration (fs-jaas.conf) (see section 5.3 page 164). All changes to these two configuration files carried out via the Server and Project Configuration are automatically stored in the respective configuration file and updated on the server.

2) **Via FirstSpirit Server Monitoring:** Further configuration settings can be carried out via FirstSpirit Server Monitoring or the JMX Console (see chapter

7.6 page 351 and chapter 9 page 390). Analogue to the Server and Project Configuration, all the changes are automatically rewritten and loaded into the respective configuration file.

3) **Changing the configuration files directly via the file system**: Direct changes to the configuration via the configuration files are only possible if access is available via the file system. If access is possible, never execute changes during operation. It is recommended to always change the configuration files via the administration environments provided by FirstSpirit (see the respective sections for further details). A server restart might be necessary to implement the changes (see section 4.3.1 ff for further information.).

4) **Via JMX Console:** The configuration settings from the configuration files can be partially displayed and changed via the JMX Console (see section 9.5 page 423 for an example).

The following sections refer to the respective configuration possibility within the Server and Project Configuration, the JMX Console or Server Monitoring.

The configuration examples in the following sections contain expressions, such as `Key _1 = ${Key_2}`. A placeholder which can accept the value of another parameter is defined via the expression `${ }`. In the example, the value `Key_2` is allocated to the parameter `Key _1` and possibly complemented by additional specifications. Therefore, it is, for example, possible to compose longer paths from individual, previously defined values (see the example in section 4.3.1.1 page 34).

! *When copying configuration examples from the PDF manuals it is necessary to ensure that all line breaks are correctly copied. If the characters are, for example, incorrectly coded on copying, this can result in problems with the configuration.*

4.3 Configuration files (FirstSpirit Server)

The FirstSpirit Server installation directory contains various configuration files. All configuration files start with the prefix `fs-` and are located in the configuration subdirectory `conf`.

Configuration files for the FirstSpirit Server:

- `fs-server.conf` FirstSpirit Server configuration (section 4.3.1).
- `fs-wrapper.conf` FirstSpirit Server start configuration (section 4.3.2).
- `fs-database.conf` Database connection configuration (section 4.3.3)
- `fs-jaas.conf` Authentication configuration (section 4.3.4).
- `fs-license.conf` FirstSpirit licence configuration (section 4.3.5).
- `fs-logging.conf` Logging configuration (section 4.3.6).
- `fs-webapp.xml` Configuration settings Jetty (section 4.3.7).

4.3.1 FirstSpirit Server configuration (fs-server.conf)

The file `fs-server.conf` located in the subdirectory `conf` of the FirstSpirit Server contains important configuration settings for the server and must be adapted, if necessary.

Changes to the configuration file `fs-server.conf` can be carried out via FirstSpirit Server Monitoring (see section 8.6.1.1 page 396). The changes are subsequently written into the configuration file and updated on the server.

> ! *Certain changes to the configuration file `fs-server.conf` (e.g. changes to the port) require a server restart. Direct changes to the configuration file `fs-server.conf` (via the configuration file and not via FirstSpirit Server Monitoring) always require a server restart!*

The file has the same structure under Windows and UNIX. Please observe the system-independent notation with “/” for the path names. Comment lines can be commenced with # at the beginning of the line; # after a parameter value is not considered a comment character. Parameter values with spaces can be entered directly without a change, i.e. without “ or \. The configuration file `fs-server.conf` is divided into function-related areas. The individual areas and the respective parameters are described below. However, please note that the sequence of entries might vary:

Communication	(section 4.3.1.1 page 34)
Server	(section 4.3.1.2 page 37)
Server Monitoring	(section 4.3.1.3 page 38)
Thread Pool	(section 4.3.1.4 page 40)
Thread Queues	(section 4.3.1.5 page 41)
JAAS	(section 4.3.1.6 page 43)
Web Applications	(section 4.3.1.7 page 43)
Path	(section 4.3.1.8 page 46)
Preview	(section 4.3.1.9 page 48)
Mail	(section 4.3.1.10 page 50)
LDAP	(section 4.3.1.11 page 50)
Storage Engine Properties	(section 4.3.1.12 page 55)
CacheManager	(section 4.3.1.13 page 55)
Internal Database	(section 4.3.1.14 page 56)
WebEdit configuration	(section 4.3.1.15 page 56)
Webstart configuration	(section 4.3.1.17 page 60)

4.3.1.1 Area: Communication

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

`HTTP_PORT:` Http port of the FirstSpirit Server (is required for standard communication between the FirstSpirit-Client and Server).

`URL (optional):` The (optional) URL to FirstSpirit start page. This specified URL will be inserted in automatically sent e-mails, which contain an URL for starting the client. Such mails e.g. are sent in defined workflows by state changes.
This URL is also taken when using the function "Copy FirstSpirit URL" in the JavaClient (menu "Extras").
Normally the URL is automatically identified, but if the server is well-known by multiple host names, this is not possible.

```
URL=http://fs4server.domain.net
```

This parameter must always be set if workflow e-mails are to be used with an external web application server (e.g. Tomcat).

In this case, the following parameters must be set, too. The values will be append to the URL which is determined automatically or defined by the parameter URL and used for establishing the connection.

fs.url.hostname (optional): Use this parameter to specify a hostname for the client connection.

fs.url.socketport (optional): Use this parameter to specify a port which is to be used for the client connection in the socket mode (wildcard %FIRSTspiritSOCKETURL% in workflow e-mails).

fs.url.httpport (optional): Use this parameter to specify a port which is to be used for the client connection in the socket http mode (wildcard %FIRSTspiritURL% in workflow e-mails).

fs.url.usehttps (optional): If HTTPS is to be taken into account when the URL is called, this parameter must be set to true. The default value is false

Example:

This configuration in the 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
```

would result for the wildcard %FIRSTspiritURL% in:

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

would result for the wildcard %FIRSTspiritSOCKETURL% in:

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

In the case where the project is started already, e.g. when using the function "Go to FirstSpirit URL" in the JavaClient, these parameters have normally no effect.

> ! *If the connection settings on the start page are activated (see Chapter 6.3.5.1 page 173), these `fs.url` parameters are not taken into account as it is documented in Chapter 7.3.9 page 227 (item **Evaluation according to priority**). But contrary to this order in evaluation, if the connections settings are deactivated, the corresponding parameters which are defined in the Server properties in the areas "Webstart" and "Start page" will not be taken into account but the `fs.url` parameter will be evaluated.*

`SYMBOLIC_HOSTNAME (optional):` Symbolic host name of the FirstSpirit Server. This host name is only displayed on the start page and has no further functions.

`SOCKET_PORT`: The FirstSpirit Server is waiting for connections for the FirstSpirit-internal Socket protocol on this TCP-Port. This port is used for the internal communication between servlets and the FirstSpirit Server and, if configured, for communication with FirstSpirit JavaClient. On startup this parameter will automatically be inserted in the web.xml of the FirstSpirit Servlets.

`SOCKET_HOST:` Host name for the bind address of the SOCKET_PORT to limit the server to one ip address, if necessary. If no value is transferred, the server connects to all host interfaces.

> ! *The parameter SOCKET_HOST (= the interface to which the socket listener should bind to) can only be used if the parameter HOST (= host name via which the server can be accessed from outside) has been configured correctly too. I.e. it must be mapped to the same network interface of the server.*

`HOST (optional):` Host name or ip address of the FirstSpirit Server. This value is required if an external application server instead of the integrated web server is configured. Via this address the servlets of the external application server are connecting to the socket port of the FirstSpirit Server. This parameter will automatically be created during startup within the files web.xml of the FirstSpirit Servlets.

`INTERNAL_SERVLET_ENGINE:` Value 1= internal web server and servlet engine (Jetty) are started (default value is 1)

Value 0= internal web server and servlet engine (Jetty) are not started (e.g. if an external application server is configured).

`ALLOWED_ENCRYPTIONS (optional):` With this parameter, the user can specify which encryption they have to use in their connection settings (see Chapter 6.3.5.1 page 173). The value `1` is set for use of TLS; value `2` is set for use of DH_ARC4. The value `0` must be set if encryption is not to be used. Any combinations of the parameters `0`, `1` and `2` are also possible. Example:
`ALLOWED_ENCRYPTIONS=1,2`
If the encryptions set by the user does not match the encryption specified here, when the user logs onto the JavaClient or to apply the server and project configuration, an error message appears that the connection parameters have been incorrectly set; communication is not possible between the client and server.

> *If Websphere is used as the application server for the web application fs4root, TLS encryption (parameter value 1) cannot be used for the parameter `ALLOWED_ENCRYPTIONS`. In this case it is advisable to use RC4 encryption (parameter value 2) for SOCKET or HTTP or to use HTTPS. No additional encryption is required for HTTPS so that the parameter value 0 must then be used.*

4.3.1.2 Area: Server

```
###########################

# server

###########################

backup_files=50
cyclicReferenceSaveTime=60
```

`backup_files`: Number of backup versions stored to the internal structure files. This exclusively applies to the administration and structure data and not to the content data. All content data is versioned separately. Increasing the value enhances the error correction possibilities, but also leads to an increased occupancy of the disk memory on the computer (default value is 50).

`cyclicSaveTime`: States the time in seconds after which changes, e.g. to the history, statistics, tasks or user settings, are saved. Cyclic saving is only executed if the respective data has been changed (default value is 60 sec.).

`HELP_URL:` Defines the relative URL under which the FirstSpirit online help is stored (see section 4.2 page 31 for the notation ${}). Example: `HELP_URL=${WEBAPP_ROOT_NAME}/help`.

`JNLP_SERVLET_URL:` The value defined here is required for generating the FirstSpirit URLs which are generated as references for starting the client in e-mails.
Example:
`JNLP_SERVLET_URL=${URL}/start/FIRSTspirit.jsp`

`DTO_LRU_SIZE:` The parameter determines the size of the DTO cache for a server project. The last used tree objects of a project are stored here. The value defines the number of objects which are administered in the cache (default value is 512 stored elements).

`SERVICES:` Via this parameter it is possible to define the system services. (These services can also be started during the server start).

`WORKFLOW.TASK.CACHE:`
`WORKFLOW.MODEL.CACHE:` The type which is defined here provides information about how long workflow models or tasks remain in the cache. A distinction is made between `WEAK` and `SOFT`: if `WEAK` is indicated, the objects will be deleted from the cache as soon as they are no more used. If `SOFT` is indicated, the objects will, depending on the used VM, remain in the cache until there is insufficient memory space. (The type `WEAK` is usually advantageous for a large memory.) The size of the LRU in KB must be attached to the type, separated by an underscore (default values: `workflow.task.cache=SOFT_1024` and `workflow.model.cache=SOFT_128`).

4.3.1.3 Area: Server Monitoring

```
###########################
# Server Monitoring - Ajax
###########################
AJAX_DATA_SYNC_TIMEOUT=20
AJAX_IMAGE_RELOAD_TIMEOUT=60

# ;-separated mail-addresses for mail-system-monitoring-site
SYSTEM_MONITORING_MAIL_RECIPIENTS=admin@yourdomain.net
```

`AJAX_DATA_SYNC_TIMEOUT:` Defines the time interval in seconds during which the dynamically updated data (via Ajax) is renewed within Server Monitoring (default value is 20).

`AJAX_IMAGE_RELOAD_TIMEOUT:` Defines the time interval in seconds during which the dynamically updated statistics (chart) (via Ajax) are renewed within Server Monitoring (default value is 60).

`SYSTEM_MONITORING_MAIL_RECIPIENTS:` E-mail addresses of users who should be informed about the system state of the FirstSpirit Server. “;“ is used as separator between the addresses.

4.3.1.4 Area: 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=
```

In version 4.2R4, the default values for the size of the thread pool and thread queue were adjusted for an improved performance.
Therefore, following installation or a change of hardware, the FirstSpirit Server is now automatically scaled to the number of existing computing cores. The number of available processors is determined via

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

`ThreadPool.minSize`: Defines the minimum size of the limited thread pool (see Figure 9-13 page 429).

`ThreadPool.maxSize`: Defines the maximum size of the limited thread pool and thus the largest possible number of tasks to be executed simultaneously (see Figure 9-13 page 429). The more processors a server has, the higher the value for the `ThreadPool.maxSize` can be configured.
The value of the actually executed threads can be higher, since this limitation does not apply to high priority tasks (unlimited thread pool).

If no explicit values are given for the following parameters

- `ThreadPool.minSize` (minimum size of the limited thread pool) and
- `ThreadPool.maxSize` (maximum size of the limited thread pool and therefore the largest possible number of tasks which can be performed in parallel)

values are automatically used which depend on the number of available processors.

4.3.1.5 Area: 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   maxmimum 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
```

Tasks are executed via the ExecutionManager, which administers various differently classified queues (see section 9.8 page 427). Some of the queues can limit the number of active tasks via parameters. Furthermore, for queues with the classification `BOUNDED`, it is possible to limit the maximum capacity of the queue.

`ThreadQueue.LOW.maxRunning`: Only a small number of the resource-intensive tasks in this queue should be executed simultaneously. The number of simultaneously executable tasks can be configured via parameter `maxRunning` (default value is 2 threads).

`ThreadQueue.DEFAULT.maxRunning`: The number of the simultaneously executable tasks of this queue can also be limited via parameter `maxRunning`. Since these tasks are not very resource intensive, the value can be higher than the value defined in `ThreadQueue.LOW`.

`maxRunning` (default value is 25 threads).

`ThreadQueue.DEFAULT.queueCapacity`: The maximum capacity of a queue classified with *`DEFAULT`* can be limited.

`ThreadQueue.BOUNDED.maxRunning`: The queue classified with `BOUNDED` can be configured via two parameters. Via parameter `maxRunning` it is possible to limit the number of active tasks (default value is 25 threads).

`ThreadQueue.BOUNDED.queueCapacity`: It is also possible to limit the maximum capacity of a queue classified with `BOUNDED` via parameter `queueCapacity`. If the value configured under `queueCapacity` has been reached, further tasks are rejected by the server. This means that if there is a high load on the server, only a certain number of tasks are placed in the queue (the number of tasks defined under `maxRunning` is directly processed) and further tasks are momentarily rejected. After a specific time period, a new attempt is made to place these tasks in the queue (default value is 50 threads).

If no explicit values are given for the following parameters

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

values are automatically used which depend on the number of available processors.

! *The values of the `ThreadQueue.LOW.maxRunning` and `ThreadQueue.LOW.queueCapacity` parameters are not dependent on the number of computing cores and cannot be overwritten with "empty" values.*

! *The value for `ThreadQueue.<name>.maxRunning` must be smaller than the value for `ThreadPool.maxSize` (see Chapter 4.3.1.4 page 40).*

4.3.1.6 Area: JAAS[2] – login procedure configuration

```
##########
# 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
```

Via this area it is possible to define important configuration settings for the login process for various FirstSpirit applications (e.g. login at the JavaClient or preview request). To achieve this, a number of different authentication modules are available (e.g. SSO). In the JAAS area a mapping of the application possibility to the connection modules which are configured in the configuration file fs-jaas.conf is established (see section 4.3.4 page 72). A convenient possibility for configuring this area is provided by the Server and Project Configuration (see section 7.3.13 Page 231) and via Server Monitoring (see section 8.6.1.8 page 401).

4.3.1.7 Area: 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
```

[2] Java Authentication and Authorization Service (Information: http://java.sun.com/products/jaas/)

```
# 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 to the start page web application. The value stored here is required for mapping within the internal servlet engine (also see section 4.3.7.2 page 94). The value always has to start with a "/" followed by the symbolic name of the respective web application. The URL is required for references to various FirstSpirit applications (default value: `/`).

`WEBAPP_WEBEDIT_URL`: URL to the WebEdit web application (fs4webedit) within the start page. The value stored here is required for mapping within the internal servlet engine (also see section 4.3.7.2 page 94). The value always has to start with a "/" followed by the symbolic name of the respective web application. The URL is required for references to various FirstSpirit applications
(default value: `/${WEBAPP_WEBEDIT_NAME}`).

`WEBAPP_WEBMON_URL`: URL to the "Server Monitoring" web application (fs4webmon) within the start page. The value stored here is required for mapping within the internal servlet engine (also see section 4.3.7.2 page 94). The value always has to start with a "/" followed by the symbolic name of the respective web application. The URL is required for references to various FirstSpirit applications
(default value: `/${WEBAPP_WEBMON_NAME}`).

`WEBAPP_STAGING_URL`: URL to the web application FirstSpirit Staging (fs4staging). The value stored hiere is required for mapping within the internal servlet engine (also see section 4.3.7.2 page 94). The value always has to start with a "/" followed by the symbolic name of the respective web application. The URL is required for references to various FirstSpirit applications (default value: `/${WEBAPP_STAGING_NAME}`).

`WEBAPP_PREVIEW_URL:` URL to the web application FirstSpirit Preview (fs4preview). The value stored hiere is required for mapping within the internal servlet engine (also see section 4.3.7.2 page 94). The value always has to start with a "/" followed by the symbolic name of the respective web application. The URL is required for references to various FirstSpirit applications
(default value: `/${WEBAPP_PREVIEW_NAME}`).

`WEB_DIR:` Path specification to the FirstSpirit web directory for using all FirstSpirit web applications ((fs4root, fs4preview, fs4staging, fs4webedit, fs4webmon) (see section 4.2 page 31 for the notation `${}`).

`WEBAPP_ROOT_PATH:` Path specification to the project directory, the directory in the file system in which the FirstSpirit Server stores the FirstSpirit start page (fs4root) (see section 4.2 page 31 for the notation `${}`).

`WEBAPP_WEBEDIT_PATH:` Path specification to the directory in the file system in which the FirstSpirit server stores the web application WebEdit (fs4webedit) (see section 4.2 page 31 for the notation `${}`).

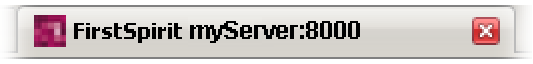
`WEBAPP_WEBMON_PATH:` Path specification to the directory in the file system in which the FirstSpirit server stores the web application Server Monitoring (fs4webmon) (see section 4.2 page 31 for the notation `${}`).

`WEBAPP_STAGING_PATH:` Path specification to the directory in the file system in which the FirstSpirit server stores the web application FirstSpirit Staging (fs4staging). In this directory the FirstSpirit server stores all generated projekt files too (see section 4.2 page 31 for the notation `${}`).

`WEBAPP_PREVIEW_PATH:` Path specification to the directory in the file system in which the FirstSpirit server stores the web application Preview (fs4preview). In this directory the FirstSpirit server stores the generated preview pages too. This parameter can be also defined using the application for the Server and Project configuration (see Chapter 7.3.3 page 215).

`WEBAPP_ROOT_TITLE:` Using this parameter the title of the FirstSpirit start page can be changed. The default value is `FirstSpirit ${HOSTNAME}:${HTTP_PORT}`. With this value the start page of the

FirstSpirit server is displayed in the browser like this:



FirstSpirit myServer:8000

If the symbolic host name of the FirstSpirit Server, which is defined by the parameter `SYMBOLIC_HOSTNAME` (see Chapter 4.3.1.1 page 34), is to be used if `WEBAPP_ROOT_TITLE` is not set,
`WEBAPP_ROOT_TITLE=FirstSpirit ${SYMBOLIC_HOSTNAME}`
can be set for example. There will be no automatic replacement, because `SYMBOLIC_HOSTNAME` is an optional parameter which can be empty.

`CLIENTJAR_URL:` URL of the FirstSpirit-Client Jar files.

4.3.1.8 Area: Path

```
###########################
# paths
###########################
BACKUP_PATH=${cmsroot}/backup
BATCHPATH=${cmsroot}/data/scripts

# base path for "rollout client applications"
CLIENTAPP_PATH=${cmsroot}/data/clientapp
```

`BACKUP_PATH:` Path specification to the backup directory, the directory in the file system in which the FirstSpirit Server should store the backups.

`BATCHPATH:` The parameter defines the path to the script files which can, e.g., be executed during deployment.

`CLIENTAPP_PATH:` The parameter is solely required for the roll-out process for native client applications (from FirstSpirit Version 4.2) (see Chapter 4.9.1 page 153). Default value: `~FS42\data\clientapps`

`CLIENT_HOME_DIR:` (from FirstSpirit Version 4.2R2) Details of path to the directory in the file system in which the Client applications are to be filed (see Chapter 4.9.2 page 153). Absolute (e.g. `CLIENT_HOME_DIR=C:/test`) or relative paths can be used. For relative paths, the `~` symbol can be used as a wildcard at the start of the path, e.g. `~/myclientapps`. `~` is then replaced by the operating system-specific, current user home directory.

If the parameter is not given, as a default, Client applications are rolled out in the `\.firstspirit` directory in the operating system-specific user home directory. **From 4.2R4** client applications are rolled out by default into an individual directory, depending on the used FirstSpirit Version. The name of the directory contains the number of the respective FirstSpirit major, minor and release version, e.g. `\.firstspirit_4.2R4`.

! *The directory used until 4.2R2 `\.firstspirit` will not be deleted by the system, but it must be deleted manually (e.g. to free memory).*

Default value **up to and including 4.2R2**:
`CLIENT_HOME_DIR=~/.firstspirit`
from 4.2R4:
`CLIENT_HOME_DIR=~/.firstspirit_${FS_MAJOR}.${FS_MINOR}${FS_RELEASE}`

`CLIENT_HOME_DIR_WINDOWS`: Details of path to the directory in the file system in which the Client applications are to be rolled out on workstations with Windows operating system. Example:
`CLIENT_HOME_DIR_WINDOWS=C:/.test`
If `CLIENT_HOME_DIR_WINDOWS` only is given, the Client application is rolled out in the operating system-specific user home directory.

`CLIENT_HOME_DIR_LINUX`: Details of path to the directory in the file system in which the Client applications are to be rolled out on workstations with Linux operating system (cf. parameter `CLIENT_HOME_DIR_WINDOWS`).

`CLIENT_HOME_DIR_MAC`: Details of path to the directory in the file system in which the Client applications are to be rolled out on workstations with Macintosh operating system (cf. parameter `CLIENT_HOME_DIR_WINDOWS`).

`CLIENT_HOME_DIR_AIX`: Details of path to the directory in the file system in which the Client applications are to be rolled out on PCs/workstations with AIX operating system (cf. parameter `CLIENT_HOME_DIR_WINDOWS`).

`CLIENT_HOME_DIR_SOLARIS`: Details of path to the directory in the file system in which the Client applications are to be rolled out on workstations with Solaris operating system (cf. parameter

CLIENT_HOME_DIR_WINDOWS).

4.3.1.9 Area: Preview

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

`preview.cacheMediaFiles:` The value `false` is set by default. The default value should only be changed if an external servlet engine is used. If the value is set to `true`, media files are buffered in the servlet engine in addition to the preview pages.

`preview.internalDelivery:` Specify a comma-separated list of file extensions here which should be directly provided by the servlet engine. Files which are not included in the list are provided to the web server via an internal redirect. (This can, e.g., be an Apache web server which accepts file processing, e.g. PHP).
Up to FirstSpirit version 4.1 the default value is `html,htm,txt,xml,pdf,jsp,shtml,ini`. From FirstSpirit version 4.2 the default value is `*`. This can be used to output all files with an extension which is not in the list of the parameter `preview.externalDelivery` by the servlet engine.

`preview.externalDeliveryURL:` This parameter can be used to specify the URL to the external web server, which is used for file types not defined in the preview.internalDelivery parameter (e.g. PHP or ASP). The URL is made up of the server name and the port, which is configured in Chapter 4.5.1 in the "Virtual Webserver" section (page 102). Example:

`preview.externalDeliveryURL=http://fs4.yourdomain.net:80`.
This parameter is empty as a default and must be set if necessary.

`preview.cacheTimeout:` The generated pages which are buffered in the `cacheDir` are only valid for the time interval defined (in seconds) here. If the interval has elapsed, the preview pages are (when requested) recreated and buffered again. All files older than the defined interval are removed from cache (default value is 120 seconds).

New parameters in FirstSpirit version 4.2 and higher:

`preview.externalDelivery:` Use this parameter to define a comma-separated list with file extensions which should be provided by an external web server and **not** by the servlet engine. This parameter will only be taken into account if the parameter `preview.internalDelivery` is set to `*`. This parameter is empty by default and must be set if requires.

`preview.cacheFileWithTimestamp:` Using this parameter files can get a timestamp in their file name. This is important e.g. when using IBM Websphere to re-compile JSP files correctly if the content of the file has been changed after a previous compilation. For this purpose, a comma-separated list with file extensions which are to obtain the timestamp in the file name must be specified for `preview.cacheFileWithTimestamp`, e.g. `preview.cacheFileWithTimestamp=jsp,jsf`. To provide the timestamp for all file names the parameter can be set to `*`. The parameter is empty by default. If IBM WebSphere is used this parameter must be set.

4.3.1.10 Area: Mail

```
############################
# mail
############################
#mail.default-recipient=
#mail.sender=
#mail.smtp=

LICENSE_EXPIRATION_WARNING_DAYS=30
LICENSE_EXPIRATION_MAIL_ADDRESS=info@e-spirit.de
```

`mail.smtp:` Specifies an SMTP mail server. It is required since the server can send a requested email with result information after website provision or generation.

`mail.default-recipient:` Specification of a default e-mail address.

`mail.sender:` Specification of an e-mail address used as sender for all FirstSpirit Server e-mails. If no e-mail address has been specified here, a default value is used.

`LICENSE_EXPIRATION_WARNING_DAYS:` The parameter defines how many days before FirstSpirit licence expiry a daily expiration reminder is to commence. If the specified value is “0”, no email is sent (default value is 30 days).

`LICENSE_EXPIRATION_MAIL_ADDRESS:` The parameter defines to which e-mail address the licence expiration reminder is to be sent. If no e-mail address has been defined, the email is sent to the administrator (see parameter `ADMIN_MAIL_ADRESS` in section 4.3.1.10 page 50).

4.3.1.11 Area: LDAP

Various LDAP configurations (so-called sections) can be created in the FirstSpirit Server. The name of a section is defined via the entry in the configuration file fs-jaas.conf (see section 4.3.4 page 72). Configuration of a section occurs in the configuration file fs-server.conf (LDAP area). The section name (from fs-jaas.conf) is specified in front of each configuration parameter. The section name (from fs-jaas.conf) is given in front of each configuration parameter which must conform to the form "LDAP_n", whereby n is the section number which is consecutively numbered beginning with 1. If only 1 LDAP section is used, the section can also be called "LDAP".

```
LDAP_n.parameter=value
```

The names of the LDAP attributes must be entered in upper/lower cases as they are given in the LDAP directory!

In the following configuration example for linking to the LDAP server of the Microsoft Active Directory (using `LDAP.AUTHENTICATION= SEARCH_BIND`), "LDAP" has been chosen as the section name. Other section names can also be chosen, depending on the configuration of the fs-jaas.conf (cf. Chapter 4.3.4), and several different LDAP sections can also be defined simultaneously.

```
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:` Labelling of the respective LDAP section, e.g. the domain name. The description is shown in the “Edit user” dialog (see section 7.2.4.2 page205).

`LDAP.HOST_URL:` LDAP URL from the respective LDAP section in the form ldap://hostname (if LDAP.SSL=false) or ldaps://hostname (if LDAP.SSL=true). For increasing the reliability more than one LDAP server can be given, which must provide both the same LDAP data.

`LDAP.SSL:` Activate (value: true) or deactivate (value: false) the encrypted SSL transfer here (see section 4.4.3).

`LDAP.AUTHENTICATION:` There are various ways to login at the server. Possible values are:

- `BIND:` Name and password are sent to the LDAP server. The "Distinguished Name" (DN), i.e. the unique key for user authentication, has to be known in the LDAP server. If the DN exists, the transferred password is checked with the help of the "Bind" operation.
  BIND can only be used if the LDAP-DNs of the user accounts all

lie within the same LDAP folder. In the example for parameter `LDAP.BIND.DN` (see below), this is the folder with the DN `ou=Benutzer,ou=Dortmund,dc=e-spirit,dc=de.`
If the user accounts are spread over different folders, either SEARCH_BIND must be used or a separate LDAP section must be entered in fs-server.conf for each of the user folders.

- `SEARCH_BIND`: If the "Distinguished Name" (DN) of a user is unknown or if user accounts are entered in different subtree of the LDAP tree, it is possible to search for it within a subtree of the LDAP server. A search filter has to be defined for this task. Example:

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

This filter searches for all entries in the LDAP tree in which the attribute "uid" matches the entered login name. Start node is the node with the DN " ou=department,dc=mycompany,dc=com". The login is successful once a user account can be found which is appropriate to the SEARCH.FILTER and the commited user password is correct.

- `SEARCH_COMPARE`: This option works similar to `SEARCH_BIND`. However, the password attribute is not used for authentification but any other LDAP attribute. Example:

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

In this case, the entered password has to match the content of the "mail" attribute.

> ! *When using `LDAP.AUTHENTICATION=SEARCH_BIND` or `LDAP.AUTHENTICATION=SEARCH_COMPARE`, `LDAP.SEARCH.BIND_DN` and `LDAP.SEARCH.BIND_PASSWORD` (see below.) must be indicated generally. The indication of this parameters is not required only in the case if the LDAP server can be retreived without authentication. In fact, this does not occur on live systems.*

`LDAP.BIND.DN`: DN of the user accounts which can be used for login to FirstSpirit. This parameter is only reasonable in combination with `LDAP.AUTHENTICATION=BIND`. `$USER_LOGIN$` is entered as wild-card for the FirstSpirit user name. Example:

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

Complete configuration example using `LDAP.AUTHENTICATION=BIND` for Active Directory:

```
LDAP_1.NAME=e-Spirit
LDAP_1.HOST_URL=ldap://ldapserver1 ldap://ldapserver2
LDAP_1.SSL=FALSE
LDAP_1.AUTHENTICATION=BIND
LDAP_1.BIND.DN=cn=$USER_LOGIN$,ou=user,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 of a technical user account, which is used to browse through the LDAP server to find a DN of a FirstSpirit user to be logged in.

`LDAP.SEARCH.BIND_PASSWORD`: Password for the LDAP-DB of the technical user account used for SEARCH.BIND_DN.

`LDAP.SEARCH.BASE_DN`: The parameter defines the start node of the search for the LDAP-DN of the FirstSpirit user to be logged in.

`LDAP.SEARCH.FILTER`: A search filter is defined via the parameter. The filter: `SEARCH.FILTER=(cn=$USER_LOGIN$)`
searches, for example, for all entries in the LDAP tree in which the attribute "cn" matches the entered login nameused in FirstSpirit. Start node is the DN given at SEARCH.BASE_DN.

`LDAP.IMPORT_USER`: Besides pure authentication, it is possible to transfer any LDAP attribute into the user attributes of a CMS user. To achieve this, parameter `LDAP.IMPORT_USER` has to be set to the value `TRUE`.

`LDAP.IMPORT_USER.LOGIN_ATTRIBUTE`: This allocation accepts the login name of an LDAP user for a FirstSpirit user. This allocated LDAP name is automatically imported after initial login of the respective user. If there are more than one LDAP section (LDAP_1, LDAP_2, …), i.e. if the user authentication is carried out using different user domains, sAMAaccountName as attribute is mostly not unique for example under Active Directory via all domains. Hence, the complete user principal userPrincipalName which also contains the domain name should be entered here.
If an invalid value is given for this parameter (e.g. LDAP attribute

returns an "empty" value, invalid LDAP attribute, etc.), this is logged from 4.2R2 in the server log file:

```
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`: This allocation accepts the user name of an LDAP user for a CMS user. This allocated LDAP name is automatically imported after initial login of the respective user.

`LDAP.IMPORT_USER.EMAIL_ATTRIBUTE`: This allocation accepts the email address of an LDAP user for a CMS user. This allocated LDAP email address is automatically imported during initial login of the respective user.

`LDAP.IMPORT_USER.GROUP_ATTRIBUTE`: The LDAP group attribute can be used to automatically assign a user to a specific FirstSpirit group of a FirstSpirit project on the basis of their group membership in LDAP.
The name of the LDAP attribute is given, the LDAP-DN that contains LDAP groups of which the respective user is a member. The attribute is read out again each time the user signs on, in order to copy their group membership into FirstSpirit. All FirstSpirit groups marked as being external, whose "external name" assigned in FirstSpirit matches the LDAP-DN of the LDAP group are assigned to the user account. Instead of an LDAP-DN, any character string can also be used, although LDAP servers usually map group membership via DNs.

`LDAP.IMPORT_USER.PHONE_ATTRIBUTE:` This allocation binds the telephone number of an LDAP user to a CMS user. The allocated LDAP telephone number is automatically imported during initial login of the respective user.

`LDAP.IMPORT_USER.ABBREVIATION_ATTRIBUTE:` This allocation binds the abbreviation of an LDAP user to a CMS user. The allocated LDAP abbreviation is automatically imported during initial login of the respective user.

See section 4.4 page 96 for further information on connecting an LDAP server.

4.3.1.12 Area: Storage Engine Properties

```
###########################
# storage engine properties
###########################
repository.je.checkpointer.wakeupInterval=10000000
repository.je.maxMemoryPercent=25
```

All Berkeley DB properties can also be used in the configuration file `fs-server.conf`. The prefix “repository.” has to be specified in front of each respective property for this task.

For the respective properties see the documentation about the Berkeley DB[3].

> ! *If the file `fs-server.conf` is changed, the server has to be restarted (see section 4.2 page 31).*

4.3.1.13 Area: CacheManager

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

In this area it is possible to set either the absolute or a relative size (in percent) for the cache memory requirement. The value should not be set to 0.

`CACHE_SIZE`: Absolute cache size in bytes. Valid values are, e.g., 456458345, 128m or 4096k.

`CACHE_PERCENT`: Cache size from the –Xmx value in percent. Valid values are, e.g., 30 or 0.5. The value 40 is set by default.

[3] http://www.oracle.com/database/berkeley-db/index.html

4.3.1.14 Area: Internal database

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

`internalDB.port:` TCP port, on which the JDBC connector of the internal database system is started. Default value: 0.

`internalDB.host:` IP address or host name to which the JDBC connector is to be linked. Default value: "", all addresses are used.

The FirstSpirit server has its own simple relational database system (Apache Derby), which is available immediately after the server has been installed. However, this database is not suitable for productive use and should therefore only be used for tests.

External clients or web applications of external servers can access the internal database (Derby) via the bind address. If the value for the `internalDB.port` has been set to 0, this function is deactivated. This has been configured by default to close a potential security gap.

The access data for the database is located in fs-database.conf (see section 4.3.3 page 71).

An example how to configure the Derby database for using it from external processes (e.g. web application with FirstSpirit module Integration in the external application server) can be found in Chapter 4.8.7.6 page 150.

4.3.1.15 Area: 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:` The parameter is used to specify a basic theme which is applied if a project has not yet been selected (e.g. login, project selection). The theme for a project is defined and evaluated via the FirstSpirit Project Configuration as soon as the user has selected the respective project (see section 7.4.14 page 284).

`webedit.errorPage:` Via this parameter it is possible to define a page (.jsp) which is displayed if an error (user or system error) has occurred while working with WebEdit.

`webedit.noLogin:` Via this parameter it is possible to prevent the display of the login page (value=0) if, e.g., the login process occurs automatically via the SAP portal.

`webedit.webSSO:` Via this parameter it is possible to define whether the login process for WebEdit is possible via SSO (Single Sign On) (value=1) or not (value=0) (default value 0).

`webedit.showReferenceName:` Via this parameter it is possible to define whether objects, e.g. pages, are displayed in WebEdit with the unique reference name (value=1) or with language-dependent names (value=0). If language-dependent names are used, additionally activate the check box “Display name language-dependent” in the Project Configuration (see section 7.4.2 page 257).

`webedit.sitestoreWorkflow:` Via this parameter it is possible to define whether a workflow started via the WebEdit bar is executed on a page reference of the Site-Store (value=1) or on a page of the Page-Store (value=0) (see section 5.2.7.1 page 163).

`webedit.serviceUrl`: Via this parameter it is possible to define an email address for the error page to which a message, including an error report, is sent if an error occurs in the WebClient (see `webedit.errorPage`). The default value `mailto:please_set_email_url_in_fs-server.conf` must be configured when using WebEdit.

`webedit.plainPassword`: Via this parameter it is possible to define whether the password is sent as plain text in the login page (value=1) or not (value=0) (see section 5.3.3 page 165).

`webedit.additionalLogoutUrl`: Here it is possible to specify an additional URL (without URL parameters) for the WebEdit logout process.

`webedit.preview.noPreviewUrl`: Preview images which can be defined in the page and section templates are stored in the dialogs which are opened via the WebEdit Quick Edit Bar when creating pages and sections. If no image has been stored for a preview in the Template-Store, the image defined in parameter `webedit.preview.noPreviewUrl` is displayed. The specification is relative to the application root of WebEdit.

`webedit.sessionCache.lru_size`: The parameter determines the size of the LRU[4] cache for each individual WebEdit session. The LRU cache is the cache for all FirstSpirit objects referenced in the project. The value defined here states for how long the referenced objects will remain in the cache. A distinction is made between "weak references" (value < 0) which are directly deleted from the cache as soon as they are no longer referenced and "soft references" (value >= 0) which, depending on the used VM, remain in the cache until there is insufficient memory space. (Weak references are usually advantageous for a large memory).

`webedit.globalCache.lru_size`: This parameter determines the size of the DTO[5] cache or all WebEdit sessions. The value given here defines the number of store elements which can be stored in the DTO cache. The pure data objects of a project are stored in the DTO cache. All users work on the same data objects (in contrast to the project objects in the Stores which are only valid locally during a session). The cache is thus divided among all WebEdit sessions (global cache).

[4] LRU: Least recently used

[5] DTO: Data Transfer Object

`web.sessionCache.lru_size`: This parameter determines the size of the cache for the project objects of a web session which are, e.g., required for preview creation. The value given here defines the number of store elements which can be stored in the cache.

`webedit.expirationMode`: This parameter determines the cache mode. The value "expires" currently solely causes the updating of the thumbnail overviews in WebEdit. The following values are possible:

- Value "none":
  No "Pragma", "Cache Control" and "Expires" header
- Value "default" (standard setting):
  "pragma"="no-cache", "Cache-Control"="no-cache", "Expires"="0": No "Pragma" and "Cache Control" header, "Expires" contains a date value conforming to RFC 1123. The value is calculated from the time of delivery plus the given value in hours (GMT, i.e. for our latitude CEST-2 or CET-1).

4.3.1.16 Area: 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
```

The following parameters are valid from FirstSpirit Version 4.2R4:

`hdd.directories`: Use this parameter to indicate the directories which are to be controlled. Default value is the FirstSpirit base directory, i.e. `hdd.directories=${cmsroot}`. This will be used if no value is given explicitly. See also Chapter 7.3.1 page 210, option "Directories for disk space check (comma separated)" and the following.

`hdd.limit`: Use this parameter to indicate the percentage from which a warning e-

mail is to be sent to the server administrator. You can use values between `1` and `99`. If no warning e-mails are to be sent, the value can be set to `-1`.
Default value is 90%, i.e. `hdd.limit=90`. This will be used if no value is given explicitly or if the given value is not between `1` and `99` or `-1`.

`hdd.limit.active`: This parameter can be used to prevent the dispatch of warning e-mails. To do this `hdd.limit.active=false` must be set. As a default, however, dispatch is activated at the given percentage.

`hdd.shutdown`: Use this parameter to indicate the percentage from which a warning e-mail is sent to the server administrator and the server will be shut down. You can use values between `1` and `99`. If the server is not to be shut down at all, the value can be set to `-1`. If the given value is lower or equal to the value of `hdd.limit`, `hdd.shutdown` will be set 5% higher than the value of `hdd.limit` by the system.
Default value is 95%, i.e. `hdd.shutdown=95`. This will be used if no value is given explicitly or if the given value is not between `1` and `99` or `-1`.

`hdd.shutdown.active`: This parameter can be used to prevent shutting down of the FirstSpirit server. To do this `hdd.shutdown.active=false` must be set. As a default, however, shutdown is activated at the given percentage.

4.3.1.17 Area: Webstart configuration

The Sun Java Runtime Environment (JRE) Version 1.5.0. or higher (Version 1.6.0 is recommended) is required to start the FirstSpirit server and project configuration and the FirstSpirit JavaClient; it contains Java Webstart. (The JRE is usually automatically installed when JDK 1.5.0. or 1.6.0. is installed.) It is configured via JNLP files:

- global (server-wide) via the FirstSpirit server and project configuration (see chapter 7.3.8 Seite 226) or via parameters in the configuration file fs-server.conf (see below)
- user-specific via the connection settings on the FirstSpirit start page (see chapter 6.3.5.1 Seite 173).

Webstart configuration for starting the FirstSpirit JavaClient: The parameters configured here affect all Quick Start entries on the start page of the type JavaClient

JAVA, unless other parameters have explicitly been defined for the entry in the "start page" area (see chapter 7.3.9 Seite 227).

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

The configuration options correspond to the user-specific Webstart configuration (see chapter 6.3.5.1 Seite 173).

> ! *If Websphere is used as the application server for the web application fs4root, TLS encryption (parameter value 1) cannot be used for the parameter webstart.client.encryption. In this case it is advisable to use RC4 encryption (parameter value 2) for SOCKET or HTTP or to use HTTPS. No additional encryption is required for HTTPS so that the parameter value 0 must then be used.*

Server and Project Configuration tab: Configuration for the Server and Project Configuration application.

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

The configuration options correspond to the user-specific Webstart configuration (see chapter 6.3.5.1 Seite 173).

> ! *If Websphere is used as the application server for the web application fs4root, TLS encryption (parameter value 1) cannot be used for the parameter webstart.admin.encryption. In this case it is advisable to use RC4 encryption (parameter value 2) for SOCKET or HTTP or to use HTTPS. No additional encryption is required for HTTPS so that the parameter value 0 must then be used.*

4.3.2 Configuration of the Java VM and the Java Wrapper (fs-wrapper.conf)

The file `fs-wrapper.conf` is located in the FirstSpirit Server subdirectory `conf` and contains important configuration settings for the server start and the Java system of the FirstSpirit Server.

The configuration file is responsible for starting and stopping the Java process and contains parameters for optimum utilisation of the main memory of the host operating system.

Changes to the configuration file `fs-wrapper.conf` can be carried out via FirstSpirit Server Monitoring (see section 8.6.1.5 page 398). The changes are subsequently written into the configuration file and updated on the server. If the modification is not valid, an error is displayed in Server Monitoring and the save operation will not be executed:

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

If file system access is available, the configuration file can also be changed directly.

For a complete description of all Java Wrapper parameters and further information see: http://wrapper.tanukisoftware.org/doc/english/properties.html.

The file has the same structure under Windows and UNIX. Please observe the system-independent notation with "/" for the path names. The individual areas and the corresponding changeable parameters are described below. The sequence of the parameters in the file is arbitrary.

- Configuration of Java-VM (section 4.3.2.1)
- General parameters (section 4.3.2.3)
- Logging (log files) (section 4.3.2.4)
- System service under Windows (section 4.3.2.5)

The file created during installation with default values can be found in section 12.2.

Each change to the configuration file `fs-wrapper.conf` requires a server restart.

A line with parameter values within the configuration file fs-wrapper.conf may not contain comments, e.g.:
`wrapper.startup.timeout=30 # Comment`

4.3.2.1 Configuration of Java VM

As FirstSpirit runs as an application within a virtual Java machine (Java VM) and the presently available Java VMs do not have any dynamic memory allocation toward the operating system, several parameters concerning the Java VM's memory allocation must be configured for optimal scaling of the use case.

The memory area usable for FirstSpirit is the so-called "Heap" of Java-VM. This heap should be set as large as possible, but should not be larger than the free main memory in the operating system. As default 75% of RAM should be set if no other services are run simultaneously with FirstSpirit on the server.

A heap which is considerably larger than 10 GByte can not be managed without problems with the parameters which are documented here. If this dimension is required in fact, the behaviour of the Garbage Collection via JMX must be analysed by means of jconsole or VisualVM and the Java VM parameters must be customised to the use case in detail.

If a heap is to be configured which is larger than 10 GByte, please ask the manufacturer before (e-Spirit AG), because in such a case further, special parameters for configuring the Garbage Collector are mostly required.

The size of the heap is set using the `wrapper.java.initmemory` or `wrapper.java.initmemory.percent` and `wrapper.java.maxmemory` or `wrapper.java.maxmemory.percent` parameters (chapter 4.3.2.3).

A large Java Heap (more than 1 GByte) requires optimisation of the parameters to adjust the Garbage Collection, which is described in the following section.

Large quantities of data in FirstSpirit projects or a large number of simultaneously active FirstSpirit editors can overload the Java VM Garbage Collector in its standard configuration. The overload is noticeable through waiting times within the range larger than 10s in the response time of the FirstSpirit Client. If the waiting times are longer, connections between the FirstSpirit Client and the server can break. The cause of the waiting time is the Garbage Collection, which in the standard configuration temporarily completely stops the FirstSpirit Server for certain operations.

The message "Full GC" or „time exceeded“ can be seen at this time in the log file of the Garbage Collector (`log/fs-wrapper.log` or `log/fc-gc.log`).

A different method to the standard configuration for the Garbage Collector must be activated to prevent complete stoppage of the FirstSpirit Server by the Garbage Collector. Since Version 1.6.0 the Sun Java VM has provided the Garbage Collector which operates in concurrently to this: Concurrent Mark Sweep GC (CMS-GC). The IBM Java-VM offers a comparable Garbage Collector in Version 1.5.0 and 1.6.0 to prevent lengthy GC pauses (-Xgcpolicy:optavgpause).

From FirstSpirit version 4.2 the parameters of the following section are used already during the installation. After the installation only a customisation of the heap’s size is only for single parameters required which contain settings in absolute MByte. All parameters which contain only proportions or percentage values can be retained unchanged.

When updating from an older version than FirstSpirit 4.2 the parameters should be sollten replaced completely by the parameters mentioned here.

4.3.2.2 Common Java parameters

The following parameters in the file `conf/fs-wrapper.conf` are valid for Java VMs of all manufacturers when using the FirstSpirit server.

```
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 Configuration of the Sun Java VM

For using the FirstSpirit server in the Java VM of Sun the lines described in the following in the file `conf/fs-wrapper.conf` are required.

First, on 64bit systems the 64bit Java VM is activated:

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

If only the 32bit Java VM is to be used the parameter “-d64“ must be deleted and the following empty entry must be given to ensure a consecutive numbering of the entries “wrapper.java.additional“:

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

To configure the CMS-GC the following lines can be used.

```
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
```

The value given at “-Xmn” (M for MByte) defines the area of the Java Heaps which is used for temporary Java objects. If many temporary objects are used in FirstSpirit, it is advisable to set the value to 50% of the value given for `wrapper.java.initmemory`.

The value given at “-XX:MaxPermSize“ (M for MByte) defines the area of the Java Heaps which is used for Java classes and JSP pages. If the internal web server (Jetty) is used and FirstSpirit projects contain many JSP files or if many Beanshell scripts are used in FirstSpirit templates, this value should be increased. By means of the FirstSpirit Web-Monitor the utilised capacity of this heap area can be monitored under “Monitoring -> VM memory“ over a period of the last hour. Normally, FirstSpirit occupies 100 MByte quite constantly in this area.

At least in the test and evaluating phase the parameters for logging the Garbage Collector invocations should be activated:

```
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
```

Logging of the Garbage Collector, and above all the logging of the FirstSpirit server pauses caused by Garbage Collector overload, then takes place in the `log/fs-wrapper.log` file. This file is automatically archived when it reaches a size defined in conf/fs-wrapper.conf and after several archiving steps, which are also set in conf/fs-wrapper.conf, it is deleted.

The following parameter can be added for test systems, to activate clear logging of the Garbage Collector in a separate file (log/fs-gc.log). This configuration is not suitable for productive operation, as the log file continuously grows while the FirstSpirit server is running and is deleted again when the FirstSpirit server is restarted.

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

For further information on the individual parameters, please refer to http://java.sun.com/javase/technologies/hotspot/vmoptions.jsp.

4.3.2.2.2 Configuration the IBM Java-VM

To activate the Garbage Collector to minimise the waiting times of the Java application, the following lines in the conf/fs-wrapper.conf file are required.

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

Other possible parameters for Xgcpolicy:

optthruput: Is the default setting, but causes long Java application waiting times. The advantage is an otherwise high data throughput.

optavgpause: Concurrent GC with minimum waiting times. The disadvantage is a higher CPU capacity utilisation and lower data throughput.

gencon: Improved concurrent GC with additional separation of the Heap into several generations. Not available until Java 1.6.0.

The following parameters also activate logging of the Garbage Collector invocations in the log/fc-gc.log file, which is meaningful during test and development phase:

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

All other parameters in the area “wrapper.java.additional“, except for those mentioned in Chapter 4.3.2.2, must be deleted when using the Java VM of IBM.

A detailed description of configuration of the IBM-JDK's Garbage Collection is given in the "Java Diagnostics Guide" in the chapter "Reference, Command Line Options" on http://publib.boulder.ibm.com/infocenter/javasdk/v6r0/ and on the site http://www.ibm.com/developerworks/java/jdk/diagnosis/.

4.3.2.3 Java Wrapper Parameters

`wrapper.java.command:` Java Interpreter. Either only `java` (under Unix und Windows) if the environment variable PATH points to the correct JDK, or an absolute path to the Java Interpreter of the JDK, e.g. `/opt/jdk1.6.0/bin/java` or `c:\JDK1.6.0\bin\java.exe`.

`wrapper.java.maxmemory:` Maximum heap size for the Java VM in MByte. This is the memory share of the operating system which the FirstSpirit Server can maximally use. It should be chosen as large as possible, but not specified larger than the physical RAM. For 32-bit systems this value is limited to approx. 2 Gbytes.
If several FirstSpirit servers or other Java processes are run on one computer, the heap size of all Java processes is distributed accordingly so that the size of the available RAM is not exceeded.

> ! *If a heap is to be configured which is larger than 10 GByte, please ask the manufacturer before (e-Spirit AG), because in such a case further, special parameters for configuring the Garbage Collector are mostly required.*

`wrapper.java.maxmemory.percent:` This parameter has the same meaning as `wrapper.java.maxmemory`. However, here the maximum heap size is given as a percentage of the physical RAM. In 32 bit systems, the value does not refer to the size of the main memory, but to 2 GB. The default setting is 75 and does not usually have to be changed.
If several FirstSpirit servers or other Java processes are run on one computer, the heap size of all Java processes is distributed accordingly so that the size of the available RAM is not exceeded.

`wrapper.java.initmemory:` Heap size which the Java VM initially reserves. This value should be set to 75% of the value set for `wrapper.java.maxmemory`. The remaining 25% are available as buffer as reserve for situations with temporarily high workload. The Java VM tries to keep constantly the memory consumption to the value specified by `initmemory`. If an enduring storage allocation which is higher than this value during operation is noticed, an overload has occurred which requires a higher starting value.

`wrapper.java.initmemory.percent:` This parameter has the same meaning as `wrapper.java.initmemory`. However, here the heap size is given as a percentage of the physical RAM. In 32 bit systems, the value does not refer to the size of the main memory, but to 2 GB. The default setting is 75 and does not usually have to be changed.

> ! *The parameters* *`wrapper.java.maxmemory`* *and* *`wrapper.java.maxmemory.percent`* *can be alternatively used; however, not simultaneously.*
>
> *The same applies to the parameters* *`wrapper.java.initmemory`* *and* *`wrapper.java.initmemory.percent`.*

`wrapper.java.additional.X:` Parameters which are directly transferred to the Java VM. Java parameters for configuring the garbage collector are mainly entered here.

> ! *Only one Java parameter per line. All specified Java parameters have to contain consecutive, unique numbering (X).*

`wrapper.java.additional.X = *.jmxremote.*:` Configuration of the JMX Connector. JMX is used for requesting the system state and provides current system information of the FirstSpirit Server and Java system. `jconsole` from the JDK 1.6 or 1.5 or system monitors which support the JMX protocol can, e.g., be used as a client. For further configuration notes see:
http://java.sun.com/j2se/1.5.0/docs/guide/management/agent.html
The following entries activate the JMX Connector on port 9000 **without** authentication:

```
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:` Maximum processing times in seconds for certain system states of the FirstSpirit Server. If these time specifications are exceeded, the wrapper terminates the Java process because an undefined state is assumed. Parameter names for *: `startup`, shutdown, jvm_exit, cpu, ping.

`wrapper.timer_slow_threshold:` If the internal timer of the wrapper deviates from the system clock by the specified number of seconds, a warning is written to log/fs-wrapper.log. This parameter can be used to recognise a CPU overload, since the wrapper timer does not receive sufficient computing time to update and will, therefore, be slower.

`wrapper.umask:` Only Unix: All newly written files of the FirstSpirit Server receive the access attributes of the specified `umask`.

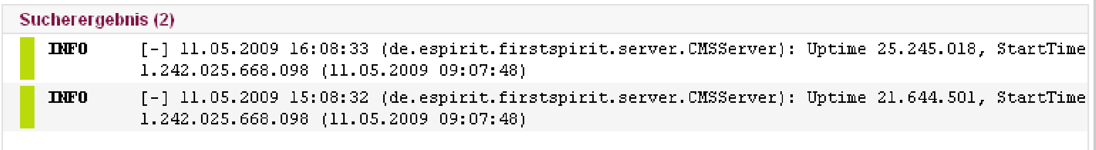
4.3.2.4 Logging

`wrapper.logfile.*:` Parameters for logging into file `log/fs-wrapper.log`. The maximum file size and the number of archive copies can be changed. Via parameter `loglevel` it is possible to switch from `INFO` for production to `DEBUG` for testing.

`wrapper.console.*:` Parameters for logging onto the current default version of the console. Logging onto the console is only active if the FirstSpirit Server has been started via “fs4 console” under Unix or via `Start menu → FIRSTspirit4 → Server Control→ Start server in console` under Windows.

`wrapper.syslog.*:` Only Unix: Configuration for logging into the system-wide syslogd system. If logging of the FirstSpirit Server has been set to stdout via `fs-logging.conf` (see section 4.3.6), this log is also sent to syslogd instead of `fs-server.log`.

The logging outputs of the FirstSpirit Server have been enhanced with **FirstSpirit Version 4.2**. The FirstSpirit Server logs one time when starting and cyclically each hour the VM StartTime and the VM Uptime:



Sucherergebnis (2)

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

Figure 4-1: FirstSpirit Server Monitoring: Log VM StartTime and VM Uptime

4.3.2.5 System service under Windows

! *Changes in the following area will only become effective after re-registering the FirstSpirit system service! The system service can be re-registered via `Startmenü` → `FIRSTspirit 4.0 or 4.1` → `Installation -> Deregister as service/ Register as service`. The FirstSpirit Server is stopped and started for this task.*

`wrapper.ntservice.name:` Object name of the FirstSpirit system service.

`wrapper.ntservice.displayname:` Displayed name of the system service.

`wrapper.ntservice.description:` Descriptive text for the system service.

`wrapper.ntservice.dependency.X:` System services required for FirstSpirit which should be located in front of FirstSpirit in the start sequence, e.g. MySQL. Use a new line with consecutive numbering (X) for each new service.

`wrapper.ntservice.starttype:` Start method, either `AUTO_START` for an automatic start during system start or `DEMAND_START` for a manual start.

`wrapper.ntservice.interactive:` Interaction with the desktop is not necessary for the FirstSpirit Server and should always be set to `false`.

4.3.2.6 Other parameters

The following parameters should not be changed, since FirstSpirit relies on the preset parameter values to function correctly:

```
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 Database connection configuration (fs-database.conf)

The file `fs-database.conf` is located in the FirstSpirit Server subdirectory `conf` and contains important configuration settings for connecting a database to the FirstSpirit Server and must be adapted, if necessary.

Changes to the configuration file `fs-database.conf` can be carried out via the FirstSpirit Server and Project Configuration (see section 7.3.6 page 220). The changes are subsequently written into the configuration file and updated on the server. If file system access is available, `fs-database.conf` can also be changed directly via the configuration file.

> *If the configuration file `fs-database.conf` is changed directly via the file system, the file is not automatically updated on the server. Therefore, changes should always be carried out via the Server and Project Configuration.*

```
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
```

See section 4.8 page 124 for further information on configuring the database connection. The parameters are described in the sections

- 4.8.4.1: obligatory parameters
- 4.8.4.2: optional parameters
- 4.8.4.3: oracle-specific parameters and
- 4.8.4.4: MS-SQL specific parameters

from page 138 ff.

4.3.4 Login process configuration (fs-jaas.conf)

The file `fs-jaas.conf` is located in the FirstSpirit Server subdirectory `conf` and contains configuration settings for the login process at the FirstSpirit Server.

The configuration file `fs-jaas.conf` can be changed via the FirstSpirit Server and Project Configuration (see section 7.3.13 page 231) or via Server Monitoring (see section 8.6.1.8 page 401). The changes are subsequently written into the configuration file and updated on the server. If access to the file system is available, `fs-jaas.conf` can also be changed directly via the configuration file. Comments commence with //.

The file created during installation with default values can be found in section 12.3.

> ! *If the configuration file `fs-jaas.conf` is changed via the file system, the file is automatically updated on the server (default: every 60 sec.). The server does not have to be restarted. The JMX Console offers another update possibility. The operation `reloadConfiguration`, which also updates the server, can be executed via the "AuthenticationManager" (see section 9.3 page 420).*

> ! *In the exceptional case in which LDAP, Kerberos or NTLM are **not** used for authentication, only ASCII characters in passwords are checked on signing on.*

FirstSpirit uses the Java standard JAAS[6] for user authentication. The following JAAS modules are already integrated in FirstSpirit and provide various user authentication methods:

4.3.4.1 Password check against the FirstSpirit user database

JAAS module name: de.espirit.firstspirit.server.authentication.**FSUserLoginModule**

The internal FirstSpirit user database is used.

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

4.3.4.2 LDAP

JAAS module name: de.espirit.firstspirit.server.authentication.**LdapLoginModule**

The LdapLoginModule provides 2 functions:

1. Authentication: The combination of user name and password entered on the FirstSpirit start page are checked against the given LDAP directory. For this application case, the LdapLoginModule will be entered in the `fs-jaas.conf` file in `webplain`.
2. Authorisation: Following authentication via any JAAS module, the information regarding group membership of the logged in user will now be read out of the LDAP directory. If the user authenticates themselves with a password, this 2nd function will be automatically performed during authentication and additional configuration is not necessary. If authentication takes place using a password-free ticket method, the LdapLoginModule must be entered in the `fs-jaas.conf` file in `websso` in the order behind the authentication module used.

An external LDAP server is used, e.g. the LDAP component of an Active Directory server. Reference to an LDAP section defined in `fs-server.conf` occurs via the parameter `section`, see section 4.3.1.11. Only 1 LDAP section may be transferred as parameter at a time. If more than one LDAP section is used, for each section an individual line must be entered into the file `fs-jaas.conf`.

4.3.4.3 Ticket from FirstSpirit user database

JAAS module name:de.espirit.firstspirit.server.authentication.**FSTicketLoginModule**

A ticket which has been generated by the FirstSpirit Server is sufficient for authentication. The ticket is generated during login at the FirstSpirit start page and forwarded via the web browser. This method is the default method after installation.

4.3.4.4 Ticket from the Windows-NETBIOS-domain (NTLM)

The NTLMv2 method is used as a default for authentication in the operating systems Windows Vista, Windows 7 and Windows Server 2008 R2.

The NTLM authentication is used by FirstSpirit Server if the NTLM login module is used for the login process. The NTLM login module is **not** compatible with NTLMv2. When using the aforementioned operating system versions and the NTLM login module, the setting of the LAN manager authentication level must be changed and NTLM(v1) allowed.

JAAS module name: de.espirit.firstspirit.server.authentication.**NTLMLoginModule**

A ticket created during login in a Windows domain is accepted. Editors only have to login once at their workstation, since the web browser automatically transfers the ticket to FirstSpirit. Only the Internet Explorer 6 and 7 are currently supported as web browsers for this login method. The Windows domains permitted for login are specified via parameter `domains`. Domain servers can be additionally specified as an option.

Entries for the parameter "domains" are possible as follows: `"Browser-Domain:Domain-Controller1,Domain-Controller2"`.

It is possible to enter multiple domains which are consecutively checked for login. `;` is used as separator.
Example:
`"Browser-Domain1:dc1,dc2;Browser-Domain2:dc3,dc4"`

Using the `userAgents` parameter: Here it is possible to enter a search pattern to activate NTLM login for selected web browsers only, as NTLM uses an HTTP header which does not fully conform to the standard ("WWW-Authenticate: Negotiate"); several older web browsers interpret this as an error. To use NTLM for all web browsers, enter `".*"`.
Default value: `".*MSIE.*"`

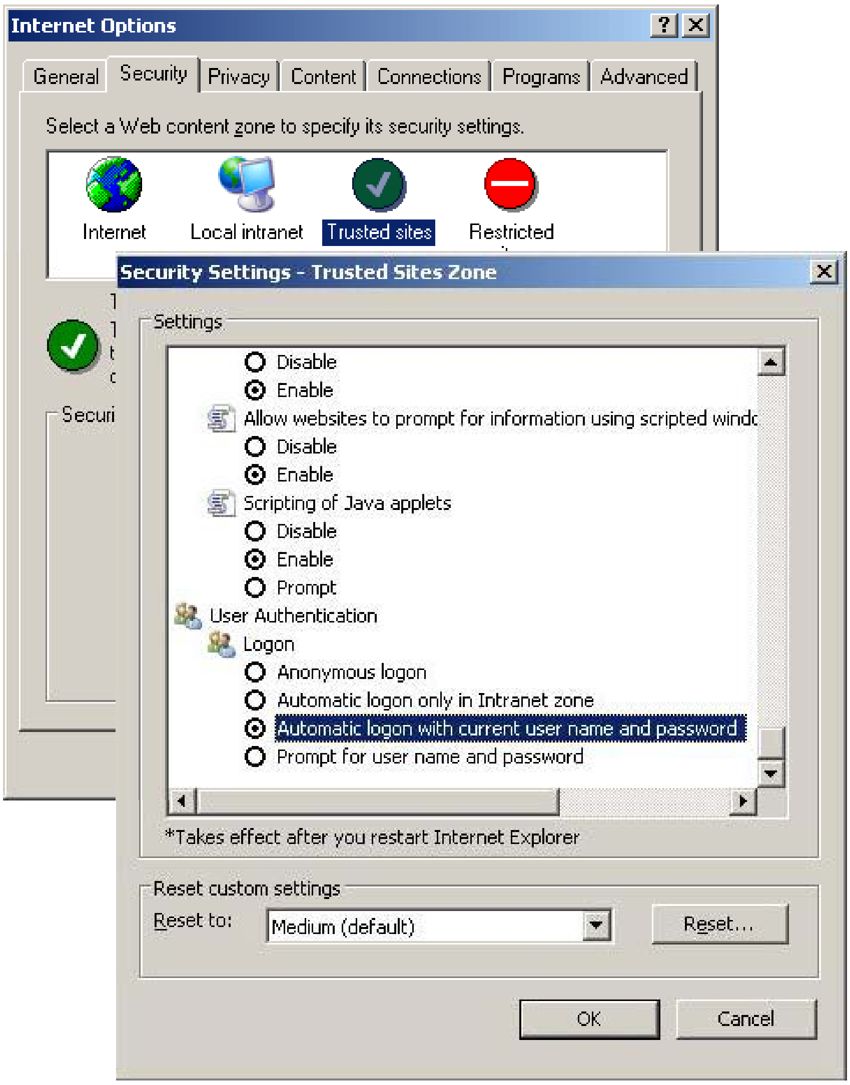
The module supports NETBIOS and Active Directory domains. "Browser-Domain" is the domain transferred by the web browser to the FirstSpirit Server in the login credentials. During login a search is carried out for an entry which matches the browser domain. The login credentials is subsequently sent to the specified domain controller for checks.

If a domain has not been entered, the login credentials is always checked at the entered domain controller(s) irrespective of the domain transferred by the browser; example ":Domaincontroller1,Domaincontroller2".

- If the Firefox is used, the following configurations are recommended, since Firefox does not transfer the domain of the user account to the server: ":Domain" or ": Domaincontroller1,Domaincontroller2".
- If the Internet Explorer is used, the following configurations are recommended: ":Domaincontroller1,Domaincontroller2" or "Browser-Domain:Domaincontroller1,Domaincontroller2”.

- If both browsers are to be used, it is possible to combine the configurations by separating them with `;`.

If login with ticket does not work when using the Internet Explorer, set the security settings as shown in Figure 4-2. User authentication should be set to "Automatic login with current user name and password". Additionally add the host name of the FirstSpirit Server at "Trusted sites".



○ Disable
● Enable
Allow websites to prompt for information using scripted windc
○ Disable
● Enable
Scripting of Java applets
○ Disable
● Enable
○ Prompt
User Authentication
Logon
○ Anonymous logon
○ Automatic logon only in Intranet zone
● Automatic logon with current user name and password
○ Prompt for user name and password

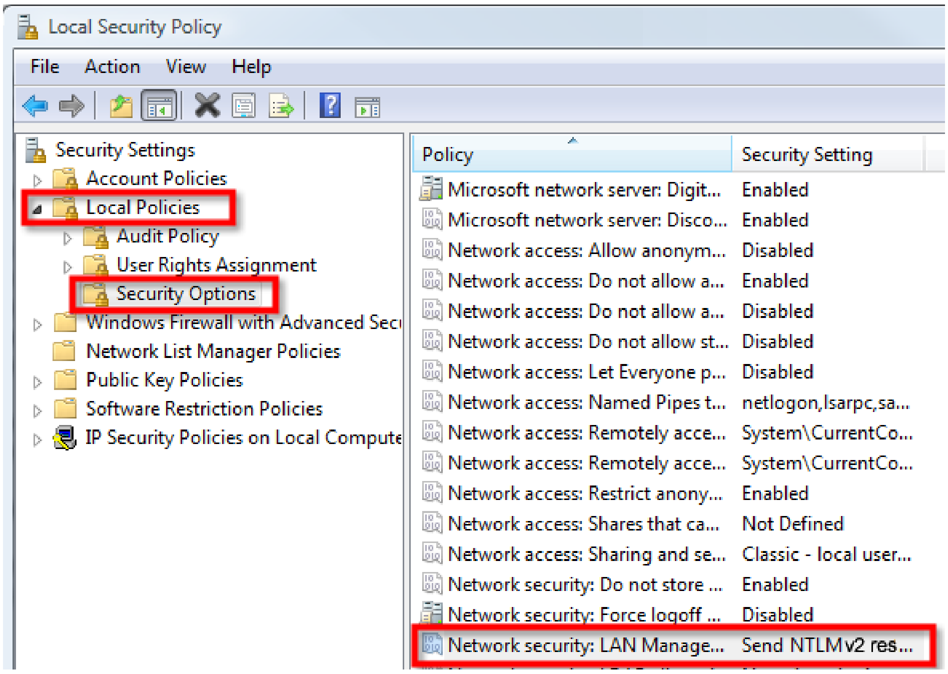
*Takes effect after you restart Internet Explorer

Figure 4-2: Adapt security settings

Settings in the operating system for adapting the behaviour to NTLM(v1)

The following instruction explains how to change over the operating system to the previous behaviour:

1. Press <Windows key> + <R>
2. enter `secpol.msc` and press <Enter>
3. Switch to "Local Policies" / "Security Options ":

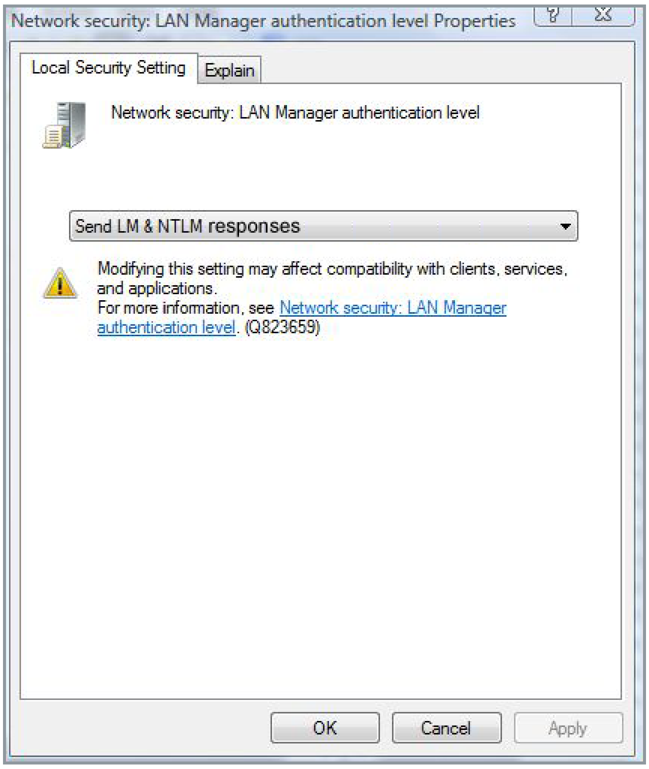


- Security Settings
  - Account Policies
  - Local Policies
    - Audit Policy
    - User Rights Assignment
    - Security Options
  - Windows Firewall with Advanced Secu
  - Network List Manager Policies
  - Public Key Policies
  - Software Restriction Policies
  - IP Security Policies on Local Compute

Policy	Security Setting
Microsoft network server: Digit...	Enabled
Microsoft network server: Disco...	Enabled
Network access: Allow anonym...	Disabled
Network access: Do not allow a...	Enabled
Network access: Do not allow a...	Disabled
Network access: Do not allow st...	Disabled
Network access: Let Everyone p...	Disabled
Network access: Named Pipes t...	netlogon,lsarpc,sa...
Network access: Remotely acce...	System\CurrentCo...
Network access: Remotely acce...	System\CurrentCo...
Network access: Restrict anony...	Enabled
Network access: Shares that ca...	Not Defined
Network access: Sharing and se...	Classic - local user...
Network security: Do not store ...	Enabled
Network security: Force logoff ...	Disabled
Network security: LAN Manage...	Send NTLMv2 res...

Figure 4-3: Network security: LAN manager authentication level

4. A window opens when the "Network security: LAN Manager authentication level" entry is double clicked.
5. NTLM must be allowed as a value for the LAN authentication in this window (the following screenshot shows the default setting in Windows XP):



Network security: LAN Manager authentication level Properties

Network security: LAN Manager authentication level

Send LM & NTLM responses

Modifying this setting may affect compatibility with clients, services, and applications.
For more information, see Network security: LAN Manager authentication level. (Q823659)

Figure 4-4: Default setting in Windows XP

6. The selection must be confirmed with the "OK" button.

> *From FirstSpirit Version 4.2, the Kerberos login module is also available, in addition to the NTLM login module. To use Kerberos, unlike NTLM, it is not necessary to make any changes to the settings in the operating system and it is the preferred option.*

4.3.4.5 Kerberos ticket (integrated Windows login) (from V4.2R2)

JAAS module name:
de.espirit.firstspirit.server.authentication.**KerberosLoginModule**

To log in, a Kerberos ticket is accepted, which is transferred from the web browser to FirstSpirit-Server. If using this login module, the FirstSpirit editor only needs to log in to their workstation (Windows, Mac, GNU/Linux) once in the morning, provided that a Kerberos infrastructure exists, and can then use FirstSpirit without renewed password input. Under Microsoft Windows, Kerberos is already known since Windows 2003 and XP as an "Integrated Windows login" and therefore replaces NTLM.

Parameter:

`useFullPrincipal`: Defines whether the full Kerberos login name including @ characters and realm (value "`true`") or without @ and realm (value "`false`") is used as the FirstSpirit user name. "`false`" is sufficient for systems whose user accounts are all entered in a Kerberos realm (corresponds to an Active Directory Domain under Windows). If logins take place from several Kerberos realms or Active Directory Domains, "`true`" must be given, as in most cases the pure user name is not unique over several domains.
Default value: "`false`"

`userAgents`: Here it is possible to enter a search pattern to activate Kerberos login for selected web browsers only, as Kerberos uses an HTTP header which does not fully conform to the standard ("WWW-Authenticate: Negotiate"); several older web browsers interpret this as an error. To use Kerberos for all web browsers, enter "`.*`".
Default value:
`".*(Firefox|Iceweasel|Konqueror|MSIE|Opera|Safari|Shiretoko).*"`

The KerberosLoginModule is entered in the `websso` area of the `fs-jaas.conf` file. In addition, the following new area must be added in the same file, at the end of the file:

```
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";
};
```

The paths and domain names must be adjusted according to the local system. The following parameters must be adjusted:

`principal`: The Service-Principal name of the FirstSpirit-Servers is given here.

`keyTab`: The path to the Kerberos-Keytab file is given here, which contains the private key, mostly in different encryption methods (e.g. RC4, DES and AES), matching the Service-Principal name. This file must be created first, as described in the following section.

Notes on the service principal name (SPN):

The keyword "HTTP" applies to the use of HTTP and HTTPS.

The host name given in the SPN including the DNS domain must be the real host name of the server. There are two options if a virtual web server is used: If the virtual server is entered as a CNAME record in the DNS, the host name to which the CNAME refers is entered in the SPN. If the virtual web server is entered as an A record in the DNS, the host name entered in the A record is used in the SPN. In both cases the IP address to which the host name entered in the SPN points, must refer back to the host names in the SPN.

Creating the Kerberos-Keytab file in the example under Microsoft Active Directory:

To create the file on a Kerberos server in a Microsoft Active Directory Domain, the additional Windows Support Tools[7] to be installed are required; they are supplied by Microsoft on the installation media of the operating system or can be downloaded from http://microsoft.com.

A normal user account is first created on the Windows Domain Controller. The password must not expire and the user must not be able to change it. The password is irrelevant and is overwritten in the next step. It is sensible to give the "host name"-"service name" as the user name, for example, fs4host-HTTP. To increase security, the "Do not trust user for delegation" option can be activated. The "Use DES encryption for this account" option must not be activated, otherwise Kerberos will not work with RC4 encryption, and this is used, e.g. by Windows 2008 and Windows 7.

A private key to the service principle name is now created on the Windows Domain Controller with RC4 encryption, which is normally the standard method in mixed networks that use Windows XP, Vista, 7, 2003 or 2008 as well as other operating systems:

```
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
```

If the version of the ktpass used does not provide `+rndpass`, a manually entered random password can also be used here via `-pass PASSWORT`.

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

If other crypto-algorithms are to be used to increase security, and these are supported by Kerberos-Realm and the clients, other keytab files can be created, for example, for 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
```

Calling `ktpass –h` displays the available crypto-algorithms, although it should be noted that these are only used if they are supported by all other domain servers and the respective client.

If problems occur, the list of all Service-Principal names of the user account can be displayed as follows:

```
setspn -l fs4host-HTTP
```

If errors occur during the input, a Service-Principal name can be removed using the following call:

```
setspn -d HTTP/fs4host.mydomain fs4host-HTTP
```

The keytab file created with kpass is now copied onto the FirstSpirit server or the external application server, to the path given in `fs-jaas.conf` at `keyTab`.
Example: `/opt/firstspirit4/conf/fs4host-HTTP.keytab`.

The file can be checked with the following call under Unix:

```
kinit -V -k –t fs4host-HTTP.keytab HTTP/fs4host.domain.net@DOMAIN.NET
```

As a result, "`Authenticated to Kerberos v5`" should be displayed.

If several crypto-algorithms are used, the individually generated keytab files must first be merged. To do this, start the `ktutil` service program under Unix:

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

and make the following entries in order to save the keytab file under the path given in `fs-jaas.conf` at `keyTab`:

```
rkt krb5-fs4host-HTTP-rc4.keytab
rkt krb5-fs4host-HTTP-aes256.keytab
wkt /opt/firstspirit4/conf/fs4host-HTTP.keytab
q
```

If an external application server is used for the FirstSpirit-Server instead of the integrated Jetty, the following parameters must be transferred to the application server on starting, for example, for Tomcat via the environmental variable

"CATALINA_OPTS":

```
-Djava.security.auth.login.config=/opt/firstspirit4/conf/fs-jaas.conf
```

If the internal Jetty application server is used, the following parameters must be added in the file `/opt/firstspirit4/conf/fs-webapp.xml` section `org.mortbay.jetty.nio.SelectChannelConnector`, as the HTTP request header can reach a size larger than 4kbyte with Kerberos:

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

If the Kerberos server cannot be determined via DNS, i.e. there are no SRV records such as `_kerberos._udp.mydomain.net` or `_kerberos._tcp.mydomain.net` available, the server needs the `/etc/krb5.conf` or `c:\windows\krb5.ini` file:

```
[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
   }
```

Configuration of the Kerberos-based password-free login is now completed on the server side.

The Kerberos error messages are logged in the
`/opt/firstspirit4/log/fs-server.log` and
`/opt/firstspirit4/log/fs-wrapper.log`
files and, if Tomcat is used as the external application server, in
`tomcat/logs/firstspirit.log`.

As soon as the Kerberos-based login has been successfully tested by the workstations/PCs, the `debug="true"` parameter must be changed to `debug="false"` in the `/opt/firstspirit4/conf/fs-jaas.conf` file, in order to prevent an unnecessary number of log messages.

If the login does not work, look first in the log files of the FirstSpirit server (fs-wrapper.log and fs-server.log) and if using an external application server in tts log file (e.g. firstspirit.log and catalina.out). One frequent error is excessively large time differences between the individual computer clocks, which must run synchronously within a range of a few minutes if Kerberos is used.

Configuration of the Clients

Depending on the web browser used, the following configurations are necessary on the Client side (if password-free Kerberos-based login is already used at the workstations/PCs for other web servers within the company network, it is not necessary to make any configuration changes):

Internet Explorer (Windows):
Add the following entries in the internet options for "trustworthy site": `https://*.mydomain.net` or `http://*.mydomain.net`, if HTTP only is used. Then activate "Integrated Windows Authentication" under "Advanced" in the Security area of the internet options. The following configuration may be necessary as well: In the internet options, under security in the "Trustworthy Sites" zone, select "Adjust level" and in the user authentication area activate "Automatic login with current user name and password".

Firefox (Windows, Mac OS, GNU/Linux):
Enter `about:config` as the URL in the address line and enter the domain name of the FirstSpirit Server with leading dot in the parameter `network.negotiate-auth.trusted-uris`. Several domains can be separated by commas. Example: `.mydomain.net`

Safari (Mac OS):
Mac OS already offers full Kerberos-integration as a default, provided the user account used is a network-based user account and the workstation/PC is logged into the Active Directory Domain or Kerberos-realm. No configuration changes are necessary. In the case of local user accounts, the first time the FirstSpirit-Start page is accessed the user is asked for their own Kerberos-user name (`username@MYDOMAIN.NET`) Principal including password.

Konqueror (GNU/Linux):
No further configuration is necessary if Kerberos has been activated in the operating system of the workstation/PC, i.e. a Kerberos ticket is automatically requested via `/etc/pam.d/common-auth` on logging in and unblocking of the screen is required.

! ***Security notice:*** *For security reasons, the KerberosLoginModule should be used in productive systems for successful prevention of replay attacks only in conjunction with HTTPS!*

4.3.4.6 Ticket from the SAP server

JAAS module name: de.espirit.firstspirit.server.authentication.**SAPLoginModule**

A ticket created during login at an SAP server is accepted.

4.3.4.7 Ticket from Windows

JAAS module name:
de.espirit.firstspirit.server.authentication.**WindowsLoginModule**

The login with ticket is enabled for the FirstSpirit JavaClient unless it is started via Java Web Start. The NTLMLoginModule (see Chapter 4.3.4.4 page 73) is sufficient for the JavaClient via Java Web Start. This module can only be used for a FirstSpirit Server installed under Windows.

4.3.4.8 General notes about the JAAS configuration

A user account is automatically transferred into the FirstSpirit system after successful authentication for all login modules. The login name is used as a unique identifier; thus ensuring the allocation of user accounts to projects in project exports.

! *From FirstSpirit-Version 4.1, automatic creation of user accounts can be suppressed by adding the parameter `JAAS.autoCreateUser` to the `fs-server.conf` file and setting it to the value `false`:*

```
JAAS.autoCreateUser=false
```

If the parameter is not set, the default value is `true`. Thus, new user accounts are automatically created if `JAAS.autoCreateUser` is not set.

The login modules can be allocated to the FirstSpirit components JavaClient, WebClient, WEBmonitor and Access API. Symbolic names are at first chosen as an intermediate step for the allocation; these symbolic names are allocated to individual

FirstSpirit components at a later date. Enter one or more login modules under each individual symbolic name in file `fs-jaas.conf`. If several login modules are entered, they are processed in the specified sequence until the user has been successfully authenticated. Please note that authentication methods without password but with ticket are entered in front of those with password check. Additionally, each login module has to be allocated with the JAAS attribute `optional`. “Optional” means that at least one of the login modules should have executed successful authentication to permit user login at FirstSpirit. Other JAAS attributes, such as `sufficient`, `required` or `requisite`, should not be used for FirstSpirit, otherwise FirstSpirit-specific login attributes will not be transferred from one login module to the other. These FirstSpirit-specific login attributes are also the reason that external JAAS modules can only be used for FirstSpirit with an additional wrapper class.

The following symbolic names are used as default allocation: `plain`, `sso`, `webplain`, `websso`, `system`.

Allocation of the symbolic names to the individual FirstSpirit components occurs in file `fs-server.conf` via the parameters JAAS.*.

The default configuration as defined during installtion is shown below:

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

Allocation of the FirstSpirit components to the parameter names:

- JavaClient: `JAAS.client`
- Server and Project Configuration: `JAAS.admin`
- all FirstSpirit web applications (WebClient, start page, Server Monitoring) with SSO authentication: `JAAS.websso`
- all FirstSpirit web applications (WebClient, start page, Server Monitoring) without SOO authentication: `JAAS.webnonsso`
- Access API: `JAAS.system`

4.3.4.9 Configuration examples

Default configuration:

In connection with the default configuration of file `fs-jaas.conf` described in section 12.3 the following login method results for the JavaClient:

1. The user is prompted to enter user name and password when calling the FirstSpirit start page via the web browser. This data refers to the entries in the FirstSpirit user database managed via the Server and Project Configuration. After successful authentication the ticket is generated and transferred by the web browser at a later date.

2. When starting the JavaClient via Web Start, the web browser transfers the previously created ticket via the JavaClient to the FirstSpirit Server for checks. Further password entry is not required.

3. If the ticket has expired or could not be transferred to the FirstSpirit Server, the JavaClient alternatively prompts the user to enter the password.

Login at a Windows domain with use of LDAP:

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

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

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

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

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


//enable for KerberosLoginModule only:

//com.sun.security.jgss.accept {
//  com.sun.security.auth.module.Krb5LoginModule required
//    principal="HTTP/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";
};
```

Extract from the file `/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 Licence configuration (fs-license.conf)

File `fs-license.conf` is located in the FirstSpirit Server subdirectory `conf` and contains the FirstSpirit licence and should not be changed.

The licence parameters of `fs-license.conf` can be displayed via FirstSpirit Server Monitoring (see section 8.6.1.2 page 396). Furthermore, it is also possible to insert a new licence file via Server Monitoring. When inserting a new configuration file `fs-license.conf`, it is not necessary to restart the server. The file is automatically updated on the server.

Manipulations to `fs-license.conf` result in an invalid licence. If changes are necessary (e.g. IP address change), please contact the manufacturer. A `fs-license.conf` does not have to contain all the information described in the example.

```
#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:` Maximum number of projects which can be created with this licence on the server. The deactivated projects are also included.

`license.EXPDATE`: Expiration date of the licence. The FirstSpirit Server will terminate automatically on this date. If the corresponding parameters have been set in the configuration file `fs-server.conf` (see section 4.3.1.10 page 50), a reminder is sent via email prior to the expiration date.

`license.VERSION`: FirstSpirit software version for which this licence is valid.

`license.USER`: Name of the licencee.

`license.FEATURES`: Licence-dependent additional functions released via this licence.

`license.ARCHIVE`: Value "1" activates the archiving function, thus enabling the archiving of generated pages (see "FirstSpirit archiving function" for further documentation).

`license.WORKFLOW`: Value "1" activates the "Workflow" function, thus enabling the creation of workflows which can be run through in specified work steps (see the "FirstSpirit Manual for Developers - Basics" for further documentation).

`license.DOCUMENTGROUP`: Value "1" activates the "Document group" function, thus enabling the page references in the Site-Store to be summarised as a group and, therefore, the generation of a result document, e.g. a pdf file (see the "FirstSpirit Manual for Developers - Basics" for further documentation).

`license.MAXSESSIONS`: Maximum number of sessions which can be simultaneously opened on the server. Internal server sessions (preview, generation) are not included. If the maximum number of sessions is exceeded, it is possible to open a maximum of two further server administrator sessions (unless two server administrator sessions have already been opended) to terminate current sessions, if necessary.

`license.WEBEDIT`: Value "1" activates the "WebEdit" function, thus enabling the direct editing of editorial contents within the preview page in the browser window (see "FirstSpirit Manual for Template Developers" for further documentation).

`license.MAXUSER`: Maximum number of users which can be created with this licence on the server.

`license.IP`: Comma-separated list of the server IP addresses for which the licence is valid. If the FirstSpirit Server is started on a computer with a different IP address, the licence is invalid.

`license.MODULES`: Licence parameter for modules, several modules can be separated by a comma.

`license.SCOPE`: Differentiation between single and corporate licence (`license.SCOPE=SINGLE` or `license.SCOPE=CORPORATE`)

`license.TYPE`: Definition of the licence type (`PRODUCTION`: "Productive", `DEVELOPMENT`, `DEMO`: "Demonstration", `STAGING`: "Quality assurance", `TRAINING`). The licence type is displayed on the FirstSpirit start page and on the project entry page of the FirstSpirit JavaClient by using a correspondent logo. In the case of a "Productive" licence the project logo which can be selected in the Server and Project configuration for the respective project (see Chapter 7.4.2 page 257) is shown. In the case of the other licence types the licence logo is shown instead of the project logo.

`license.KEY`: Licence key

`license.OFFICE_INTEGRATION`: If the value is "1", Microsoft Office-, OpenOffice- or Google Docs text documents can be used in the FirstSpirit AppCenter.

`license.APPTAB_SLOTS`: Maximum number of AppCenter applications which can access the application API. With `license.APPTAB_SLOTS=5`, for example, five different applications can be used or URLs can be opened. It does not matter which applications these are. Because unlike licensing of a FirstSpirit (module) add-on, it is not the function that is licensed here, but the number of integrated applications. (See also *FirstSpirit Release Notes Version 4.2R4*, Chapter 6.5.2 "license.APPTAB_SLOTS" page 145.)

4.3.6 Logging configuration (fs-logging.conf)

The file `fs-logging.conf` is located in the FirstSpirit Server subdirectory `conf` and contains important configuration settings for the "log" outputs. It must be adapted, if necessary.

Changes to the configuration file `fs-logging.conf` can be carried out via FirstSpirit Server Monitoring (see section 8.6.1.3 page 397). The changes are subsequently written into the configuration file and updated on the server. If access to the file system is available, `fs-logging.conf` can also be directly changed via the configuration file.

Occurred errors and info messages are transferred to the logging system "log4j"[8]. The log outputs can be categorised via the framework. The configuration example shows, e.g., the categories `DEBUG`, `INFO` and `ERROR`. It is, however, also possible to configure additional categories (e.g. `FATAL`, `WARN`). The two stages `ALL` and `OFF`, which either deactivate the logging completely (`OFF`) or output all messages unfiltered (`ALL`), are an exception.

Further logging files are included in the FirstSpirit scope of installation. Activation of a certain logging configuration as well as filtering and output type can be configured via Server Monitoring during runtime (see section 8.6.1.3 page 397).

Configuration files should have the following syntax:
`fs-logging_myLogging.conf` to ensure they are recognised by FirstSpirit and activated via Server Monitoring.

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

For further information on the logging framework "log4j" and a parameter description see: http://logging.apache.org/log4j/docs/documentation.html

Details of paths under Windows: If a file is to be specified for output of the log files, the path must be given as follows (path details separated by /): `log4j.appender.file.File=D:/FIRSTspirit4/log/err.log`

Example:

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

[8] Further information http://logging.apache.org/log4j/docs/documentation.html

```
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
```

Specific parameters of the FSAppender:

Parameters for selecting the logging method: The configuration parameters described below are used for selecting the logging method and can be activated/deactivated independently of each other. Permitted values are 0 for "deactive" and 1 for "active".

`log4j.appender.fs.consoleLogging`: Toggle for log message output to the console (command line/Shell) from which FirstSpirit has been started.

`log4j.appender.fs.plainLogging`: Toggle for log message output as plain text in a file. The log file for the server always has the name `fs-server.log`.

`log4j.appender.fs.datedLogging`: Toggle for log message output as plain text in a file. The date of the first entry is always added to the log file name, e.g. `fs-server.20070322_100436.log` (with first entry on March 22, 2007 at 10:04:36 a.m.). Activate this parameter to search within the Server Monitoring log files.

Parameters for maximum file size configuration:

`log4j.appender.fs.maxFileSize`: This parameter influences the max. permitted log file size. The size is stated in Bytes. Default value is 5 MB.

Configuration of the log rotation: Depending on the logging method, a reset or rotation occurs if the max. permitted log file size has been achieved.

1. If the parameter `plainLogging` has been activated and the parameter `datedLogging` deactivated, the current log file (e.g. fs-server.log) is renamed. The date of the first entry is added (e.g. fs-server.20070322_100436.log). The renamed file is subsequently compressed and the additional suffix .gz is added to the name. Meanwhile, further logging takes place in a newly created log file with the original name.
2. If the parameter `plainLogging` has been deactivated and the parameter `datedLogging` activated, the current log file (e.g. fs-server.20070322 _100436.log) is compressed and the additional suffix .gz is added to the name. Meanwhile, further logging takes place in a newly created log file (with new date extension).

3. If the parameters `plainLogging` and `datedLogging` have both been activated, logging occurs in the log files with and without date extension. Rotation occurs similar to point 2 (the log file without date extension is simply reset).

Parameters for logging behaviour configuration:

`log4j.appender.fs.buffer` influences the buffer size (in bytes) (default value: 8192 bytes) which should be used internally. The buffer accepts log messages und stores them until the buffer size has been reached. Only then are the messages written into the log file to avoid unnecessary and time-intensive write operations. When terminating the server, it is still written into the buffer even if the buffer size has not been reached.

`log4j.appender.fs.flushCycle:` Determines the maximum time (in seconds) between two write operations. If this time has elapsed, it is still written into the buffer even if the buffer size has not been reached.

4.3.7 Web server configuration (fs-webapp.xml)

FirstSpirit provides an integrated web server including servlet engine for preview creation, Server and Project Configuration and working with WebEdit; this web server is automatically configured and activated during installation. Jetty[9] is used.

If necessary, the integrated web server can be partially or completely replaced by another web server and servlet engine combination for utilisation, e.g., PHP or ASP on Apache or IIS in FirstSpirit projects. See section 4.5 page 99.

Advantages of the integrated web server:

- Operating system independent since it is 100% Java.
- Simple configuration.
- Integration of project-specific web application configurations.

The file `fs-webapp.xml` is located in the FirstSpirit Server subdirectory `conf` and contains the configuration settings of the internal web server.

Changes to the configuration file `fs-webapp.xml` can be carried out via FirstSpirit Server Monitoring (see section 8.6.1.6 page 399). The changes are then written into

[9] Further information http://www.mortbay.org/

the configuration file. If access to the file system is available, `fs-webapp.xml` can also be changed directly via the configuration file.

The configuration file with the default values entered during installation can be found in section 12.4.

> ! *Changes in the configuration file will only become effective after restarting the web server via FirstSpirit Server Monitoring (see section 8.6.2.2 page 402).*

The default configuration of the configuration file consists of the following elements after installation:

- Connectors
- Web applications
- Logging

For further configuration possibilities see the Jetty documentation: http://docs.codehaus.org/display/JETTY/Jetty+Documentation

4.3.7.1 Connectors

Only the HTTP connector is activated as default setting:

```
<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>
```

General parameters of all connectors:

`port:` TCP port of the connector. On Unix systems it is only possible to specify a value greater than 1024 here, also see section 4.3.7.4.

`host` (optional): Bind address of the connector. Used to make the connector available on certain server IP addresses only. Specify an IP no. or host name here.

`MaxIdleTimeMs`: If the idle time of a client exceeds the specified time in ms, the connection is disconnected. However, a FirstSpirit user does not have to login again, since the session data is still valid in the web browser once the connection has been automatically re-established. See section 4.3.1.15 for configuration of the FirstSpirit session timeout.

Further possible connectors are AJP (section 4.5) and HTTPS (section 4.7).

4.3.7.2 Web applications

Only the internal FirstSpirit web applications are entered here; the configuration does not have to be changed. (User-defined project-specific web applications are entered in file `data/server/fs-webapp-project.xml`.)

```
<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 Logging

Activate logging of client accesses analogue to an access.log with Apache via the following entry:

```
<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 Use default port numbers under Unix

On Unix systems it is only possible to specify port numbers greater than 1024 for the connectors, since the FirstSpirit Server is not started as `root`. In order to use the default values 80 for http or 443 for https, it is therefore necessary to either configure a TCP port redirect or use an additional external web server (section 4.5).

A TCP port redirect, in this example from 80 to 8000, can be carried out under Linux via the following call of the local firewall configuration:

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

On other Unix systems it is also possible to use an internal firewall or alternatively `rinetd`[10] for redirecting.

4.4 Connection to an LDAP server

4.4.1 Authenticate via LDAP

In FirstSpirit the login/password authentication of a user can be executed via an LDAP[11] server. Users are allocated with a flag which shows whether they are LDAP users or not. Various LDAP configurations (so-called sections) can be created (see section 4.3.4 page 72) and configured in the FirstSpirit Server (see section 4.3.1.11 page 50). An LDAP user is associated with exactly one section (see section 7.2.4.2 page 205). Three authentication types are possible:

1. **LDAP Bind:** Name and password are sent to the LDAP server. To achieve this, the "Distinguished Name" (DN), i.e. the unique key for user authentication, has to be known in the LDAP server. If the DN exists, the transferred password is checked via the "Bind" operation. For an example about using LDAP Bind see Chapter 4.3.1.11 page 50.

2. **LDAP Search & Bind:** If the "Distinguished Name" (DN) of a user is unknown, it is possible to search in a subtree of the LDAP server. Define a search filter and a start node for this. Example:

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

   This filter searches for all entries in the LDAP tree in which the attribute "cn" matches the entered login name. Start node is the node with the DN "`dc=mycompany,dc=com`". If such a node is found, a "Bind" is executed. (see LDAP Bind).

3. **LDAP Search & Compare:** (This option works similar to point 2). However, no "Bind" operation is executed after a matching node has been found. The entered password is compared to any LDAP attribute instead.
   Example:

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

[10] rinetd: http://www.boutell.com/rinetd/

[11] LDAP (Lightweight Directory Access Protocol)

In this case, the entered password has to match the content of the "mail" attribute of the LDAP node.

If LDAP authentication is successful, if the user is not yet known on the FirstSpirit system, they are created on the FirstSpirit server as an external user (see Chapter 7.4.7.2 page 272). The login and password of the external LDAP user are automatically copied to FirstSpirit after logging in for the first time. For this reason, in earlier versions of FirstSpirit, following successful initial authentication, it was possible to authenticate yourself on the FirstSpirit server if the LDAP server was unavailable. However, this response resulted in password changes made in LDAP not being passed to FirstSpirit. From FirstSpirit Version 4.1.33, the password of an **external LDAP user** is therefore emptied following successful authentication in FirstSpirit. Logins with an empty password are rejected by the FirstSpirit server. An external LDAP user can therefore only log into the FirstSpirit server if the LDAP server is available during the login process. In the case of **internal LDAP users**, the password in FirstSpirit is retained. In this case, the user can log in with both the FirstSpirit password and with the LDAP password.

The **Server administrator** has a special role in the LDAP login (Login: Admin). This user is automatically created during installation of the FirstSpirit server. The server administrator's password is never emptied, regardless of whether they have been configured as an "external" or an "internal" user. It is therefore recommended that the server administrator's password (initially: "Admin") be changed directly after installing the FirstSpirit server.

The FirstSpirit w b applications in the internal Jetty web server then forward the HTTP queries for special file types over HTTP to the external web server.

4.4.2 Bind LDAP attributes to a FirstSpirit user

Besides pure authentication, it is possible to bind any LDAP attribute to the user attributes of a CMS user. To achieve this, set parameter `LDAP.IMPORT_USER` in configuration file `fs-server.conf` to `TRUE` (see section 4.3.1.11 page 50).

Additionally allocate an attribute in `fs-server.conf`:

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

All LDAP attributes defined in this manner are automatically imported during initial login of the respective user.

If several attributes from the LDAP server are to be mapped onto an attribute in

FirstSpirit, separate the individual attributes via comma (`,`). Via the parameter

```
LDAP.MULTI_VALUE_SEPARATOR=[separator]
```

the separator can be defined which is to be used to separate the read out attribute values. Use for example the configuration

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

to output first and last name, divided by a colon.

The following CMS user attributes can be overwritten by LDAP attributes during login (see section 4.3.1.11 page 50 for a configuration example):

- User name: Name of the FirstSpirit user.
- Email: Email address of the FirstSpirit user.
- Telephone: Telephone number of the FirstSpirit user.
- Initials: Initials of the FirstSpirit user.

4.4.3 Use TLS or SSL

If the FirstSpirit Server is to connect the LDAP server via TLS/SSL, the certificate of the LDAP server has to be imported into the FirstSpirit keystore first. The Java tool `keytool` from the "bin" directory of the JDK is used for this task. If the certificate has a different format, it can be converted into the keytool importable PEM format via the external service program `openssl`[12]. Example call for conversion:

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

If the certificate is, e.g., located in file "mycompany.crt" and has previously been moved to the FirstSpirit Server installation directory, it can be imported into the keystore as follows:

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

Additionally enter the path and password of the keystore as Java parameters in fs-wrapper.conf (section 4.3.2):

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

[12] http://www.openssl.org/

```
wrapper.java.additional.Y=-Djavax.net.ssl.trustStorePassword=changeit
```

A self-signed certificate which is created as follows after changing to the FirstSpirit Server installation directory can be used for test installations:

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

If the “first and last name” are requested, the fully qualified host name (host name incl. domain) has to be specified.

After a FirstSpirit Server restart, communication to the LDAP server can take place via TLS or SSL.

4.5 Integration into an external web server

Jetty is used as HTTP server and servlet engine in the FirstSpirit Server by default. If special, server-sided implementations (e.g. PHP or ASP) are to be used in FirstSpirit projects which cannot be evaluated by Jetty, it is necessary to additionally integrate an external web server. The external web server then redirects some HTTP requests to Jetty via the AJP connector.

This chapter describes use of the Apache HTTP server in combination with PHP. Other web servers can be integrated according to the same principle, provided they allow forwarding to the servlet engine over HTTP or AJP.

If web applications to the Java EE 5 standard are to be used in FirstSpirit projects, the Jetty web server must also be replaced by another servlet engine, for example, Tomcat 6, as described in Chapter 4.5.2. Version 6.1 of Jetty, used by FirstSpirit, only offers J2EE 1.4 with the add-on JDK 1.6.

If AJP is to be used as the protocol between an external web server and the servlet engine, another servlet engine must also be used, as the Jetty web server is not fully compatible with AJP 1.3. See Chapter 4.5.2.

Another configuration possibility is load balancing to several servlet engines (see section 4.5.4 page 113).

4.5.1 Apache HTTP server with Jetty servlet engine

Version 2.2 of the Apache HTTP server is used in this configuration in conjunction with the servlet engine of the Jetty web server integrated in FirstSpirit. As the Jetty web server is not fully compatible with AJP 1.3, in this case, HTTP must be used as the protocol between Apache and Jetty.

mod_proxy_http[13] is used to establish the HTTP connection with the servlet engine.

The configuration environment of the Apache depends on the operating system and in most cases is spread over several configuration files. A common convention is to use the /etc/apache2/httpd.conf file for general parameters, to use the /etc/apache2/mods-available directory for module configuration and to use a file under /etc/apache2/sites-available. for each virtual web server.

A separate virtual web server should be used for FirstSpirit, which is configured by the following entries so that HTTP queries for the FirstSpirit web applications are forwarded over mod_proxy_http to Jetty. Then, if necessary, an additional internal HTTP connection is then automatically established from the FirstSpirit web applications installed under Jetty to deliver special file types such as PHP or ASP.

The FirstSpirit-Jetty-configuration should be checked before Apache is configured. The parameters WEBAPP_ROOT_PATH und WEBAPP_ROOT_URL may not be entered in the firstspirit4/conf/fs-server.conf file and the following parameter must be defined:

```
INTERNAL_SERVLET_ENGINE=1
```

It is followed by the Apache configuration:

In the configuration example, the paths and addresses in the lines marked with "EXAMPLE" have to be adjusted to the local configuration. The "# EXAMPLE" entry must then be removed, otherwise syntax errors are displayed when the web server is started.

Following configuring and restarting the Apache server, the FirstSpirit homepage can be opened via Apache, for example, http://fs4.yourdomain.net.

PHP module configuration:

```
LoadModule php5_module modules/libphp5.so            # EXAMPLE

AddType application/x-httpd-php .php .phtml
AddType application/x-httpd-php-source .phps
```

mod_proxy_http modular configuration:

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

```
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>
```

Virtual web server:

```
<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 administation 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>
```

If, for tests or administration, the start page is to be displayed directly via Jetty by bypassing the Apache httpd, http://fs4server:8000 must be entered as the start page in the browser. In this case, port 8000 corresponds to the port of the Jetty server entered for `HTTP_PORT` in `fs-server.conf`.

The following changes are necessary to use https in the combination of Apache httpd and Jetty:

The following parameter must be added in the Apache configuration:

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

The file path given for `SSLCertificateFile` must point to a valid TLS/SSL certificate.

Change the line

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

to

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

In the `firstspirit4/conf/fs-webapp.xml` file, activate the "HTTPS-Connector" area and also enter 8443 there for the `port`. The self-signed certificate in the keystore `conf/fs-keystore.jks` supplied with the program and already entered in the configuration can be used for this configuration without any problems, because the Jetty HTTPS connector is only used internally between Apache httpd and Jetty.

4.5.2 Apache HTTP server with Tomcat servlet engine

Version 2.2 of the Apache HTTP server is used in this configuration in conjunction with the Tomcat 6 servlet engine. AJP is used as the protocol between Apache and Tomcat.

Unlike the Jetty servlet engine integrated in FirstSpirit, Tomcat 6 provides the Java EE 5 standard for web applications.

mod_jk and mod_proxy_ajp[14] exist as AJP connectors for Apache. This chapter describes the use of mod_proxy_ajp, as this module has been included in the scope of supply since Version 2.2 and therefore makes installation easier compared to mod_jk. Load distribution (i.e. load balancing) between several servlet engines can be achieved with both modules (see Chapter 4.5.4 page 113).

The configuration environment of the Apache depends on the operating system and in most cases is spread over several configuration files. A common convention is to use the /etc/apache2/httpd.conf file for general parameters, to use the /etc/apache2/mods-available directory for module configuration and to use a file under /etc/apache2/sites-available. for each virtual web server.

A separate virtual web server should be used for FirstSpirit, which is configured by the following entries so that HTTP queries for the FirstSpirit web applications are forwarded over mod_proxy_ajp to Tomcat. Then, if necessary, an additional internal HTTP connection is then automatically established from the FirstSpirit web applications installed under Tomcat to deliver special file types such as PHP or ASP.

The FirstSpirit configuration must be changed before Apache and Tomcat are configured. The following parameters must be defined in the firstspirit4/conf/fs-server.conf file to disable Jetty and to define the default web application used when the FirstSpirit homepage is opened via Apache, for example, http://fs4.yourdomain.net:

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

Then shut down the FirstSpirit server, remove the firstspirit4/web/fs4root directory and restart the FirstSpirit server.

The Apache is now configured:

In the configuration example, the paths and addresses in the lines marked with "EXAMPLE" have to be adjusted to the local configuration. The "# EXAMPLE" entry must then be removed, otherwise syntax errors are displayed when the web server is started.

PHP module configuration:

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

```
LoadModule php5_module modules/libphp5.so             # EXAMPLE

AddType application/x-httpd-php .php .phtml
AddType application/x-httpd-php-source .phps
```

mod_proxy_ajp module configuration:

```
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>
```

Virtual web server:

```
<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 administation 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>
```

The Tomcat configuration is described in the next chapter (4.5.3).

4.5.3 External servlet engine (Tomcat)

The Jetty 6.1 web server integrated in FirstSpirit 4 only offers J2EE 1.4 with add-on JDK and it is not fully compatible with AJP. If web applications are to be used in to Java EE 5 standard or AJP is to be used to link an HTTP server (e.g. Apache), it is advisable to use Tomcat 6[15], whose configuration for FirstSpirit is described in the following section.

Tomcat can either be run independently with the HTTP server integrated in Tomcat, or the Apache HTTP server can be added, as described in Chapter 4.5.2.

The FirstSpirit configuration must be changed before Tomcat is configured. The following parameters must be defined in the `firstspirit4/conf/fs-server.conf` file to disable Jetty and to define the default web application used when the FirstSpirit homepage is opened via Tomcat, for example, http://fs4.yourdomain.net:8080:

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

Then shut down the FirstSpirit server, remove the `firstspirit4/web/fs4root` directory and restart the FirstSpirit server.

In this example, Tomcat is installed in `/opt/firstspirit4/tomcat.` In this example, the Tomcat is installed in If it is to be installed in another directory, in the following configuration example, all occurrences of "${catalina.home}/../" must be replaced by "`/opt/firstspirit4/`".

In `tomcat/conf/server.xml`, change the `appBase` entry for`<Host>` to the `web` directory of the FirstSpirit installation:

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

In `tomcat/conf/server.xml` , set the coding for URI parameters to UTF-8, in addition add the parameter `URIEncoding` to the existing HTTP and AJP connectors:

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

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

[15] http://tomcat.apache.org. A bug (https://issues.apache.org/bugzilla/show_bug.cgi?id=50700) prevents that the context parameters in the file `context.xml` which are necessary for FirstSpirit are interpreted in the Tomcat versions 6.0.30 to 6.0.32. So Tomcat 6.0.29 or at least 6.0.33 or 7.0 should be used.

In the `tomcat/conf/web.xml` file, enable checking and compiling of the JSP files for each change without delay. To do this, replace the following existing lines in the file

```
<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>
```

by the following lines:

```
<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>
```

To activate the listing of the directories for the area `/fs4staging`, replace the lines

```
<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>
```

by the following in the same file:

```
<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>
```

To enable the Tomcat Manager for status monitoring, add a user account for the "manager" role of the Tomcat Manager in the `tomcat/conf/tomcat-users.xml` file:

```
<?xml version='1.0' encoding='utf-8'?>

<tomcat-users>

  <role rolename="manager"/>

  <user username="Admin" password="tomcat-password" roles="manager"/>

</tomcat-users>
```

Create the `tomcat/conf/Catalina/localhost/manager.xml` file with the following content to activate the Tomcat Manager:

```
<Context docBase="${catalina.home}/webapps/manager"

  privileged="true" antiResourceLocking="false"

  antiJARLocking="false">



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

</Context>
```

Replace the existing file `tomcat/conf/context.xml`[16] with the following content, to enable immediate checking for file changes (HTML and JSP) and to configure the TCP connection of the FirstSpirit web applications with the FirstSpirit server:

```
<?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`, extend the "`common.loader`" entry to include `fs-server.jar` and `shared/lib` of the FirstSpirit installation (**write it all without spaces in one line**):

[16] A bug (https://issues.apache.org/bugzilla/show_bug.cgi?id=50700) prevents that the context parameters in the file `context.xml` which are necessary for FirstSpirit are interpreted in the Tomcat versions 6.0.30 to 6.0.32. So Tomcat 6.0.29 or at least 6.0.33 or 7.0 should be used.

Up to and including FirstSpirit Version 4.2 Release 0:

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

From FirstSpirit Version 4.2 Release 2 this entry must be:

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

If, instead of `${catalina.home}`, another path is to be used under Windows, the following lower case notation of the drive letters must be used (**write it all without spaces in one line**):

Up to and including FirstSpirit Version 4.2 Release 0:

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

From FirstSpirit-Version 4.2 Release 2 this entry must be:

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

Create the `tomcat/lib/log4j.properties` file with the following content to divert the logging of the FirstSpirit web applications into a separate file:

```
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
```

From FirstSpirit Version 4.2 Release 2: Download the file `log4j-1.2.*.jar` from http://logging.apache.org/log4j/1.2/download.html and copy it to `tomcat/lib/`.

If using Tomcat instead of Jetty, "InternalJetty" is displayed as the web server's identifier in all locations in the server and project configuration (in the server properties cf. Chapter 7.3.15 and 7.3.16 from page 238, in the project properties Chapter 7.4.17 page 289) and in the `fs-server.conf` file, as from the view of the FirstSpirit server, the Tomcat server uses exactly the same files in the same place as the Jetty web server.

The configuration requirements of an application server described in Chapter 4.6.2 (page 119) apply to the Java-VM used by Tomcat. In Tomcat, the shell environment variable `CATALINA_OPTS` defined in the *.bashrc* file (or comparable) is used to configure the Java-VM. In addition, the environment variable `CATALINA_PID` should be defined, i.e. the path of a PID file should be given, e.g. `$HOME/.catalina.pid`, so that Tomcat can be shut down, defined by the following call, without a process "hanging":

```
catalina.sh stop –force
```

4.5.4 Load balancing to several servlet engines

This section describes the configuration for load balancing to two servlet engines which are installed on the hosts tomcat-host1 and tomcat-host2. The Apache HTTP server with `mod_proxy_balancer` in version 2.2 is used as the load balancer.

Other load balancers and servlet engines or application servers can also be used, whereby it is then necessary to ensure session affinity is enabled on the basis of the cookie entry for the session ID. This ensures that a user session is always processed by the same servlet engine, from the initial opening of the FirstSpirit homepage until logging off after finishing the work. Tomcat uses the entry "JSESSIONID" as the cookie entry for the session ID; this must be adjusted accordingly for other server combinations.

This configuration is only used for load balancing and not to increase the fault tolerance! Session failover between the servlet engines does not take place and is also not supported by FirstSpirit, as the FirstSpirit web applications are not able to persistently serialise their state.

The static files of the FirstSpirit web applications, which are automatically written by FirstSpirit in `/opt/firstspirit4/web` with each update, and the files generated by FirstSpirit for display in the web server are made available to the individual servlet engines via NFS (read only).

First, configure the HTTP server and the servlet engine as described in Chapters 4.5 to 4.5.3. Then enter further servlet engines in the Apache configuration as "BalancerMembers" and define the session affinity as cookie value JSESSIONID.

Addition to the Apache configuration of the virtual web server in the "<Proxy balancer" area:

```
<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 the Tomcat configuration file `tomcat/conf/server.xml`, change the line

```
<Engine name="Catalina" defaultHost="localhost">
```

to

```
<Engine name="Catalina" defaultHost="localhost" jvmRoute="nodeX">
```

and at the same time, replace "nodeX" with the respective route name of the servlet engine entered for BalancerMember.

In addition, replace all occurrences of ${catalina.home}/../, where reference is made to the FirstSpirit directories web and server/lib, **from FirstSpirit Version 4.2 Release 2** on data/fslib additionally, in conf/server.xml and conf/catalina.properties, with /import/firstspirit4/.

Make the /opt/FirstSpirit4 directory of the FirstSpirit server via NFS read-only available to all servlet engines as a local mount point /import/firstspirit4.

In this configuration, the preview cache directories must lie outside the read-only file system /import/firstspirit4, as they are written by the fs4preview and fs4webedit web applications. To this end, both preview cache directories are re-routed via Symlinks to the relevant local file system. To do this, open the following on the FirstSpirit server:

```
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
```

The local cache directories are created on each Tomcat instance:

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

Other directories below /opt/firstspirit4 are not required by the Tomcat instance as they are already available in the read access via NFS to /import/firstspirit4.

In addition, the host name or the IP address of the FirstSpirit server must be entered in the configuration file /opt/firstspirit4/conf/fs-server.conf, so that it can be used as the destination address for the TCP connection (FirstSpirit SOCKET) from the servlet engines to the FirstSpirit server:

```
HOST=my-local-FirstSpirit-hostname
```

Then restart the FirstSpirit server, start the servlet engines and restart Apache and if necessary check the availability of the servlet engines over http://fs4.yourdomain.net/balancer-manager/.

The FirstSpirit homepage can now be opened, for example, over http://fs4.yourdomain.net.

4.6 Integration into an external application server

If the application server can be configured with access to the FirstSpirit installation directory web, the configuration described in section 4.5.2 is recommended.

If it is not possible to access the file system of the FirstSpirit Server from the application server, the configuration described below can be used:

The FirstSpirit web applications are provided as a WAR file. The WAR file is automatically created with appropriate configuration by the FirstSpirit Server during each startup. The connection parameters socket host and socket port between FirstSpirit web applications and the FirstSpirit Server are read from file fs-server.conf and entered into the deployment descriptor of the WAR file for configuring FirstSpirit web applications.

Carry out the following steps to configure the application of an external application server:

1. The host name or the IP address of the FirstSpirit server must be entered in the configuration file /opt/firstspirit4/conf/fs-server.conf, so that it can be used as the destination address for the TCP connection (FirstSpirit SOCKET) from the servlet engines to the FirstSpirit server:

```
HOST=my-local-FirstSpirit-hostname
```

2. The Server and Project Configuration application can now be opened for integrating an external application server. The start page for invoking the application is provided by the web server integrated in FirstSpirit regardless of the external application server. This ensures that management of the FirstSpirit Servers continues to be possible if configuration problems occur with the external application server. Any web server control can be added and configured under “Server properties” (see section 7.3.15 page 238). At first, add the desired external web server (see 7.3.15.3 page 242). When carrying out the installation by means of WAR files like in this case, the input box “Webdirectory” remains empty, because FirstSpirit does not have any access to the file system of the application server.

3. Each FirstSpirit web application must be changed to the new

configured external web server, as described in section 7.3.16.5 page 247.

4. WAR files are available for installing the FirstSpirit web applications in order to integrate an external application server without access to the file system of the FirstSpirit Server. The WAR files can be automatically generated and downloaded via the Server and Project Configuration as described in section 7.3.16.5 (page 247 ff). Installation of the WAR files subsequently occurs manually via the respective web interface of the application server.

5. The FirstSpirit web applications (e.g. Server Monitoring) can now be installed on the web server (see section 7.3.16.5 page 247).

4.6.1 Integration into WebSphere Application Server

FirstSpirit has no further prerequisites regarding the application server to be used, beyond those set forth in the *Technical Datasheet*. In practice, however, it has become evident that especially the use of IBM WebSphere requires several special considerations.

In addition to the previously mentioned general configuration for integration into an external application server the following settings are necessary to be made when using IBM WebSphere as application server:

The file `firstspirit4/server/lib/fs-server.jar` (**from FirstSpirit Version 4.2 Release 2** `firstspirit4/data/fslib/fs-webrt.jar`) must be entered into the global class path of the WebSphere server, because the JARs which are digitally signed by e-Spirit can not be loaded in the web applications under WEB-INF/lib. The file `fs-server.jar` (**from FirstSpirit Version 4.2 Release 2** `fs-webrt.jar`) must not be integrated as “common library" but must be transferred to WebSphere as parameter of the Java-VM.

The parameters of the Java-VM can be found in the web interface for the WebSphere administration under “Servers > Application Servers > Your Server > Java and Process Management > Process Definition > Java Virtual Machine“. Here, enter the complete path to the file `fs-server.jar` (**from FirstSpirit Version 4.2 Release 2** `fs-webrt.jar`) to “Class path“, for example: `/opt/firstspirit4/server/lib/fs-server.jar` (**from FirstSpirit Version 4.2 Release 2** `/opt/firstspirit4/data/fslib/fs-webrt.jar`). If the FirstSpirit Server is operated on another server than websphere it is recommended to pass the directory `/opt/firstspirit4/server` (**from FirstSpirit Version 4.2**

Release 2 `opt/firstspirit4/data/fslib`) via NFS to WebSphere. In this way you need not to copy the file fs-server.jar (**from FirstSpirit Version 4.2 Release 2** `fs-webrt.jar`) manually to the WebSphere server in case of an update of the FirstSpirit Server.

All JAR files must be entered into the classpath of the Java-VM of the Websphere server, too, which are used by modules which are delivered by e-Spirit. The following proceeding is necessary after having activated a module for the areas preview (fs4preview), WebEdit (fs4webedit) or staging (fs4staging): First, open the WAR file/s which has/have been downloaded via the Server and Project properties by means of a ZIP tool or `jar xvf fs_<modulename>.war` and copy all JAR files with the naming pattern `fs-*.jar` from the directory `WEB-INF/lib` into the classpath directory of the websphere server where `fs-webrt.jar` has been stored already. In analogy to `fs-webrt.jar`, add the copied JAR-files via the Websphere admin interface to the classpath and restart Websphere. This proceeding is necessary after each update of FirstSpirit!

Moreover, type the following entries into the field “Generic JVM arguments“: `-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 see Chapter 4.3.1.9 Seite 48).

! *The Java VM of the Websphere application server must be configured in that way that it will delete Java classes for JSP files which are no more existing automatically and regularly. This can be achieved in the case of using the Java VM of Sun by using the parameter*
`-XX:+CMSClassUnloadingEnabled`
in combination with
`-XX:+UseConcMarkSweepGC`

Re-start subsequently the WebSphere server to activate these modifications in WebSphere!

4.6.1.1 Logging configuration fir FirstSpirit web applications under Websphere

First, add the following parameters in the Websphere interface under "Server > Application Server > Server Name > Process Definition > Java Virtual Machine > User-Defined Features", to display errors in the log configuration:

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

This setting should only be active temporarily for the next start of the web applications during these configuration steps, as otherwise, too much data is written in `SystemOut.log`. After logging has been successfully enabled, this parameter should be removed again.

The following configuration is permanently retained:

Download Version 1.1x of the `commons-logging.jar` file from http://commons.apache.org/downloads/download_logging.cgi, save as a local file available to the Websphere server and use the Websphere Admin interface to add under "Environment > Common Libraries" with the name "FirstSpirit Commons Logging".

Now, either add a reference to the common library "FirstSpirit Commons Logging" via a new class loader (Parent First) global under "Application Server > Server Name > Class loader > Classloader_ > Library Reference“ or add the reference later for each individual web application, after the web applications have been installed.

Add the following files with the respective given content on the FirstSpirit server below `firstspirit4/web/fs4preview`, `firstspirit4/web/fs4root` and `firstspirit4/web/fs4webmon`:

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

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

File: `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
```

On test systems the log level can be changed over from INFO to DEBUG.

Download Version 1.2.x of the `log4j-1.2.14.jar` file from http://logging.apache.org/log4j/1.2/download.html and copy into `WEB-INF/lib/`.

The logging takes place with these settings in the `SystemOut.log` file of the Websphere instance. This file is automatically rotated in the default settings of the

Websphere server when a specific limit is reached.

Log messages of the FirstSpirit web applications are labelled with the package name `de.espirit.*`.

The web applications are now configured and can be installed in the Websphere server in the next step. After installing the web applications, the Websphere server must be restarted to activate the logging!

> ! *Please read and follow the following chapter (4.6.2) for configuration of the Java VM of the Websphere server.*

4.6.2 External application servers requirements

The requirements of the FirstSpirit web applications firstly correspond to those of each web application; i.e. that continuous operation is ensured. Above all, this means pre-planned configuration of the Java VM with respect to the heap size and Garbage Collector. The same principles as those described in Chapter 4.3.2.1 "Configuration of Java VM" regarding the FirstSpirit server apply to configuration of the Java-VM. The following parameters can be used as the basis of the configuration, if the Sun Java VM is used:

```
-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
```

The parameter `–Xms` defines the initial heap size (corresponds to `wrapper.java.initmemory` in `fs-wrapper.conf`), the parameter `–Xmx` defines the maximum heap size (corresponds to `wrapper.java.maxmemory` in `fs-wrapper.conf`). A value of maximum 75% of the available main memory (RAM) is advisable. The chosen heap size should not be unnecessarily large, as FirstSpirit attempts to fill the whole free memory as a cache. The parameter `com.sun.management.jmxremote` is used to monitor the heap capacity utilisation, including during productive operation, so that memory allocation trends can be identified and if necessary the heap size can be enlarged. Current monitoring systems provide a JMX interface to enable monitoring of web application servers. It is advisable to use the "Lambda Probe" web application to query the current values (http://www.lambdaprobe.org). In addition, the `jconsole` program (http://java.sun.com/j2se/1.5.0/docs/guide/management/jconsole.html) can be used for interactive query. It is part of each Sun JDK (not JRE).

In test mode, the following additional parameters should be added to enable monitoring of the Garbage Collector's capacity utilisation. If this monitoring is not wanted during productive operation, a method for archiving the output must be inserted instead of the file name in `Xloggc`, so avoid a continuously growing file:

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

Especially for application servers, configuration of the `MaxPermSize` parameter is important, as a Java class is created in this memory area for each JSP file in use. The chosen area must therefore be sufficiently large. The current memory allocation can be queried using JMX.

4.7 HTTPS server configuration

The web server integrated in FirstSpirit can be configured for HTTPS in order to encrypt the transferred data of the web applications (WebClient, start page and Server Monitoring) via TLS/SSL.

Firstly, install a server certificate via the program `keytool` provided by the JDK and then activate the HTTPS listener of the web server.

4.7.1 Install a security certificate for a test server

Use the self-signed certificate of the provided keystore (`conf/fs-keystore.jks`) for test installations. Furthermore, the web server configuration can be directly changed (see section 4.7.2.2 page 122). Generate an independent test certificate with a different host name via the following call:

```
keytool -genkeypair -alias fs4.yourdomain.net -keyalg RSA -validity 1000 -keystore
conf/fs-keystore.jks -storepass changeit
```

If the “first and last name” (CN) are requested, the fully qualified host name (host name incl. domain) which is visible to the client has to be specified.

To delete a certificate with a specified alias name, in this example “jetty”, from the keystore:

```
keytool -delete -alias jetty -keystore conf/fs-keystore.jks -storepass changeit
```

To list all certificates:

```
keytool –list -v -keystore conf/fs-keystore.jks -storepass changeit
```

The web server configuration of the FirstSpirit Server is subsequently changed (section 4.7.2.2 page 122).

In order to enable use of the self-signed test certificate on sites of the FirstSpirit JavaClient, if it is not opened via Java Webstart or if the Webstart certificate cannot be handed over to the Java VM, the following parameters must be added on opening the JavaClient and the certificate file must be copied onto the client computer:

```
-Djavax.net.ssl.trustStore=pfad/zur/datei/fs-keystore.jks
-Djavax.net.ssl.trustStorePassword=changeit
```

4.7.2 Install a trusted security certificate

A security certificate has to be digitally signed by an official certification authority (CA), e.g. http://thawte.com, to be classified as trusted. There are two ways to create this type of security certificate in FirstSpirit. Either via `openssl`[17] or via the `keytool` provided by Java. Certificates created via openssl are advantageous, since any other web server, e.g. Apache, IIS, Tomcat, etc., can use them. A certificate created via keytool can only be used for Java-based web servers.

4.7.2.1 Create a security certificate via `keytool`

A private key is generated first. To achieve this, enter the following command after changing to the FirstSpirit Server installation directory:

```
keytool –genkey –keystore conf/fs-keystore.jks –storepass=mypass –
alias fs4.yourdomain.net –keyalg RSA –keysize 1024 –validity 3650
```

The key length and the validity in days are specified by the “keysize” and the “validity” respectively. If the key password is requested, specify the same as for “–storepass”. If the “first and last name” are requested, enter the fully qualified host name as visible to the client, e.g. fs4.yourdomain.net.

[17] http://www.openssl.org/

In the next step, a certification request has to be generated:

```
keytool –certreq –keystore conf/fs-keystore.jks –storepass
changeit –alias fs4.yourdomain.net –file request.csr
```

The file "request.csr" is subsequently sent to the certification authority. Import the received response certificate (public.pem) into the keystore:

```
keytool –import –trustcacerts –keystore conf/fs-keystore.jks –
storepass changeit –alias fs4.yourdomain.net –file public.pem
```

If the certification body issued certificates via a hierarchy (certificate chain), all certificates in the hierarchy must either already exist in the certificate store of the Java VM or must be imported into the certificate store, otherwise keytool issues the error message: "Failed to establish chain from reply". To import the certificates of the hierarchy, the command must be called for each file and, e.g. the file name always given for "-alias":

```
keytool –import –trustcacerts –keystore conf/fs-keystore.jks –
storepass changeit –alias chain_CA_1 certificate1 –file
chain_CA_1.pem
```

The certification authorities might sometimes send certificates which the keytool cannot understand. These certificates can be converted by OpenSSL, e.g.:

```
openssl x509 –in public.crt –out public.pem –outform PEM
```

4.7.2.2 Generate a security certificate via `openssl`

Generate the private key first:

```
openssl genrsa -out private.key 1024
```

A certification request is then created (`request.csr`) and subsequently transferred to the certification authority (CA) for signing:

```
openssl req -new -key private.key -out request.csr
```

The certification authority subsequently returns the public signed key (certificate), usually in PEM format, as a text file (`public.pem`) which commences with "-----BEGIN CERTIFICATE-----". The private and the signed public key have to be summarised in a keystore in PKCS12 format for the FirstSpirit web server. This is achieved by changing to the FirstSpirit Server installation directory and entering the following command to create the keystore. Select a password for the keystore. This password is immediately entered into the web server configuration. "changeit" has been selected in the example.

```
openssl pkcs12 -inkey private.key -in public.pem -export -out
conf/fs-keystore.p12 -caname root
```

4.7.2.3 Change the web server configuration of the FirstSpirit Server

Activate the HTTPS listener in configuration file fs-webapp.xml. The configuration parameters have been entered in the file, but are deactivated via “<!-- -->”. The following configuration parameters are necessary:

```
<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>
```

“Password” is the keystore password and “KeyPassword” is the certificate password (or private key password), which is usually the same as the keystore one.

If certificate generation via `openssl` has been selected, the following option has to be added

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

and the name of the keystore changed from `fs-keystore.jks` to `fs-keystore.p12` under keystore.

After restarting the FirstSpirit Server, the start page is also available via https://fs4.yourdomain.net:8443 next to http://fs4.yourdomain.net:8000.

For further information on the HTTPS configuration of the Jetty web server used by FirstSpirit see:

http://docs.codehaus.org/display/JETTY/How+to+configure+SSL

4.8 Database connection

FirstSpirit stores the highly structured contents of the Content-Store in a database to enable efficient, complex search requests within this data.

FirstSpirit provides a graphical user interface which enables users to create and modify structured database tables and to formulate requests. FirstSpirit implements a database abstraction layer which maps the universal FirstSpirit content type system onto the database system to be used.

The following targets are achieved via this architecture:

1. All databases for which a database abstraction layer has been implemented can be directly used as Content-Stores.
   (currently: MySQL, Oracle, PostgreSQL, DB2, MS-SQL-Server, Derby). See the current *FirstSpirit Technical Datasheet* for detailed information.
2. Support of new database systems can be easily realised by implementing the database abstraction layer.
3. A project export or import usually occurs from one database system to another.

The minimum requirement to realise a database abstraction layer is a JDBC driver Version 2.0, which can be executed on the operating system of the server.

4.8.1 Storing the JDBC driver files

Different options exist for integrating the JDBC driver files in the FirstSpirit server:

1. Via the `shared/lib` directory:
   The relevant drive files must be copied into the `firstspirit4/shared/lib` directory as JAR or ZIP files, so that they are in the CLASSPATH of the FirstSpirit server's Java VM. It is then necessary to restart the FirstSpirit server.
2. As a FirstSpirit module: **recommended from FirstSpirit Version 4.2R4**
   It is possible to integrate the JDBC drivers as a FirstSpirit module to enable simultaneous use of various versions of a JDBC driver in different FirstSpirit projects and to exchange the JDBC driver while the FirstSpirit server is running without restarting it. The JAR file of the JDBC driver, including an additional XML file for name definition is combined here in an FSM file (ZIP archive). This FirstSpirit module is then referred to in the layer configuration via the additional `module` parameter (see Chapter 4.8.4.2 page 140).

! *JDBC driver files can be used furthermore in the directory `.../shared/lib` even after FirstSpirit Version 4.2R2, but it is no more advised.*

! *If different database drivers exist in the directory `.../shared/lib` and as module, the drivers which are deposited in `.../shared/lib` are used preferentially.*

4.8.2 Creating a JDBC driver module

The following files are required to create a JDBC driver module:

- file(s) of the JDBC driver, e.g. JAR file, licences etc. (see Chapter 4.8.2.1 Page 125)
- file `module.xml` (see Chapter 4.8.2.2 Page 126)

optionally:

- file `web.xml` (see Chapter 4.8.2.2 Page 126)

These files must be integrated by means of an FSM file (ZIP archive) to a FirstSpirit module (see Chapter 4.8.2.3 Page 130).

4.8.2.1 Files of the driver

If you have not used a module for JDBC driver yet, the files of the JDBC driver which are required for creating the module can be found in the directory `.../shared/lib` of the FirstSpirit Server or in the respective directory of the servlet engine (e.g. external Tomcat web server).

For Oracle, the JDBC driver of the series 10.1 (ojdbc14_10.1.0.x.jar) should be used because problems can arise if the data type LONG is used with version 10.2 from 4000 characters on and UTF-8 coding. As an alternative, the compatibility mode for Oracle 9 LONG must be activated when using the driver 10.2, because LONG is deprecated since Oracle 9. For this purpose the parameter

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

must be added in the database configuration.

4.8.2.2 module.xml and web.xml

The file `module.xml` contains the definition of the driver module and must be composed according to the following example. The basic framework is always the same, some tags and parameters vary depending on the used database type and version.

The following example represents the design of a `module.xml` file for a **PostgreSQL 8.2** database:

```
<!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>
```

The module consists of two parts: one part defines the resources for the FirstSpirit Server, the so-called "server component" (within the `<public>` tag), the other part for the web applications, the so-called "web application component" (within the `<web-app>` tag). Thus, this JDBC driver can be used in the FirstSpirit Server **and** in web applications. If the driver is required only for the server, the definition within `<web-app>` can be omitted.

For the use of the FirstSpirit web applications a servlet engine is required which implements the servlet API in the Version 2.4.

`<name>`: This tag must be used for assigning a unique name for the components. Only the following characters are allowed:
capital and lower case letters (A-Z, a-z), figures (0-9) and the special characters ; , _ \ -
The name which is assigned for the server component will be also used in the overview of modules of the FirstSpirit Server (see Figure 4-7) and must be indicated in the database layer configuration, too (see Chapter 4.8.3.2 page 132). The name which is assigned to the web application component will be also used in the Project properties, area "Web components" (see Figure 4-11). For a clearer overview *_Server* and *_WebApp* have been attached in the example to the names of the server and the web application component. .

`<description>`: This tag can be used to specify a description for the component.

`<class>`: This tag must be used to specify the complete class name of the used JDBC driver.

`<resources>` / `<resource>`: Use these tags to indicate the path to the JAR file of the JDBC driver.

`scope`: The value *module* should be used for this parameter within the `<resources>` / `<resource>` tags. This ensures that the JAR file applies only to the JDBC driver module and not for the whole server.

`<webresources>` / `<resource>`: Use these tags to indicate the path to the JAR file of the JDBC driver within the web application component. The following parameters should be used in addition:

`name`: The following default names should be used for the databases supported by FirstSpirit for the respective JAR files:
postgresql (PostgreSQL)
oracle (Oracle)
mssql (Microsoft SQL Server)
mysql (MySQL)
db2 (IBM DB2)
derby (Apache Derby)

`version`: This parameter should be used to indicate the complete version of the driver, e.g. *8.2.508* for version 8.2 build 508.

`minVersion` / `maxVersion`: Use these parameters to indicate the minimal or maximal version that can be used with the driver. In our example this means, that drivers of the versions 8.2.1 until 8.2.999 can be used. If a second driver is provided by another module, e.g. build 509, this can be used also from 8.2.1 until 8.2.999. In this case only the higher driver version (i.e. 509) will be copied to or assumed by the web application.

`<configuration>`: Contains information about the layer class and about the class name of the used JDBC driver.

`<layerclass>`: Use this tag to indicate the class which implements the database layer for this special database system, e.g.

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

for PostgreSQL or

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

for Oracle.

(See also parameter `layerclass` in Chapter 4.8.4.1 page 138 and the chapters containing the database specific sample configurations in Chapter 4.8.7 page 148.)

`<DRIVER>`: Contains the complete class name of the used JDBC driver, e.g. `org.postgresql.Driver` for PostgreSQL. (See also parameter `DRIVER` in Chapter 4.8.4.1 page 138 and the chapters containing the database specific sample configurations in

Chapter 4.8.7 page 148.)

If the JDBC driver should be available in a web application, the file `web.xml` is required:

```
<?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"/>
```

The value of the parameter `id` should be the name of the JDBC module (server component).

If the integrated **Derby database** is to be used in the web applications of a Tomcat web server, you need also a `module.xml` file. An exemplary `module.xml` could look like this:

```
<!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>
```

The part for the server component is only then required if both the internal Jetty and an external Tomcat web server are used at the same time.

The file `web.xml` is required as well. Example:

```
<?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"/>
```

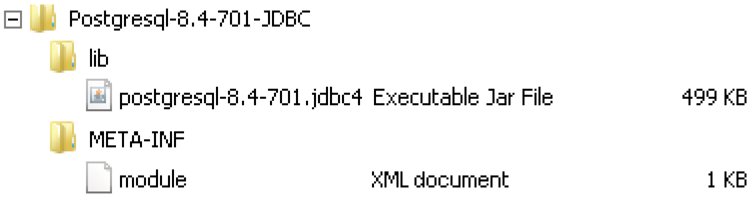
If the part for the server component is required, i.e. if internal Jetty and external Tomcat web server are used at the same time, the database layer must be adjusted (see Chapter 4.8.3.4.1 Seite 135).

The Derby database, integrated in FirstSpirit, is not dedicated for productive operation and should be used for test purposes only

An explication for the most tags used in these examples can be looked up also in the *FirstSpirit Manual for Developers (Components)* (German only).

4.8.2.3 Directory structure of a JDBC driver module

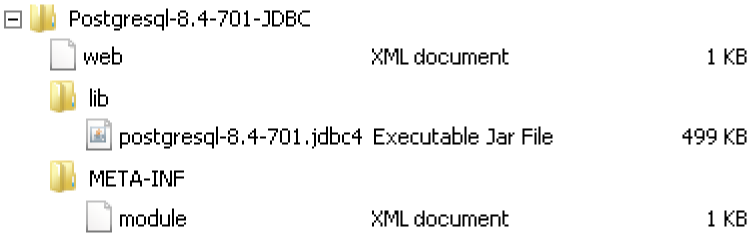
If the driver is only used for the FirstSpirit Server ("server component"), the files must be deposited in the following directory structure (cf. Chapter 4.8.2.1 page 125 and Chapter 4.8.2.2 page 126):



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

Figure 4-5: Directory structure server component

If it should be used also in web applications the file `web.xml` must be integrated on the highest level:



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

Figure 4-6: Directory structure server and web application component

To get a valid FirstSpirit module, a ZIP file must be created from the content of the superordinate folder ("Postgresql-8.4-701-JDBC"). The superordinate folder must not be included in the ZIP file. This ZIP file must then be renamed into `*.fsm`. If, in the example of Figure 4-6, the folder name was taken as file name, the module file should have the name `Postgresql-8.4-701-JDBC.fsm`.

As an alternative, the module file can be created using the following command:

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

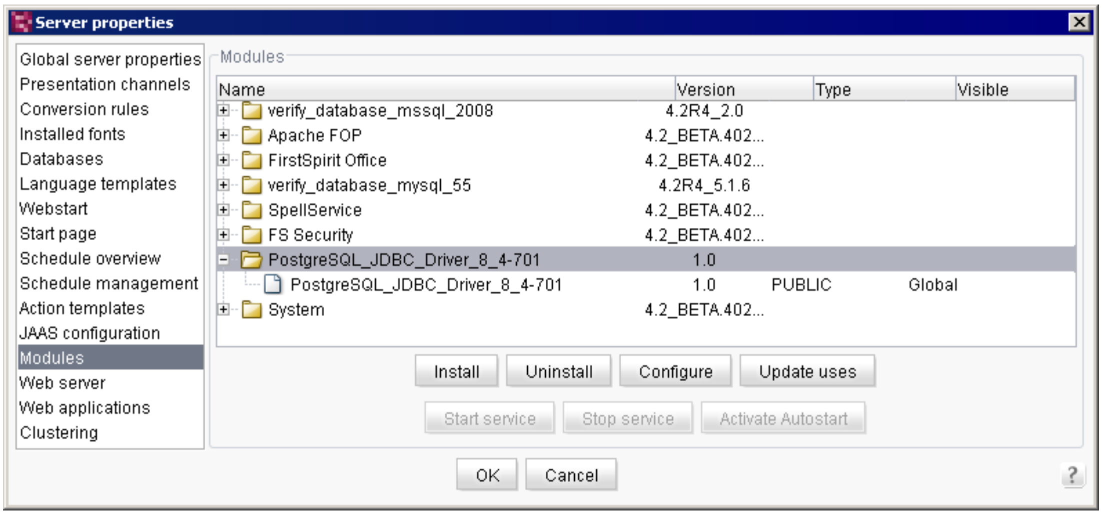
The programme "jar" is part of each JDK and can be found, depending on the operating system and installation, for example under `c:\programmes\jdk1.6.0\bin\jar.exe` or `/opt/jdk1.6.0/bin/jar`.

4.8.3 Installation and configuration of the JDBC driver module

4.8.3.1 Installation of the JDBC driver module

If the JDBC driver module has been created successfully as described in Chapter 4.8.2 page 125, it must be installed on the FirstSpirit Server. This will be carried out by means of the application for the Server and Project configuration.

For this purpose, the button "Install" must be clicked in the Server properties in the area "Modules". The driver module file can be selected from the locale file directory in the following dialog and uploaded to the server. The successfully installed file will then be displayed in the overview as module with its name (here: `PostgreSQL_JDBC_Driver_8_4-701`, see Chapter 4.8.2.2 page 126, tag `<name>`), version (tag `<version>`) and the included component (tag `<components>`):



Name	Version	Type	Visible
verify_database_mssql_2008	4.2R4_2.0		
Apache FOP	4.2_BETA.402...		
FirstSpirit Office	4.2_BETA.402...		
verify_database_mysql_55	4.2R4_5.1.6		
SpellService	4.2_BETA.402...		
FS Security	4.2_BETA.402...		
PostgreSQL_JDBC_Driver_8_4-701	1.0		
PostgreSQL_JDBC_Driver_8_4-701	1.0	PUBLIC	Global
System	4.2_BETA.402...		

Figure 4-7: Server properties – JDBC driver as module

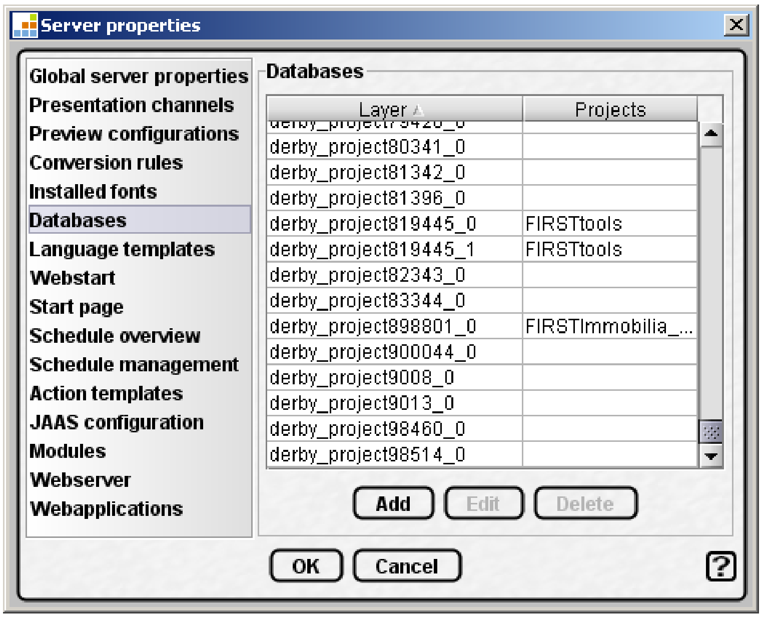
If the `module.xml` file contains the definition for a web application, this will be displayed here as well.

Here, no further configuration is required.

4.8.3.2 Configuration of the database layer

Subsequently, the parameter `module` must be used to refer to this driver module in the configuration of the layer of the database for which the driver module was created.

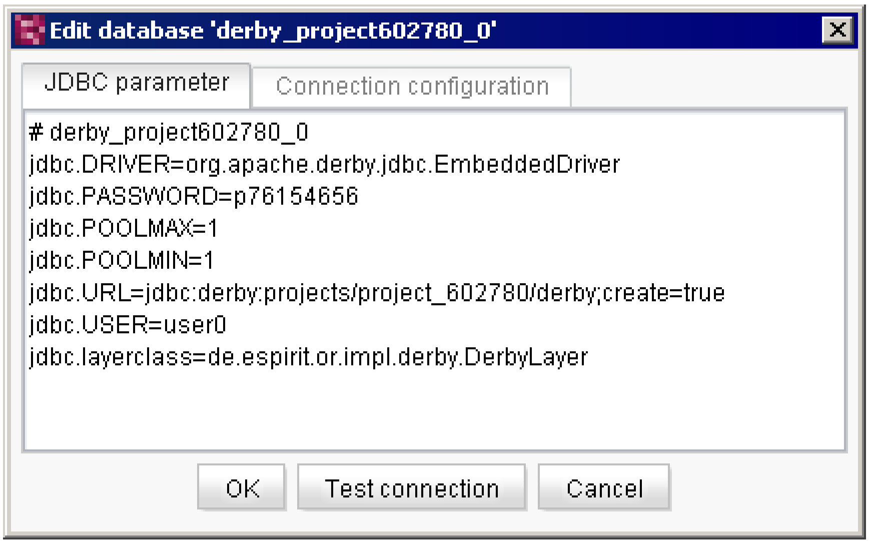
Select the respective database in the Server properties in the area "Databases":



Layer	Projects
derby_project79420_0	
derby_project80341_0	
derby_project81342_0	
derby_project81396_0	
derby_project819445_0	FIRSTtools
derby_project819445_1	FIRSTtools
derby_project82343_0	
derby_project83344_0	
derby_project898801_0	FIRSTImmobilia_...
derby_project900044_0	
derby_project9008_0	
derby_project9013_0	
derby_project98460_0	
derby_project98514_0	

Figure 4-8: Server properties – Databases

Double-click on the respective entry to open the dialog "Edit database". Here, you can edit the JDBC parameters for the database connection:

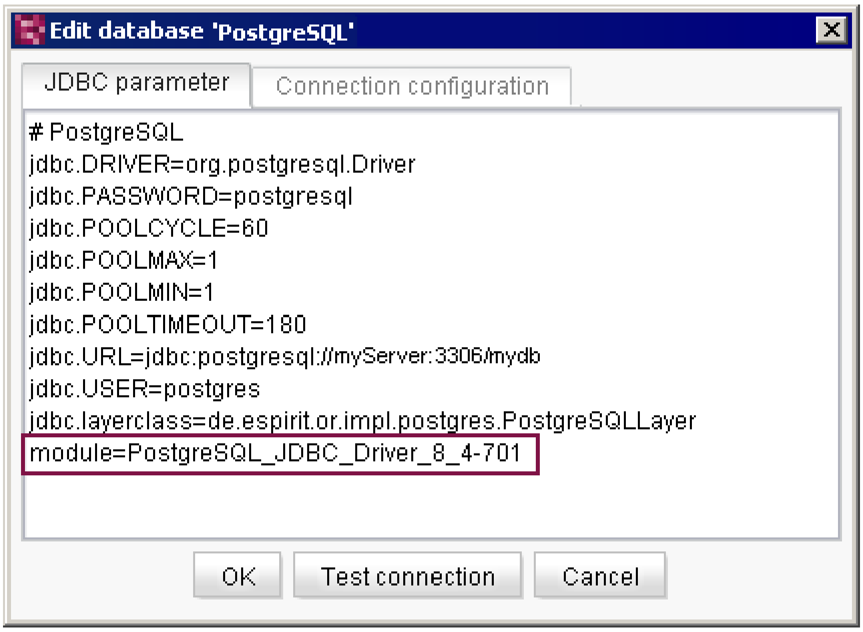


```
# derby_project602780_0
jdbc.DRIVER=org.apache.derby.jdbc.EmbeddedDriver
jdbc.PASSWORD=p76154656
jdbc.POOLMAX=1
jdbc.POOLMIN=1
jdbc.URL=jdbc:derby:projects/project_602780/derby;create=true
jdbc.USER=user0
jdbc.layerclass=de.espirit.or.impl.derby.DerbyLayer
```

Figure 4-9: Database configuration

Add the parameter `module` with the name which has been defined via `<name>` in the `module.xml` file, in our example:

```
module=PostgreSQL_JDBC_Driver_8_4-701
```



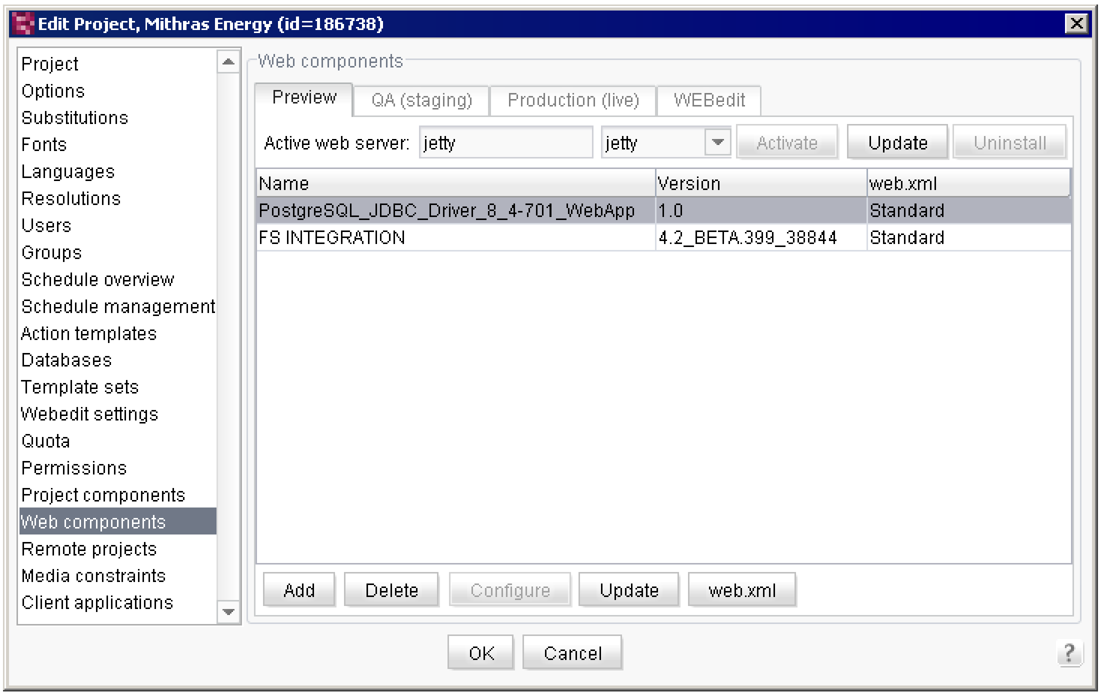
```
# PostgreSQL
jdbc.DRIVER=org.postgresql.Driver
jdbc.PASSWORD=postgresql
jdbc.POOLCYCLE=60
jdbc.POOLMAX=1
jdbc.POOLMIN=1
jdbc.POOLTIMEOUT=180
jdbc.URL=jdbc:postgresql://myServer:3306/mydb
jdbc.USER=postgres
jdbc.layerclass=de.espirit.or.impl.postgres.PostgreSQLLayer
module=PostgreSQL_JDBC_Driver_8_4-701
```

Figure 4-10: Database configuration with parameter `module`

This modification can be saved after a successful test of the connection with a click on the button "OK".

4.8.3.3 Usage in web applications

If the database is to be used in a web application, the module must be added to the desired web component. This is effectuated in the Project properties in the area "Web components". Click on "Add". A list will open from which you can select the module components which are available on the server. Select the component of the JDBC driver module. The name results from the value given by means of `<name>` within the definition of `<web-app>` (see Chapter 4.8.2.2 page 126):



Active web server: jetty jetty

Name	Version	web.xml
PostgreSQL_JDBC_Driver_8_4-701_WebApp	1.0	Standard
FS INTEGRATION	4.2_BETA.399_38844	Standard

Figure 4-11: Project properties – JDBC driver as web component

The component of the JDBC driver module can not be configured furthermore.

4.8.3.4 Usage of the Derby database in web applications

If a Derby database is used in web applications (e.g. in the FirstSpirit module "Integration"), the JDBC driver module must be added to the web application as well, as described in Chapter 4.8.3.3 page 134.

The Derby database, integrated in FirstSpirit, is not dedicated for productive operation and should be used for test purposes only.

4.8.3.4.1 Example: Module "FirstSpirit Integration"

When using the module "FirstSpirit Integration" with a Tomcat web server, the connection configuration must be adjusted for each schema. In this case, the Derby database can be only accessed by means of the TCP port. For this purpose, the parameter `internalDB.port` must be indicated in the configuration file `fs-server.conf`.

In addition, the following parameters must be adjusted in the Configuration of the database layer (see Chapter 4.8.3.2 page 132) for each schema:

`jdbc.URL`: This parameter must point to the TCP port of the Derby database instead of a locale directory. For this purpose, host and port must be added in the existing URL and `create` must be deleted, e.g

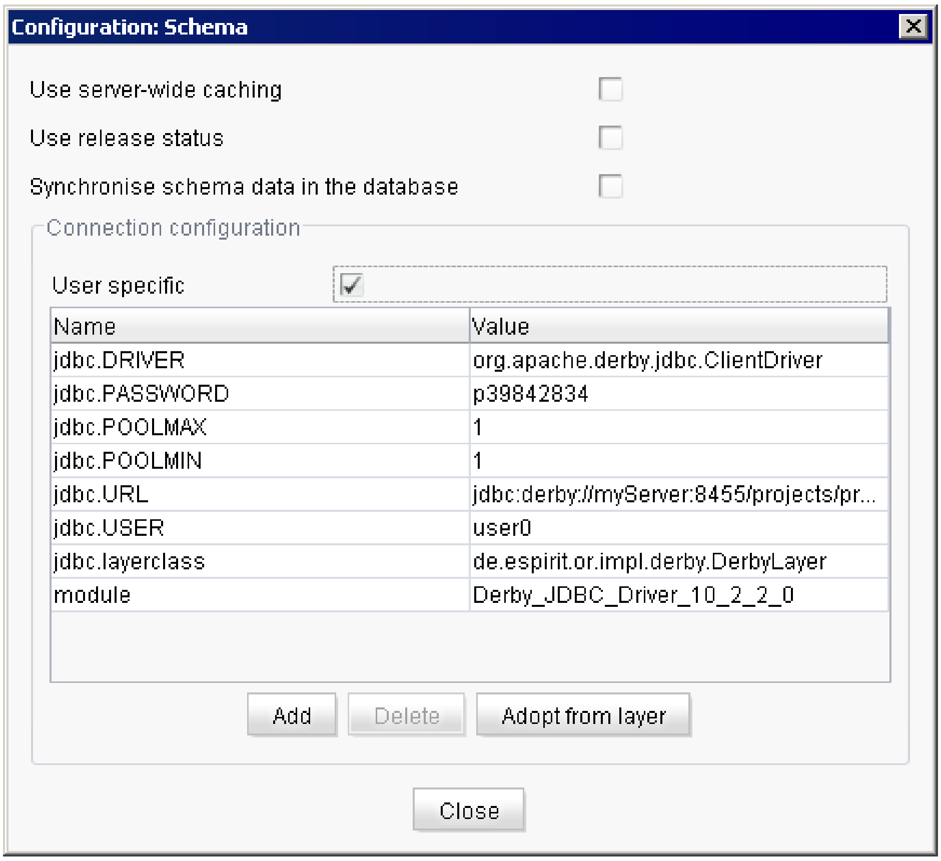
```
jdbc:derby:projects/project_29703/derby;create=true
```

will become

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

`jdbc.DRIVER`: Change this parameter to `org.apache.derby.jdbc.ClientDriver` if you use a Tomcat web server. When using a Jetty web server no adjustment is necessary.

If you use the FirstSpirit module "Integration" the configuration of the JDBC driver module must be updated manually after these modifications, if the option "User specific" is activated in the database connection:



Configuration: Schema

Use server-wide caching □

Use release status □

Synchronise schema data in the database □

Connection configuration

User specific ☑

Name	Value
jdbc.DRIVER	org.apache.derby.jdbc.ClientDriver
jdbc.PASSWORD	p39842834
jdbc.POOLMAX	1
jdbc.POOLMIN	1
jdbc.URL	jdbc:derby://myServer:8455/projects/pr...
jdbc.USER	user0
jdbc.layerclass	de.espirit.or.impl.derby.DerbyLayer
module	Derby_JDBC_Driver_10_2_2_0

Figure 4-12: FirstSpirit Integration – Database configuration

If this option is not activated, the values which are stored currently in the JDBC driver module and in the configuration of the database layer will be assumed.

If the internal Jetty web server is used in parallel to the external Tomcat web server, the database layer must be adjusted as well (see Chapter 4.8.3.2 page 132).

4.8.3.4.2 Individual implementations of modules

You must carry out these adjustments, which are carried out for the example "FirstSpirit Integration" (see Chapter 4.8.3.2 page 132), in each module which has been developed individually and which works together with the Derby database.

4.8.4 Data source configuration

The database integration of the FirstSpirit Server is configured via the Server and Project Configuration (see section 7.3.6 page 220) or directly via “DATABASES” in configuration file `fs-database.conf` (see section 4.3.3 page 71). It is recommended to use the Server and Project Configuration (see section 7.3.6 page 220) to edit the database configuration, since all the changes are automatically written into `fs-database.conf` and updated on the server. Moreover, the database connection can be tested (see Figure 7-41).

The FirstSpirit database connection can be used for various purposes:

1. Creation of “internal data sources”.
2. Integration of external databases (“external data sources”).

Prerequisites for using data sources in FirstSpirit:

- Database as an external or internal content source:
  Licence key with the feature code “content” (fs-license.conf)

- Database configuration via the Server and Project Configuration (recommended) or directly via the file `firstspirit4/conf/fs-database.conf`

It is possible to define as many data sources as desired for a FirstSpirit Server. The data sources can be individually allocated to the projects via the administrator user interface or selected during project import.

The following applies for direct configuration via the file `fs-database.conf`:

- Numerous databases can be listed one after another when separated by commas.

- The names of the external data sources can be freely chosen; nevertheless, the following conventions have to be adhered to.
  Permitted characters: A-Z, a-z, 0-9, _, -

Example:

```
DATABASES=content1, content2, extern1, extern2
```

The following parameters have to be additionally defined in the configuration (schema) for each database element specified in “DATABASES”:

```
<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>
```

Configuration example (for `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
```

For a description of the obligatory parameters see section 4.8.4.1 page 138.

The following parameters can be used optionally:

```
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
```

For a description of the obligatory parameters see section 4.8.4.2 page 140.

4.8.4.1 Description of the obligatory parameters

`<database>.jdbc.DRIVER:` Contains the complete class name of the used JDBC driver which is obligatory for defining the content database (see section 4.8.1). Ensure that the specified class can also be loaded by the FirstSpirit Server. To achieve this, the JAR file, which contains the JDBC driver, has to be stored in the FirstSpirit Server directory `shared/lib`. Each change in this directory demands a server restart. **From FirstSpirit Version 4.2R4** the integration of JDBC driver files as FirstSpirit module is recommended (see also Chapter 4.8.2 page 125).

```
content1.jdbc.DRIVER=com.mysql.jdbc.Driver
```

`<database>.jdbc.URL`: Contains the specification of the JDBC URL to a database server and a database available therein, for example:

```
content1.jdbc.URL=jdbc:mysql://myServer:3306/mydb
```

In this example a MySQL database "mydb" on the database server "myServer" is addressed. The structure of the JDBC connection URL varies from database to database and has to be taken from the respective database documentation (see section 4.8.7 page 148).

`<database>.jdbc.USER`: Valid login name of a database user. The FirstSpirit Server uses this account to establish a connection to the database during runtime.

```
content1.jdbc.USER=db2admin
```

`<database>.jdbc.PASSWORD`: Valid password for the login under `<database>.jdbc.USER`.

```
content1.jdbc.PASSWORD=admin
```

`<database>.jdbc.layerclass`: The class which implements the database layer for this special database system is specified via parameter `layerclass`, for example:

```
content1.jdbc.layerclass=de.espirit.or.impl.mysql.MySQLLayer
```

> *The parameter `layerclass` may not be empty as otherwise errors occur in the configuration of the database link (see chapter 7.3.6 page 220).*

The following layer classes are included in the FirstSpirit standard scope of delivery:

- 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

The Derby DBMS contained in FirstSpirit is not suitable for productive use and should therefore be used for tests only.

4.8.4.2 Description of the optional parameters

`<database>.jdbc.SCHEMA:` This parameter defines the schema on the DBMS (Database Management System) to be used by FirstSpirit. A schema is also frequently called a "database". Under Oracle, it corresponds to one normal user account, in other DBMS, e.g. PostgreSQL, a normal user account can also include several schemata.

If this parameter is defined, it is a **default layer**. In a FirstSpirit project to which default layers only are assigned, a FirstSpirit user cannot create any new additional schemata. Only the FirstSpirit administrator can add further default layers to the project.

To enable the creation of other schemata for FirstSpirit users too, a so-called **DBA layer** is required, but which in most DBMS requires DBA rights (DBA = Database Administrator). `The SCHEMA` parameter is not entered in a DBA layer. The FirstSpirit user can use a DBA layer to independently generate new default layers.

Before FirstSpirit Version 4.2, the following terms were used:
Multi-Project Layer (sic): since FirstSpirit Version 4.2 corresponds to the term "Default Layer" (standard layer)
Single-Project Layer (sic): since FirstSpirit Version 4.2 corresponds to the term "DBA Layer"

For further details of the differences between default layers and DBA layers and their advantages and disadvantages, please refer to the "FirstSpirit Manual for Developers (Basics)", "Data Source" chapter.

Example:

```
database.jdbc.SCHEMA=goodsdatabase
```

For default layers in PostgreSQL databases, `public` must always be given for this parameter!

`<database>.jdbc.POOLMAX`: FirstSpirit uses one separate pool instance for every usage, e.g. one schema node in a project, one deployed web applikation (e.g. FirstSpirit Integration) etc..
The number of unused DB connections which can remain maximally in the pool is defined by POOLMAX. There is no parameter for limiting the maximal number of open connections to a database.

```
content1.jdbc.POOLMAX=15
```

If a value has not been specified, the number of DB connections is limited to the value of POOLMIN + 5.

`<database>.jdbc.POOLMIN:` The minimum number of DB connections which are held available per pool is defined via parameter POOLMIN.

```
content1.jdbc.POOLMIN=10
```

If a value has not been specified, 5 DB connections per pool are held available.

`<database>.jdbc.POOLCYCLE:` The time interval (in seconds) during which FirstSpirit removes expired DB connections from the pool is defined via parameter POOLCYCLE. A DB connection is classified as expired when either the POOLTIMEOUT or the CONNECTIONTIMEOUT has elapsed. If a value is not specified, the minimum value accepted by FirstSpirit – 90 seconds – is set.

```
content1.jdbc.POOLCYCLE=120
```

`<database>.jdbc.POOLTIMEOUT:` The time interval (in seconds) during which the FirstSpirit Server can use a DB connection is defined via parameter POOLTIMEOUT. If the server does not release this connection after the time interval has elapsed, it is closed automatically. If a value is not specified, the value “180” is set by default.

```
content1.jdbc.POOLTIMEOUT=240
```

`<database>.jdbc.CONNECTIONTIMEOUT:` The time interval (in seconds) after which a DB connection is considered as old by the FirstSpirit Server and closed is defined via parameter CONNECTIONTIMEOUT. If a value is not specified, a timeout of 30 minutes (value 1800) is set by default. A value <= 0 deactivates the timeout. This value must always be less than the idle timeout des of the database server, which amounts normally some hours.
Example for setting the timeout to 15 minutes:

```
content1.jdbc.CONNECTIONTIMEOUT=900
```

`<database>.jdbc.CONNECTIONRETRY:` During SQL query execution FirstSpirit tries to use a connection from the connection pool. If there are no free connections, an attempt is made to establish a new database connection. Requests can be rejected (e.g. due to the database configuration). The number of connection attempts to the database is defined via parameter CONNECTIONRETRY. If the number is exceeded, a failed connection attempt is aborted with a warning message. If a value is not specified, the value “5” is set by default.

```
content1.jdbc.CONNECTIONRETRY=3
```

`<database>.jdbc.CONNECTIONRETRYCYCLE:` After a failed attempt to establish a database connection, the ORMapper waits for the time specified in CONNECTIONRETRYCYCLE (in ms) to elapse before trying again. If a value is not specified, the value “300” is set by default.

```
content1.jdbc.CONNECTIONRETRYCYCLE=500
```

`<database>.jdbc.MAXSTRINGLENGTH:` The parameter MAXSTRINGLENGTH determines the maximum number of characters of a VARCHAR column when creating a new DB table. If a higher value is specified for a string attribute than the one defined via parameter MAXSTRINGLENGTH, this string attribute is stored as BLOB or CLOB in the database. (For DB2 this value depends on the size of the used “Page Size” of the used table area.) If a value has not been set here, a default value is set depending on the used database:

- Derby: 32672
- MSSQL-Server: 4000
- Oracle: 2000
- PostgreSQL: 255
- Others: 1024

```
content1.jdbc.MAXSTRINGLENGTH=4000
```

`<database>.jdbc.JNDI:` If the ORMapper runs in a web container or application server, it is possible to establish a database connection via a data source. Therefore, the ORMapper uses the pooling capacity of the web container. The parameter JNDI determines the JNDI name of the used data source.

```
database.jdbc.JNDI=java:comp/env/jdbc/ORMapper
```

`module:` If a FirstSpirit module is used for JDBC driver, you must specify the name of the JDBC driver module by using this parameter. For more information about creating and using JDBC driver modules see Chapter 4.8.2 page125.

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

4.8.4.3 Description of the Oracle-specific parameters

`<database>.jdbc.oracle.TABLESPACES:` When creating a table under Oracle, it is possible to use this parameter to specify the table space used for this table. If this optional parameter is not specified, the value “USERS” is set by default.

```
Database.jdbc.oracle.TABLESPACES=USERS
```

4.8.4.4 Description of the MS-SQL specific parameters

`<database>.jdbc.CATALOG:` The meta data of the datenbases are classified in namespaces. These have a tree structure with the CATALOG name as root node. This parameter restricts the name space for the meta data, the ORMapper works with.

```
database.jdbc.CATALOG=ormapper
```

4.8.5 Required permissions for database user accounts

Depending on the used database specific permissions are necessary. These are described in the following sections.

4.8.5.1 Oracle databases

For DBA layers:

```
CREATE USER <dbuser> IDENTIFIED BY <password>;
GRANT DBA TO <dbuser>;
```

For standard layers:

```
CREATE USER <dbuser> IDENTIFIED BY <password>;
GRANT CONNECT TO <dbuser>;
GRANT RESOURCE TO <dbuser>;
```

4.8.5.2 MySQL databases

For DBA layers:

```
# 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>;
```

For standard layers:

```
# mysqladmin --default-character-set=utf8 create database <dbname>
# mysql
mysql> CREATE USER <dbuser> IDENTIFIED BY <password>;
mysql> GRANT ALL PRIVILEGES ON <dbname>.* TO <dbuser>;
```

Specifying UTF-8 character coding does not make sense until MySQL Version 5 and higher.

The InnoDB storage engine must be enabled on the MySQL server! Advisable MySQL server parameter values for production systems:

```
[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

For DBA layers:

```
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;"
```

For default layers:

```
createdb -E UTF8 myDBname "my DB description text"
createuser –D –A -P -E myDBuser
```

All `createuser` queries can be answered with "No" as, apart from those assigned to DBA layers via `grant create`, extended user privileges are not necessary either for DBA layers or for default layers.

The password authentication (type MD5) must then be entered in the

/etc/postgres/pg_hba.conf file for the given user on the database used. The following must be called to make the change known to the database server:

```
pg_ctl reload
```

4.8.5.4 IBM DB2

Creating the database (DBA layers and default layers):

```
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
```

The last line is only necessary if the default schema "myUser" of the database "myDB" is not yet available.

"myUser" is the JDBC user name which in the case of DB2 is equal to the name of the instance, i.e. for example "db2inst1". If necessary, a different schema name can be used.

The following permissions are necessary for the DB2 user account:

For DBA layers: DBADM

For default layers: CONNECT, CREATETAB, BINDADD, IMPLICITSCHEMA

4.8.6 Notifications and restrictions concerning the specific database systems

This chapter contains notifications and restrictions concerning the use of specific database systems, e.g. concerning the use of specific driver versions, the configuration, the unicode support, restrictions concerning functions etc. Further notes can be found in the respective sub-chapters of Chapter 4.8.7 from page 148.

4.8.6.1 General notifications and restrictions

It is recommended to use a compatible JDBC driver version for the applied database version, except as noted otherwise.

Restrictions

Some databases have restrictions regarding the maximum name length (especially column names) or database line length. Therefore, always observe the following when creating content data structures:

1. All text input fields (or similar) should only be generated as large as really necessary.
2. All column names should be chosen as short as possible.
3. Language-dependent input fields should only be used if they are really required.
4. Not every database can store unicode characters in UTF-8 format. If you are planning to create multilingual projects with unicode characters, ensure that the database being used is unicode-capable and correspondingly configured.

4.8.6.2 MySQL

Unicode: Unicode support from MySQL Version 5.

Further restrictions for MySQL databases (V 4.x - 5.1):

Big tables can not be stored:

- 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."

This means: a table with 11 columns of the type TEXT or VARCHAR (>730) is to big for MySQL. This restriction applies to the following examples:

- 4 languages with 2 DOM input components and a language dependent string column (more than ca. 230 characters) or
- 2 languages with 5 DOM input components plus 1 language independent string column (more tha 320 characters) or
- 1 language with 11 string comlumns (each more than 730 characters)

4.8.6.3 Oracle

! *When using Oracle databases, the saving of database schemes and changes to them can take some time especially since version 4.2R4.*

Unicode: When installing an Oracle database, UNICODE support should be activated to enable all international characters to be displayed. When creating the Oracle instance the following parameters must be set in the Create Database statement:

```
NLS_CHARACTERSET: UTF8

NLS_NCHAR_CHARACTERSET: AL32UTF8
```

UTF16 can also be used, but causes problems with some rarely used special characters, as the Oracle translation table apparently contains gaps for this coding.

Driver: For Oracle, the JDBC driver of the series 10.1 (ojdbc14_10.1.0.x.jar) should be used because problems can arise if the data type `LONG` is used with version 10.2 from 4000 characters on and UTF-8 coding. As an alternative, the compatibility mode for Oracle 9 LONG must be activated when using the driver 10.2, because `LONG` is deprecated since Oracle 9. For this purpose the parameter

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

must be added in the database configuration.

4.8.6.4 IBM DB2

Unicode: The UNICODE support should be activated during DB2 database creation.

Deleting columns: When using DB2, it is not possible to delete columns via the JDBC driver. However, the columns can be deleted in the database schema of the FirstSpirit JavaClient, but remain in the database.

Other notes: The configured heap size of DB2 is too small by default (128 x 4KB) and should be at least 1024x4KB. Execution of the following statement on the DB2 console is recommended:

```
db2 update db cfg for myDB using applheapsz 1024
```

4.8.6.5 Microsoft SQL Server 2000

Unicode: No UNICODE support. Although the SQL server supports unicode it is not possible to store unicode characters in the database via the standard JDBC-ODBC bridge. This problem can be avoided by using a different brigde or a native JDBC driver.

4.8.7 Examples for linking different database systems

The differences in the connection of the different DBMS are shown here in detail by way of examples. If a DBMS is to be prepared before a FirstSpirit installation and uncertainties exist regarding the database administrator's access parameters or the necessary driver files, in most cases it helps to test the database link by means of an external JDBC client. This can be done, e.g. using the DB Visualizer from http://www.minq.se/products/dbvis/.

The following configuration examples show use as a default layer. If used as a DBA layer, the `jdbc.SCHEMA=…` line is omitted.

4.8.7.1 Configuration example: MySQL

Driver: 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
```

It is only necessary to specify `?useUnicode=true&characterEncoding=UTF8` for jdbc.URL if using a UTF8-coded database (MySQL 5 and higher). If using the default MySQL coding (latin1, ISO-8859-1), it is not necessary to specify this parameter.

The Connector/J 5.1 must be used for MySQL 5.0 with FirstSpirit, too, at least in version 5.1.10: http://dev.mysql.com/downloads/connector/j/5.1.html.

4.8.7.2 Configuration example: MS-SQL-Server

Driver: 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=tes
tDB;selectMethod=cursor
jdbc.USER=testuser
jdbc.layerclass=de.espirit.or.impl.mssql.MSSQL2000Layer
```

or

```
jdbc.CATALOG=testDB
jdbc.DRIVER=com.microsoft.sqlserver.jdbc.SQLServerDriver
jdbc.PASSWORD=testpasswort
jdbc.URL=jdbc:microsoft:sqlserver://myserver:1433;DATABASENAME=tes
tDB;selectMethod=cursor
jdbc.USER=testuser
jdbc.layerclass=de.espirit.or.impl.mssql.MSSQL2005Layer
```

For description of the MS-SQL specific parameters see chapter 4.8.4.4 page 143.

4.8.7.3 Configuration example: Oracle

Driver: ojdbc14_x.x.jar

Layer parameters:

```
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
```

For `URL` the instance name of the oracle server is specified as last parameter (in the example `ORCL`) and not the schema name. The schema name given with `jdbc.SCHEMA` complies with the user name specified by `jdbc.USER`.

4.8.7.4 Configuration example: PostgreSQL

Driver: 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
```

If using as a default layer, the value `public` must be entered for the parameter `jdbc.SCHEMA` in PostgreSQL and not the database name.

4.8.7.5 Configuration example: DB2

Layer parameter:

```
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
```

Driver: db2java.zip (must precisely fit the DB2 server used)

Port: DB2 Java connector (is provided via: db2jstrt PORTNUMBER).

From FirstSpirit Version 4.2, a JDBC Type 4 driver must be used to configure an IBM DB2 database:

```
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
```

If using as a default layer, the same value is entered as parameter `jdbc.SCHEMA` in DB2 as for jdbc.USER, if the default schema of the given database (here "myDB") is to be used. Optionally, another schema can be used. In both cases the schema must be created beforehand outside FirstSpirit using the SQL command "create schema myUser;".

Drivers:

- before DB 9.5: db2jcc.jar, db2jcc_license_cu.jar
- since DB 9.5: db2jcc4.jar, db2jcc_license_cu.jar

The JAR files are located on the DB2 server in the directory `db2inst1/sqllib/java`. It is recommended to use always the driver version which is supplied with the DB" server to prevent incompatibilities.

Port: DB2 connector (db2jstrt is not necessary for type 4). Under Unix, the port number can be read out of the /etc/services file, db2_db2inst entry and as a default is set to 50000.

4.8.7.6 Configuration example: Internal Apache Derby database

The FirstSpirit server already contains a simple relational database system (Apache Derby) for test systems. FirstSpirit normally stores all data in the file system (Berkeley database) and – depending on the project requirements – only stores a few in relational databases. As a default, when a new project is created, this default database is activated for the project (see chapter 7.2.3.1 page 189) and write access to the database is set for this project (see chapter 7.4.12 page 281). A Derby database can be subsequently created in the server properties (see chapter 7.3.6 page 220) and added to the project properties of the project (see chapter 7.4.12 page 281).

In order to use the Derby database from external processes, e.g. web application with FirstSpirit module integration in the external application server, the JDBC connector must first be activated for network connections (see chapter 4.3.1.14 page 56). The parameters from the database settings of the respective project are copied first as the connection parameters in the web application, e.g.

```
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
```

Then, in the connection parameters for the web application, lines `jdbc.URL` and `jdbc.DRIVER` are replaced with the following, where the host name of the FirstSpirit server is entered instead of "fs4server" and the Project ID 12345 is replaced by the actual ID:

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

The JDBC driver for integration in the web application can be downloaded as a file, `derbyclient.jar` from http://db.apache.org/derby/, in order to then copy it to `WEB-INF/lib` or into a global `classpath` directory of the application server. The respective active version of the Derby database in FirstSpirit can be read from the log file `firstspirit4/log/fs-database.log`. The Derby version was changed to 10.5.3.0 in FirstSpirit Version 4.2R4.

In order to automatically create the JDBC configuration for the individual web applications, this modified database configuration can also be made directly in the layer settings of the FirstSpirit server. Then `derbyclient.jar` must also be copied to `firstspirit4/shared/lib`.

If precisely 1 schema is to be accessed in the external application, if necessary, extend `jdbc.URL` to include the `DATABASENAME` parameter. For `{SCHEMA-ID}` and `{PROJECT-ID}`, refer to the relevant FirstSpirit project for each.

```
jdbc.URL=jdbc:derby://fs4server:1527/projects/project_12345/derby;
DATABASENAME=P{SCHEMA-ID}_{PROJECT-ID}
```

In FirstSpirit Version 4.2R4 and higher, it is recommended that JDBC drivers be integrated as a FirstSpirit module instead of in `firstspirit4/shared/lib` and manually in `WEB-INF/lib`, so that they are automatically integrated in all FirstSpirit web applications. Then, the parameter `module=JDBC module name` must additionally be given. For details of creation and use of JDBC driver modules, see

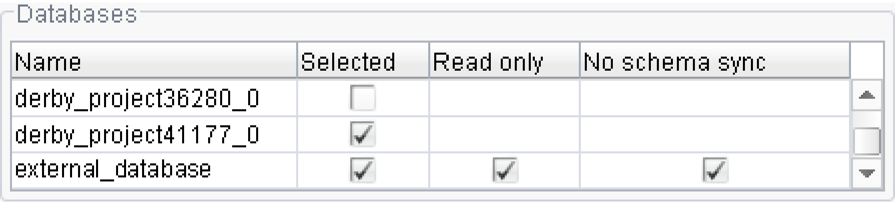
Chapter 4.8.2 page 125.

4.8.8 Procedure for connecting external databases

1. Server and Project Configuration: Configure a new database connection in the **Server properties** (menu item "Server" / "Properties" / "Databases"; see section 7.3.6 page 220). The following entry is for example configured for an external MySQL database:

```
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 and Project Configuration: Check the check boxes "Selected", "No schema sync" and "Read only" for the respective database which should use the configured database (see 1.) in the **project properties** for each project (menu item "Project" / "Properties" / "Databases"; see section 7.4.12 page 281).



Databases

Name	Selected	Read only	No schema sync
derby_project36280_0	☐		
derby_project41177_0	☑		
external_database	☑	☑	☑

Figure 4-13: Configuration of internal (Derby) and of an external database

Structure and contents of the external database must not be changed. In contrast to internal databases, external databases can only provide reading but no writing access.

3. In the **Template Store** of the project for which the database has been "selected" (see 2.) the context menu is now called on the folder "Database schema". Select the respective database via "Create schema from database" to create a database schema for this project on the basis of the external database (see *FirstSpirit Documentation for developers*).
   Depending on the number of tables in the database, the display of these tables in the schema (Template-Store) can take a few minutes.
   If the schema is not automatically unlocked, e.g. because erroneous tables have been detected in the database, the project has to be reopened without previously unlocking the schema manually.

4. The desired table templates and table queries can now be created in the new

schema (see *FirstSpirit Documentation for developers*).

4.9 Roll-out process for native applications (from V4.2)

4.9.1 Roll-out process (server)

A Mozilla engine is used in FirstSpirit within the scope of the "Integrated preview“ (inline preview) function, introduced with FirstSpirit Version 4.2. The native system components required for this must first be distributed to the editors' workstation systems.

In FirstSpirit Version 4.2, the `fs-server.jar` file contains a current version of all platform-dependent, native components of the browser integration (client applications). As a default, when the server is started, the client applications are extracted into the `~FS42\data\clientapps` directory. The directory for the roll-out process of the client application files can be configured using the `CLIENTAPP_PATH` parameter in fs-server.conf (see Chapter 4.3.1.8 page 46).

Each client application is then located in an individual directory below `~FS42\data\clientapps` (default setting), for example, the Mozilla engine in Windows operating systems under:

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

An application consists of operating system dependent files and common files. The common files are located in the "common" directory. The operating system dependent files are located in separate, operating system-specific directories, for example, `clientapps\JXBrowser\windows` or `clientapps\JXBrowser\mac`.

Server version management: If the respective application directory contains a `version.txt` file, the application is subject to server version management and, if necessary, is updated with each server start.

If `version.txt` is deleted, the application is no longer updated by the server and must be managed manually.

4.9.2 Roll-out process (workstation computer)

The directories created on the server on rolling out the native system components (see Chapter 4.9.1 page 153), are created as an exact copy in the User home directory of the editors' workstation computer. To this end, all components which match the workstation computer's operating system are automatically identified. These components are rolled out on the workstation computer, in the editor's User home directory `~\Documents and Settings\<USER>\.firstspirit\`, **from 4.2R4** `~\Documents and Settings \<USER>\.firstspirit_<FirstSpirit-Major-Version>.<FirstSpirit-Minor-Version><FirstSpirit-Release-Version>`. All corresponding client applications then lie below this directory, for example, in Windows operating systems the Mozilla engine is under:

```
C:\Documents and Settings\<USER>\.firstspirit\jxbrowser\xulrunner
```

for FirstSpirit Version 4.2R4:

```
C:\Documents and Settings\<USER>\.firstspirit_4.2R4\jxbrowser\xulrunner
```

From FirstSpirit version 4.2R2 the directory to be used for rolling out the client applications can be defined via the parameters `CLIENT_HOME_DIR` or `CLIENT_HOME_DIR_WINDOWS` in the file `fs-server.conf` or in the connection settings (see Chapter 4.3.1.8 page 46 and Chapter 6.3.5.1 page 173). The evaluation order is as follows:

1. First, the operating-system specific path details are evaluated, which are set in the **connection settings** (e.g. `CLIENT_HOME_DIR_WINDOWS`).
2. Then the path information set in the **connection settings** using the parameter `CLIENT_HOME_DIR` are evaluated.
3. Then the operating system-specific path details defined in **`fs-server.conf`** are evaluated (e.g. `CLIENT_HOME_DIR_WINDOWS`).
4. Then path information set in the **`fs-server.conf`** file using the parameter `CLIENT_HOME_DIR` are evaluated.
5. If the parameter is not set, either in the connection settings or in `fs-server.conf`, as a default the operating system-specific user home directory is used.

The information, which can be set server-wide for all users using `fs-server.conf`, can therefore be overwritten on a user-specific basis.

If a directory is given to which the respective user does not have any access rights, a corresponding exception is output. The respective Client application is then not rolled out and cannot be used.

! *With 4.2R4 the JxBrowser which is used for the Integrated preview has been updated to the version 2.5 (with Gecko Engine 1.9.2 / Firefox 3.6). In some cases this update can lead to errors (e.g. `java.lang.UnsatisfiedLinkError`), and the Integrated preview will not be displayed. This problem can be solved by installing the Microsoft Visual C++ 2008 SP1 Redistributable Package (x86).*

4.9.3 Updating the native system components

A further objective of the roll-out process for native application is central configuration and updating of the individual components. For example, proxy settings or plug-ins for the Mozilla engine can be managed centrally and distributed to the individual workstation computers.

The directories are synchronised by means of hash values which clearly describe the complete (operating system specific) sub-tree for each client application. This value is cached and saved in a CRC.TXT file, which is located in the respective application directory. This file exists on the client and server.

Server-side changes: Each time a configuration change is made on the server, a new hash value is calculated which enables the client to recognise that a change has been made:

- The hash values are automatically recalculated if the server is restarted following a change. On establishing a connection with a client computer, the hash value is then compared to the workstation computer's files. If there is a difference the file system of the changed client applications is transferred from the server to the workstation computer concerned. Only changed files are transferred.
- The hash value must be removed manually if manual changes are made to the client application files on the server. This can be done by deleting the CRC.TXT file in the operating system-specific client application directory of the server, for example, in Windows operating systems under:

```
~FS42\data\clientapp\jxbrowser\windows\crc.txt
```

  On establishing a connection with a client computer, the server recognises that the file is missing and automatically calculates a new hash value. This hash value is compared with the files of the workstation computer and then the update is started.

Client-side changes: The native system components are updated in one direction only; from the server to the client computer. This means that the client-side hash value is not recalculated if a manual change is made to the client-side client application files. The next time the client is restarted, the manual changes are therefore retained and are not synchronised with the server-side files (as both hash values are identical).

To reset inadvertent changes to the client-side client application files to the status of the server, firstly, the CRC.TXT file in the operating system specific client application directory of the workstation computer must be deleted, for example, in Windows operating systems under:

```
C:\Documents and Settings\<USER>\.firstspirit\jxbrowser\crc.txt
```

from 4.2R4 e.g.

```
C:\Documents and Settings\<USER>\.firstspirit_4.2R4\jxbrowser\crc.txt
```

On establishing a connection with a server computer, the server now recognises that the file is missing and automatically starts to transfer the file system of the changed client applications from the server to the workstation computer concerned.

4.9.4 Preventing the overwriting of files during the roll-out process

With each update of the native system components on the server (via the fs-server.jar file) and with each ensuing update of the workstation computer, existing configuration settings can be overwritten. If this overwriting is unwanted, file system write protection must be set for these files. This means, all files changed on the client or server side can be assigned write protection and are then protected against overwriting in the event of an update. In this case, when the files are updated, a warning is logged as information for the user.

5 FirstSpirit web application configuration

> ! *The simultaneous login via a web browser (e.g. in different windows or tabs) to several FirstSpirit servers with the same host name (e.g. myServer:8200 and myServer:8400) is not supported.*

5.1 FirstSpirit start page configuration (fs4root)

The default FirstSpirit Server start page is located under:

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

A valid login is required for calling the start page (see chapter 6 page 168). The start page can be used to provide users with simple access to the FirstSpirit-Clients (see section 6.3 page 170).

The FirstSpirit start page can be configured and adapted via the FirstSpirit Server and Project Configuration (see section 7.3.8 page 226).

The quick start entries can also be configured via the FirstSpirit Server and Project Configuration (see section 7.3.9 page 227).

There is also the possibility to configure the connection settings user-specifically (see section 6.3.5.1 page 173).

5.2 WebClient configuration

WebEdit has been developed as an extension to the editorial system FirstSpirit JavaClient. The WebEdit mode provides a browser-based user interface for quick and simple management of editorial contents. Authors can immediately use the various functions of the FirstSpirit editorial environment, since, in contrast to the FirstSpirit JavaClient installation, a Java environment (JRE) is not required for WebEdit. Technically speaking, WebEdit only works on the basis of HTML and JavaScript.

> ! *The WebEdit configuration occurs among other things via configuration file `fs-server.conf` under "Web Applications" (see Chapter 4.3.1.7 page 43) and "WEBedit configuration" (see section 4.3.1.15 page 56). Before using WebEdit for productive operations, the configuration should be adapted to the respective requirements.*

From FirstSpirit Version 4.1 WebEdit is also available as project local, configurable web application. This means that a separate WebEdit instance can be installed, configured and activated for each project in the server and project configuration (see Chapter 5.2.2 Seite 159).

Certain prerequisites have to be fulfilled before using WebEdit in FirstSpirit projects (see section 5.2.1 page 158 and section 5.2.2 page 159).

All configuration files have been optimised for application with internal Jetty and should be executable immediately after FirstSpirit installation. Only change the configuration if an external application server is to be used instead of the default configured internal Jetty (see section 4.6 page 115).

5.2.1 Project prerequisites when using WebEdit

The following basic requirements have to be fulfilled before using the WebEdit mode in FirstSpirit projects. Check these requirements prior to application:

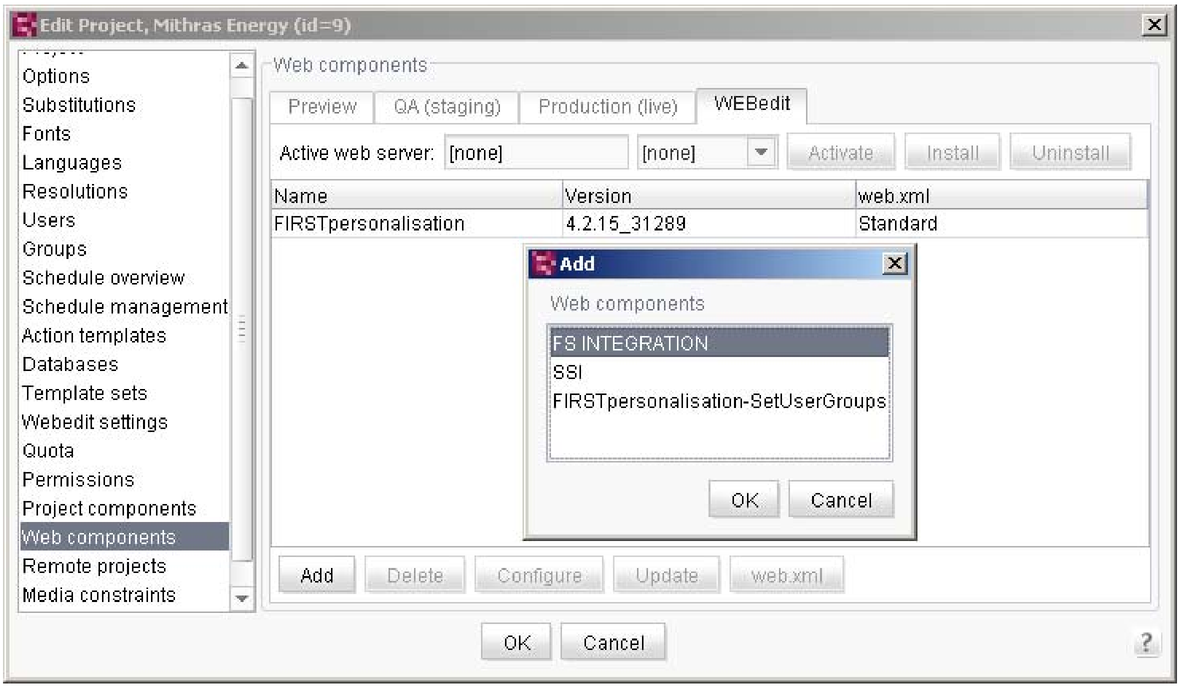
1. WebEdit should support all input components required in the projects. Pay special attention to complex input components of the Content-Store, e.g. TABLIST or OBJECTCHOOSER. Also check customer-specific components properly.
2. Browser compatibility:
   - Which browser is used?
   - Which security settings are used in the company? Does WebEdit run in this configuration?
3. Proxy / Firewall configuration: Is a proxy or firewall used? Does WebEdit run in this configuration (especially the refresh problem)?
4. Secured access: Is it safe to release the http/https port of the FirstSpirit Server for WebEdit users (if necessary also externally?
5. Screen resolution: min.: 1024x768 pixel

5.2.2 WebEdit as a local project application (from V4.1)

WebEdit is now also available as a configurable, local project web application. This means that a separate WebEdit instance can be installed, configured and activated for each project via the server and project configuration.

To do this, the "Web Components" area not only includes the web areas available to date: "Preview", "QA" and "Production", but also the new "WebEdit" area. The tab can be used to configure a WebEdit instance for the selected project (see Chapter 7.4.17 page 289).

Other web components can be added to the WebEdit instance (e.g. FirstSpirit Integration, FirstSpirit Personalisation). These web components can be individually configured for this web area ("WebEdit"). Both default configuration using the "Configuration" button and manual editing of the "web.xml" file are possible.



Edit Project, Mithras Energy (id=9)

Active web server: [none] [none]

Name	Version	web.xml
FIRSTpersonalisation	4.2.15_31289	Standard

Add

Web components

- FS INTEGRATION
- SSI
- FIRSTpersonalisation-SetUserGroups

Figure 5-1: WebEdit as a local project web application

The local project configuration enables an "advanced" WebEdit instance to be configured and to be installed and activated on the required web server. All web components added are taken into account. When installing on the required web server, all the required components are installed and a web.xml is generated, which summarises all files configured to date (web.xmls of the individual components) to form one file.

This means that WebEdit can now be combined with a project-specific personalisation configuration for individual projects.

If the software on the server is updated, for example with a new WebEdit version, it is possible to update all local project WebEdit instances.

5.2.3 Browser configuration when using WebEdit

Problems may arise while copying, cutting or pasting texts into the input component DOM editor when working with the Firefox web browser. (It might not be possible to save contents or to paste them repeatedly.) This behaviour does not result from WebEdit malfunctioning, but a browser security setting. For security reasons (default setting), Firefox prevents the pasting or changing of contents from the clipboard via JavaScript. However, these contents have to be prepared for the DOM editor.

The function can be activated via the respective configuration of the browser settings (in file “user.js”).

> ! *For security reasons, this setting should NOT be carried out globally (for all URLs), but only for the required 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");
```

Several sites can be specified for `capability.policy.allowclipboard.sites` if separated via spaces:

```
user_pref("capability.policy.allowclipboard.sites", "http://aServer:10000
http://aServer:11000");
```

Please contact your system administrator if problems arise with the browser configuration.

5.2.4 Activate scripts in WebEdit

In WebEdit, scripts can be executed when embedded in workflows or individually. Certain settings are required to implement the script functions:

At first, the following lines in the file “fs-webapp.xml.full” in the WebEdit directory

Verzeichnis (FirstSpirit Server/www/WEBedit/WEB-INF) have to be adapted to the server configuration.

```
<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>
```

Normally it is adequate to accept the port from configuration file “fs-webapp.xml” (see section 8.6.1.6 page 399). Subsequently transfer the manual changes to the “fs-webapp.xml” files into this file one after the other.

After carrying out the changes, create a backup copy of “fs-webapp.xml”, rename file “fs-webapp.xml.full” in “fs-webapp.xml” and restart the server.

This completes all server-sided changes. To retain the changes to the WebEdit format templates in the project - in which scripts are to be used – uncheck the box “Use WebEdit” in the project properties under WebEdit settings via the Server and Project Configuration (see section 7.4.14 page 284).

The query whether WebEdit templates should be removed must be answered with “Yes”. Subsequently, reactivate the check box under “Use WebEdit”. Script functions are then fully available in the project.

For further information on using scripts in WebEdit see the FirstSpirit Manual for Developers (Part 1 - Basics).

5.2.5 Prevent "directory browsing"

“Directory browsing” is activated by default in the “Jetty” servlet engine provided by FirstSpirit. If directory browsing is to be prevented, e.g. for company use, adapt file `"webdefault.xml"`.

Replace the following lines:

```
    <init-param>

      <param-name>dirAllowed</param-name>

      <param-value>true</param-value>

    </init-param>
```

with:

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

In order to define an independent error page, additionally insert the following lines in front of the closing `"web-app"` tag:

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

The actual URL to the error page has to be specified here instead of "*URL*"...

> ! *Deleting a start page in the Site-Store can lead to "directory listings" in WebEdit. This behaviour can be prevented by ensuring that a "substitute page" (maybe an error page) is defined in the project properties for all WebEdit projects (see section 7.4.3 page 264).*

5.2.6 Configure WebEdit tree presentation

WebEdit distinguishes between the reference name and symbolic name of the object. Each object has just one reference name, but can, depending on the language, have several symbolic names.

Example: The Site-Store menu levels are displayed as navigation entries on a website. If a new menu level is created in the Site-Store in WebEdit, this new menu level is, e.g., allocated with the reference name "company". Furthermore, there are the two project languages "German" and "English" which are allocated with the menu names "Unternehmen" and "Company" respectively for the new menu level. The reference name "company" is always used for referencing in the project. This reference name is also displayed in the tree view of the Site-Store. However, the names of the navigation entries are displayed language-dependently via the menu names "Unternehmen" or "company".

Since this discrepancy between menu names in the navigation and reference name in the tree view may confuse the editor, the presentation of the tree view can be configured via "fs-server.conf". The tree view presentation can be configured via parameter showReferenceName (see section 4.3.1.15 page 56):

- Value “1”: If the parameter has been set to “1”, the WebEdit tree view presentation is analogue to the one of the FirstSpirit JavaClient. This means: All menu levels of the Site-Store and all objects of the Media-Store are displayed with their reference names in the tree presentation.

- Value “0”: If the parameter has been set to “0”, all menu levels of the Site-Store and all objects of the Media-Store are displayed with their language-dependent menu names or file names in the tree presentation. If language-dependent names are used, additionally check the box “Display name language-dependent” in the Project Configuration (see section 7.4.2 page 257).

5.2.7 Configure workflows in WebEdit

5.2.7.1 Start workflows on page references

Until now, it was only possible to start a workflow on a page of the Page-Store in the WebEdit toolbar. However, it is often necessary to start the workflow on the page reference. The configuration can be adapted via parameter `webedit.sitestoreWorkflow` in file “fs-server.conf” (see section 4.3.1.15 page 56):

- Value “1”: If the parameter has been set to “1”, click on the “Start workflow” icon in the WebEdit symbol bar to start the workflow on the page reference in the Site-Store.
- Value “0”: If the parameter has been set “0”, click on the “Start workflow” icon in the WebEdit symbol bar to start the workflow on the page in the Page-Store.

5.2.7.2 Configure email notifications

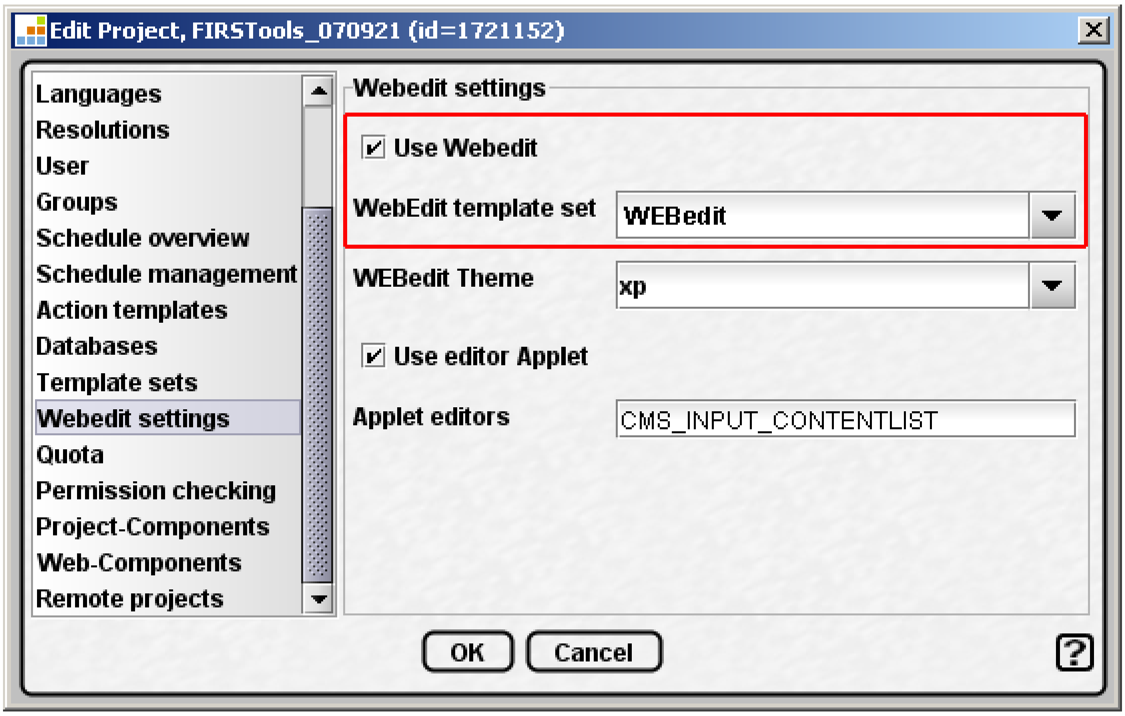
It is possible to specify a notification via email in the JavaClient Template-Store if a workflow reaches a certain activity. It is possible to transfer a reference to a preview page in WebEdit via the placeholder %WEBeditURL% (see the FirstSpirit Documentation for Developers section “Properties of an activity” for further information on workflow configuration).

5.3 Single Sign-On (SSO) configuration

5.3.1 SSO in connection with the JavaClient

If WebEdit is used in a project, the preview from the editorial environment JavaClient can also be generated via WebEdit. Until now, users had to re-authenticate in WebEdit each time they requested a preview from the JavaClient. Due to the “Single Sign-On” function users no longer need to additionally login in WebEdit for a preview from the JavaClient (activated by default).

This is achieved by checking the box “Use WebEdit” and specifying the path “WebEdit” for the “WebEdit template set” in the project settings under menu item “WebEdit settings” (see section 7.4.14 page 284):



Edit Project, FIRSTools_070921 (id=1721152)

Webedit settings

☑ Use Webedit

WebEdit template set: WEBedit

WEBedit Theme: xp

☑ Use editor Applet

Applet editors: CMS_INPUT_CONTENTLIST

Figure 5-2: Configure WebEdit for a preview from the JavaClient

> *Set parameter “webedit.webSSO” in configuration file fs-server.conf to the value “1” (see section 4.3.1.15 page 56).*

Further configuration settings are not required in addition to those described. The configuration file “fs-webapp.xml” and the SSO filters in file “web/fs4webedit/WEB_INF/web.xml” have already been configured for utilisation of

SSO.

For a preview from the JavaClient the SSO module is now used for WebEdit authentication.

If the JavaClient is closed, all the WebEdit sessions generated via the JavaClient also become invalid. In this case, users have to login again.

5.3.2 WebEdit with portal authentication

This method is required if the user login process should not occur via the WebEdit login dialog, but, e.g., via a portal environment. In this case, deactivate the normal login dialog. And extend configuration file `fs-jaas.conf` as follows (websso module) (see section 4.3.4 page 72):

```
websso {

de.espirit.firstspirit.server.authentication.FSTicketLoginModule sufficient;

de.espirit.firstspirit.server.authentication.SAPLoginModule

de.espirit.firstspirit.server.authentication.FSUserLoginModule optional;

};
```

Subsequently allocate the configuration settings for the login process in WebEdit to this module (see section 7.3.13 page 231).

5.3.3 Login with plain text password

For security reasons, the password is encrypted with JavaScript by default during login in WebEdit. This ensures that it is not transferred in plain text. This configuration can be carried out in file `fs-server.conf` via parameter `plainPassword`. If the parameter has been set to “0”, the password will not be transferred in the plain text.

```
plainPassword=0
```

In addition to the configuration of `fs-server.conf`, `fs-jaas.conf` has to be adapted (see section 4.3.4 page 72):

```
websso {

de.espirit.firstspirit.server.authentication.FSTicketLoginModule sufficient;

de.espirit.firstspirit.server.authentication.FSUserLoginModule optional
hash="true";

};
```

5.3.4 Restrictions for using WebEdit

A detailed list with the restrictions for using WebEdit in projects can be found in the FirstSpirit Online Documentation, section "Template Development" / "WebEdit" / "Restrictions".

The following are not planned for WebEdit:

- Changes in the Template-Store (i.e. all types of templates, workflows and the content schemata CANNOT be edited via WebEdit.)
- The support of variables in the Site-Store
- Resolutions in the Media-Store
- The support of document groups

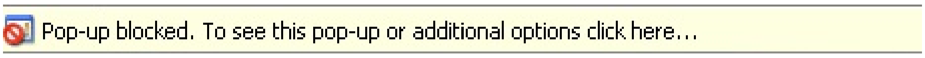
5.4 Pop-up blocker deactivation

The application of Windows XP SP2 involves numerous changes which also affect utilisation of the FirstSpirit web applications.

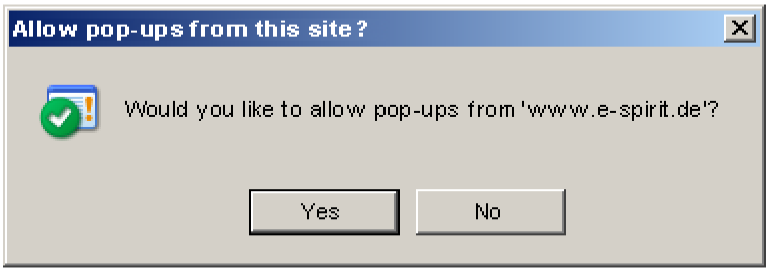
Prerequisite for utilisation of the FirstSpirit web applications is the deactivation of the pop-up blocker for the FirstSpirit Server domain.

The following measures must be taken to ensure that the applications can be used as usual (example: Internet Explorer).

If a pop-up is blocked, a message in a yellow bar appears in the browser window. A pop-up block icon is additionally displayed in the bottom status bar of the browser .


Pop-up blocked. To see this pop-up or additional options click here...

Right mouse click on the message to select "Always allow pop-ups from this site" in the context menu.


Allow pop-ups from this site?

Would you like to allow pop-ups from 'www.e-spirit.de'?

Figure 5-3: Allow pop-ups

If the query is confirmed with “Yes”, the restrictions for working with the FirstSpirit web applications are lifted. The browser displays the “pop-up allowed” icon in the bottom window bar ✓.

6 FirstSpirit start page

Initial access to the FirstSpirit Server usually occurs via the Internet. The default connection to the FirstSpirit Server is established according to the settings during installation (see FirstSpirit Installation Instructions).

Depending on the login process configuration, login can occur automatically (section 6.1 page 168) or by specifiying the user name and password (section 6.2 page 168). The login process is configured via configuration file `fs-jaas.conf` (for parameters see section 4.3.4 page 72).

! *The simultaneous login via a web browser (e.g. in different windows or tabs) to several FirstSpirit servers with the same host name (e.g. myServer:8200 and myServer:8400) is not supported.*

6.1 Automatic login via Single Sign-On

If the server has an SSO-capable login module, users can be automatically authenticated (e.g. with their Windows login) at the FirstSpirit Server. For this, configure parameter `jaas.default` for SSO (for the JAAS[18] configuration see section 4.3.1.6 page 43). Use the FirstSpirit Server and Project Configuration to carry out the respective configuration (see section 7.3.13 page 231).

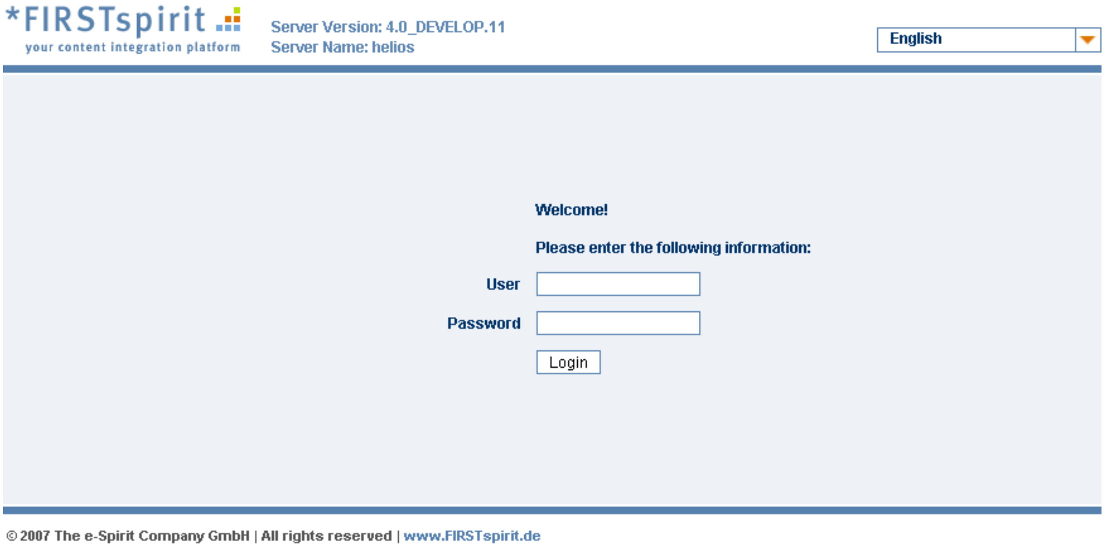
When calling the start page, the system checks whether automatic login is possible. If the server has an SSO-capable login module and simultaneously uses the Internet Explorer, users are automatically logged in at the FirstSpirit Server under their Windows login and the start page is opened (section 6.3 page 170). If users have not yet been registered under their Windows login at the server, they will be created as external users.

6.2 Login with user name and password

If automatic login fails (or if login via SSO is not configured), a login page will be displayed. Users can log in at the FirstSpirit Server via the login window. This login is

[18] Java Authentication and Authorization Service (Information: http://java.sun.com/products/jaas/)

valid for all applications on the server and is maintained for a certain period even for inactive users.



Server Version: 4.0_DEVELOP.11
Server Name: helios

English ▾

Welcome!

Please enter the following information:

User

Password

Login

© 2007 The e-Spirit Company GmbH | All rights reserved | www.FIRSTspirit.de

Figure 6-1: Login

Information on the name and version of the FirstSpirit Server is located at the top of the page:

Server version: The version is automatically provided by the server.

Server name: Name of the FirstSpirit Server. If a symbolic name has been defined for the server in configuration file `fs-server.conf`, it is displayed on the start page (see section 4.3.1.1 page 34). If a symbolic name has not been defined, the host name from the access path is displayed, i.e. the server name “myServer”, when calling via http://www.myServer.de:4050.

English ▾ Set a language for further work with FirstSpirit via the select box at the top right of the page.

User: Enter the user name under which the user is logged in at the FirstSpirit Server in this field.

Password: Enter the user’s password in this field.

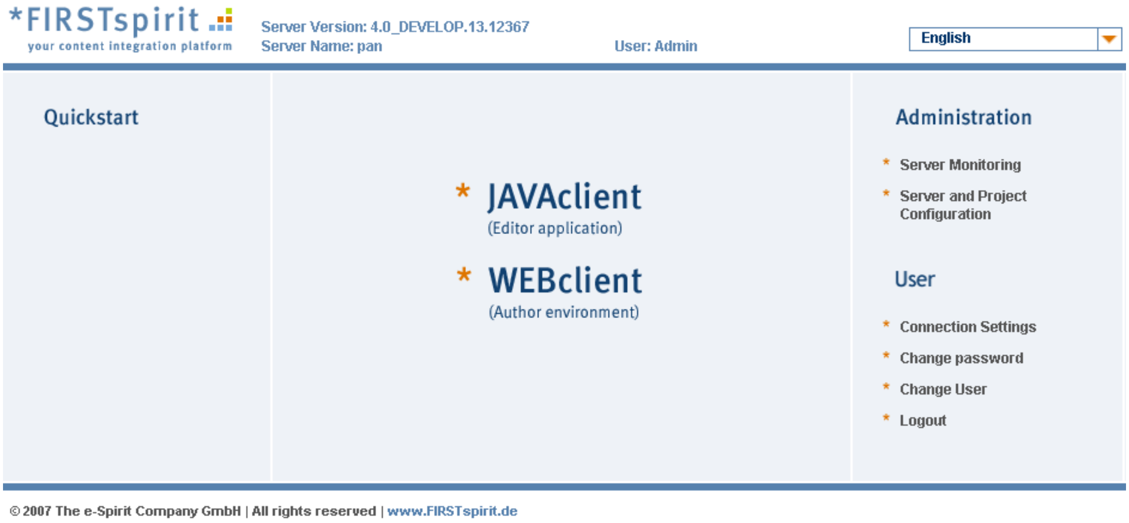
Click on Login to login under the entered user name.

6.3 FirstSpirit start page

The FirstSpirit start page opens after automatic or manual login. The start page is divided into various sections which are, depending on the respective user permissions, either shown or hidden:

- Quick start area (see section 6.3.1)
- Client area (see section 6.3.3)
- Administrator area (see section 6.3.4)
- User area (see section 6.3.5)

Sun Java Runtime Environment (JRE) Version 1.5.0 or later (version 1.6.0. is recommended), including Java Web Start, is required to start the Server and Project Configuration and the JavaClient. (The JRE is usually automatically installed during JDK 1.5.0. or 1.6.0. installation).



*FIRSTspirit your content integration platform

Server Version: 4.0_DEVELOP.13.12367
Server Name: pan

User: Admin

English

Quickstart

* JAVAclient (Editor application)

* WEBclient (Author environment)

Administration

* Server Monitoring
* Server and Project Configuration

User

* Connection Settings
* Change password
* Change User
* Logout

© 2007 The e-Spirit Company GmbH | All rights reserved | www.FIRSTspirit.de

Figure 6-2: FirstSpirit start page

Information on the name and version of the FirstSpirit Server is located at the top of the page. The current logged-in user at the server is also displayed.

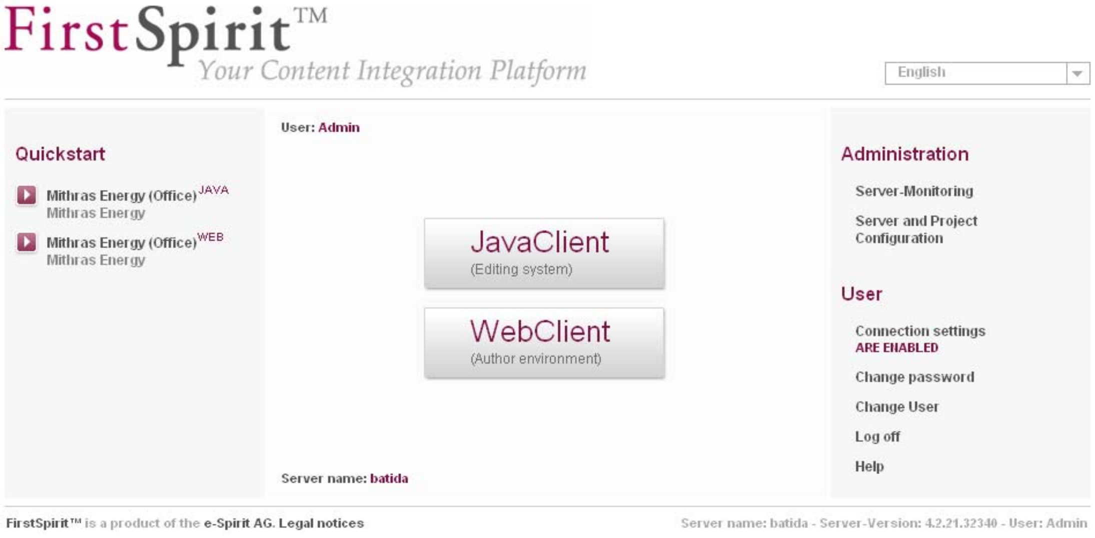
English

Set a language for further work with FirstSpirit via the select box at the top right of the page.

(See chapter 5, page 157 ff for configuration of the web application FirstSpirit start page.)

6.3.1 Revised start page in FirstSpirit Version 4.1

The FirstSpirit start page was also revised as part of the launch of the new "look & feel":



FirstSpirit™ Your Content Integration Platform

English

Quickstart

Mithras Energy (Office) JAVA
Mithras Energy

Mithras Energy (Office) WEB
Mithras Energy

User: Admin

JavaClient (Editing system)

WebClient (Author environment)

Server name: batida

Administration

Server-Monitoring

Server and Project Configuration

User

Connection settings ARE ENABLED

Change password

Change User

Log off

Help

FirstSpirit™ is a product of the e-Spirit AG. Legal notices

Server name: batida - Server-Version: 4.2.21.32340 - User: Admin

Figure 6-3: New start page

Apart from the modified display, several items of additional information is also displayed on the start page:

- Name of the FirstSpirit server
- FirstSpirit version used
- Name of the logged in user

Using the language selection in the top right-hand part of the page it is now possible to select Italian as the language setting for the menu labelling, context menus and dialogs of the FirstSpirit applications.

6.3.2 Quick start

Quick start entries, which are directly linked to a project, are located on the left-hand side of the page. The client (JavaClient or WebClient) which has been configured for the entry is automatically started and the selected project opened via these entries. Only the projects which the logged-in user is permitted to open are displayed in this list (see section 7.3.9 page 227).

6.3.3 Client start

The middle of the page displays entries for starting the FirstSpirit-Clients.

- **JavaClient (editorial environment):** Click on this entry to start the FirstSpirit editorial system. The editor can select the desired project. A connection to the server is automatically established (see section 6.4).

- **WebClient (author environment):** Click on this entry to start the FirstSpirit author environment via a browser. In contrast to the JavaClient, the FirstSpirit author environment offers a limited functional scope for editorial work with FirstSpirit (see section 6.4).

6.3.4 Administration

The right-hand side of the page is further divided. At the top there are the entries for the server and/or project administrators.

- **Server Monitoring:** Click on this entry to open Server Monitoring. See chapter 8 (page 376) for a detailed description.

- **Server and Project Configuration:** Click on this entry to open a console for the FirstSpirit Server and Project Configuration. See chapter 7 (page 184) for a detailed description.

! *This area is only visible for server and project administrators. (If a project is deactivated, the entries for the project administrators of this project will be hidden.)*

6.3.5 User

At the bottom there are the user settings of the currently logged-in user:

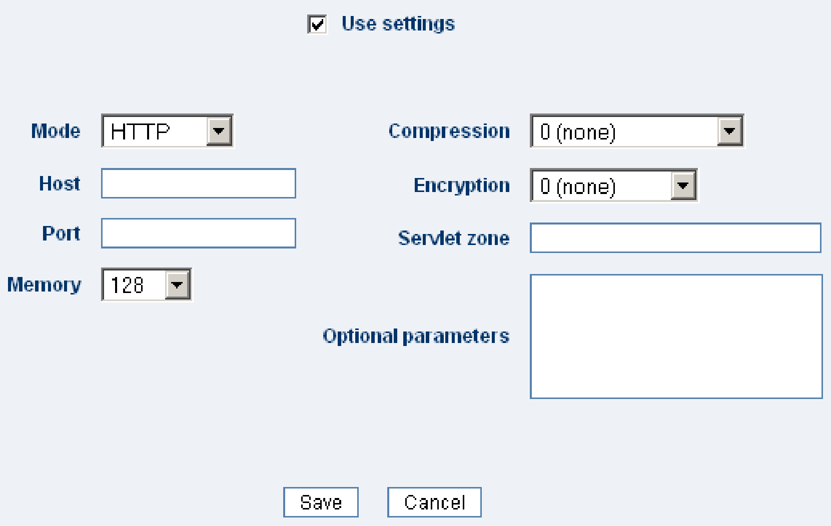
- **Connection settings:** It is possible to change the connection settings of the currently logged-in user here (see section 6.3.5.1 page 173).

- **Change user:** In some cases you might wish to authenticate yourself at the FirstSpirit Server under a different user name, e.g. to log in as server administrator (see section 6.3.5.3 page 177).

- **Change password:** It is possible to change the password of the currently logged-in user here (see section 6.3.5.2 page 176).

- **Logout:** Click on this entry to terminate the current FirstSpirit session of the logged-in user (see section 6.3.5.4 page 177).

6.3.5.1 Configure connection settings



☑ Use settings

Mode: HTTP

Host:

Port:

Memory: 128

Compression: 0 (none)

Encryption: 0 (none)

Servlet zone:

Optional parameters:

Figure 6-4: Configure connection settings

Connection settings of the currently logged-in user for starting the JavaClient and the Server and Project Configuration. The values configured here overwrite the server-sided web start settings for this user (see section 7.3.8 page 226). Only change the settings for test purposes.

Mode: It is possible to set the connection mode for default communication between FirstSpirit-Clients and Server for the currently logged-in user in the fold list:

- HTTP: Normal Internet connection (default setting)

- Socket: Direct connection mode.

Host: Server name or IP address of the FirstSpirit Server to which the client should connect during web start.

Port: Port number of the FirstSpirit Server.

Memory: Specify the memory space (in MB) which is to be provided for the virtual machine of the Client here. Valid values are 128m, 256m, 512m or 1024m.

Compression: Compression for communication between FirstSpirit-Clients and Server for the currently logged-in user:

- None: No compression for data transfer between Client and Server.
- Deflate: Utilisation of the deflate algorithm with default compression for data transfer between Client and Server.
- Deflate speed: Utilisation of the deflate algorithm with the quickest possible compression for data transfer between Client and Server.
- Deflate best: Utilisation of the deflate algorithm with the best compression for data transfer between Client and Server.

Encryption: Encryption for communication between FirstSpirit-Clients and Server for the currently logged-in user:

- None: No encryption for data transfer between Client and Server.
- TLS[19]: Utilisation of the TLS protocol for data transfer between Client and Server.
- DH ARC4: Utilisation of the DH ARC4 encryption algorithm for data transfer between Client and Server.

Servlet zone: Specification of the servlet zone (the servlet zone corresponds to the mapping of the root application URL (see parameter `WEBAPP_ROOT_URL` in section 4.3.1.7 page 43).

Optional parameters: Optional parameters for the Webstart configuration can be saved in this field. The parameters can be given in the following form, consecutively and separated by a semi-colon: PARAMETER1=VALUE1; PARAMETER2=VALUE2. The optional parameters partly correspond to the setting options which can be entered within the input fields of the dialog (cf., e.g. encryption, compression):

[19] Transport Layer Security

- `compression`: Optional parameter for the compression (for the communication between FirstSpirit Clients and Server).
  Possible values: 0 (none), 1 (Deflate), 2 (Deflate Speed), 3 (Deflate Best)
  Example: `compression=3`
- `encryption`: Optional parameter for the encryption (communication between FirstSpirit Clients and Server).
  Possible values: 0 (none), 1 (TLS), 2 (DH ARC4)
  Example: `encryption=3`
- `login`: Optional parameter for specifying the login module (plain, sso). Example: `login=plain`
- `autologin`: Optional parameter for specifying user login and user password. Both values are transferred in plain text separated by ":". The password can be omitted here; however the colon must always be transferred.
- `host`: Optional parameter for specifying the host name (preselection of the server).
- `port`: Optional parameter for specifying the port number (integer).
- `mode`: Optional parameter for specifying the connection mode (http, socket).
- `httpproxy`: Optional parameter for specifying the proxy to be used in the HTTP connection mode. If this parameter is given, only this proxy is used. If the parameter is not specified, the system tries to evaluate the Java Webstart proxy configuration. This evaluation can also be prevented by specifying the parameter `nohttpproxy=1` .
- `httpsproxy`: Optional parameter for specifying the proxy to be used in Socket connection mode. In this case the proxy is used to tunnel the TCP connections.
- `nohttpproxy`: Optional parameter to prevent evaluation of the Java Webstart proxy configuration (cf. `httpproxy`).
  Example: `nohttpproxy=1`
- `usehttps`: Optional parameter used to define whether communication in HTTP connection mode is transmitted over the secure HTTPS protocol (value=1) or not (value=0).
  Example: `usehttps=1`
- `proxybypass`: Optional parameter, used to define which hosts the proxy may bypass. In HTTP mode the communication (for these hosts) does not run over a proxy. Several hosts can be transferred as a semicolon-separated list. If one (or several) host names are specified, all hosts which begin with the given host name may bypass the proxy (see e.g. myServer_1, myServer_2,...).
  Example: `proxybypass=myServer;localhost`
- `lookAndFeel`: Optional parameter for switching over the new default look & feel (see Chapter 1.3 page 19).
- `inlinebrowser.httpproxy` (from FirstSpirit version 4.2R2): Optional parameter for configuring inline browsers (for the Integrated preview) for the

communication via a HTTP proxy (analog to the configuration of the proxy of the FirstSpirit JavaClient). The name of the server or the IP address of the proxy and the port must be given for this. If the HTTP proxy is configured by using this parameter, it removes the local configuration settings of the browser temporarily, but it will not be saved.
Example: `inlinebrowser.httpproxy=myServer:8888`

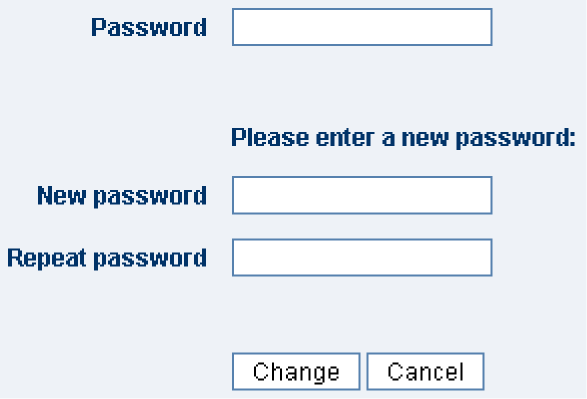
- `CLIENT_HOME_DIR` (from FirstSpirit version 4.2R2): Optional parameter for defining a directory in the file system in which client applications are to be stored (see Chapter 4.3.1.8 page 46 and Chapter 4.9.2 page 153).

Click on **Save** to save the changed connection settings for the currently logged-in user. Check the box ☑ Use settings to activate the settings. The following information is then displayed on the start page:

Connection settings
ARE ENABLED

6.3.5.2 Change password

Users can change their password for the FirstSpirit Server login by clicking on this entry.



Password

Please enter a new password:

New password

Repeat password

Figure 6-5: Change password

Password: The current password has to be re-entered in this field.

New password: The new password is entered in this field.

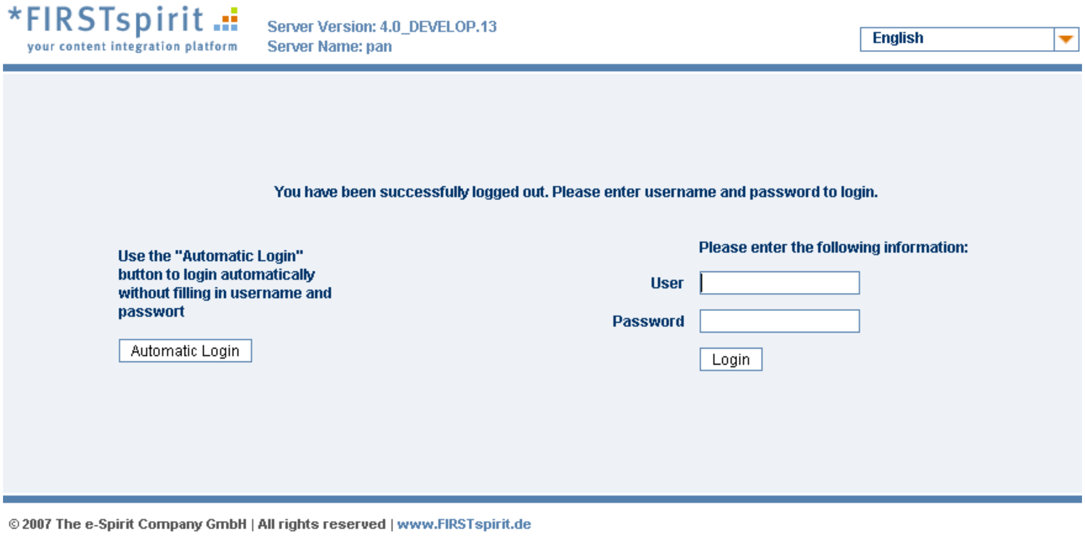
Repeat password: The new password is repeated in this field to exclude possible typing errors while changing the password.

The logged-in user's new password is accepted after clicking on Change.

! *This entry is only available to users who have been manually created on the server. Not for external users who have been created in via an automatic SSO login.*

6.3.5.3 Change user

A different user can be logged in at the server by clicking on this entry. The login page (see section 6.2 page 168 ff), via which automatic login is now possible, opens again.



*FIRSTspirit
your content integration platform

Server Version: 4.0_DEVELOP.13
Server Name: pan

English

You have been successfully logged out. Please enter username and password to login.

Use the "Automatic Login" button to login automatically without filling in username and passwort

Automatic Login

Please enter the following information:

User

Password

Login

© 2007 The e-Spirit Company GmbH | All rights reserved | www.FIRSTspirit.de

Figure 6-6: Java Web Start – Change user

If the server has an SSO-capable login module, it is possibile to automatically login at the server under the Windows login on the right-hand side of the screen.

Automatic Login Login under the Windows login takes place by clicking on this button.

6.3.5.4 Logout

The logged-in user can log out from the server by clicking on this entry. The login page (see section 6.2 page 168 ff).

6.4 Application start

6.4.1 JavaClient

A web browser with “Java Web Start”[20] is required for starting the JavaClient. Software updates of the FirstSpirit products are automatically transferred to the client systems via Java Web Start during start-up[21]. This is achieved by, amongst other things, configuring the required permission configurations (e.g. file creation permissions on the system or the user layer.

Just click on the respective entry on the FirstSpirit start page to start the client (see section 6.3 page 170).

During JavaClient start-up a project selection dialogue with a list of available projects for the logged-in user opens after establishing connection. After installation, only the example project is available at first.

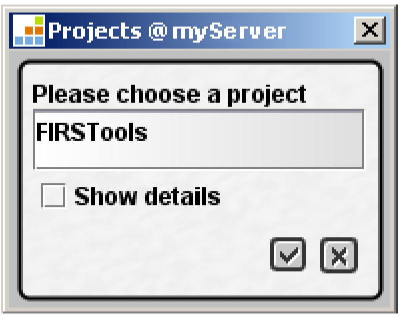


Figure 6-7: Choose project

Double click on the entry or click on the ☑ icon to load the selected project.

It is now possible to start familiarising yourself with the FirstSpirit-Client.

[20] Further information: http://java.sun.com/products/javawebstart/

[21] Method: http://de.wikipedia.org/wiki/Java_Web_Start

6.4.2 WebClient

The WebClient does not require a Java environment and can be called directly via the web browser. Just click on the respective entry on the FirstSpirit start page to start the client (see section 6.3 page 170).

! *The link is only displayed if the browser being used is supported (e.g. Internet Explorer Version 6 or later) (see the "Technical Datasheet" for more information).*

During WebClient start-up a project selection dialogue with a list of available projects for the logged-in user opens after establishing connection. Only the example project is available at first.

6.4.3 Server Monitoring

The WebClient does not require a Java environment and can be called directly via the web browser. Just click on the respective entry in the administrator area on the FirstSpirit start page to start the client (see section 6.3 page 170).

! *The link is only displayed if the browser being used is supported (e.g. Internet Explorer Version 6 or later) (see the "Technical Datasheet" for more information).*

6.4.4 Server and Project Configuration

A web browser with "Java Web Start"[22] is required for starting the JavaClient. Software updates of the FirstSpirit products are automatically transferred to the client systems via Java Web Start during start-up[23]. This can be achieved by, amongst other things, carrying out the required permission configurations (e.g. file creation permissions) on the system or user layer.

[22] Further information: http://java.sun.com/products/javawebstart/

[23] Method: http://de.wikipedia.org/wiki/Java_Web_Start

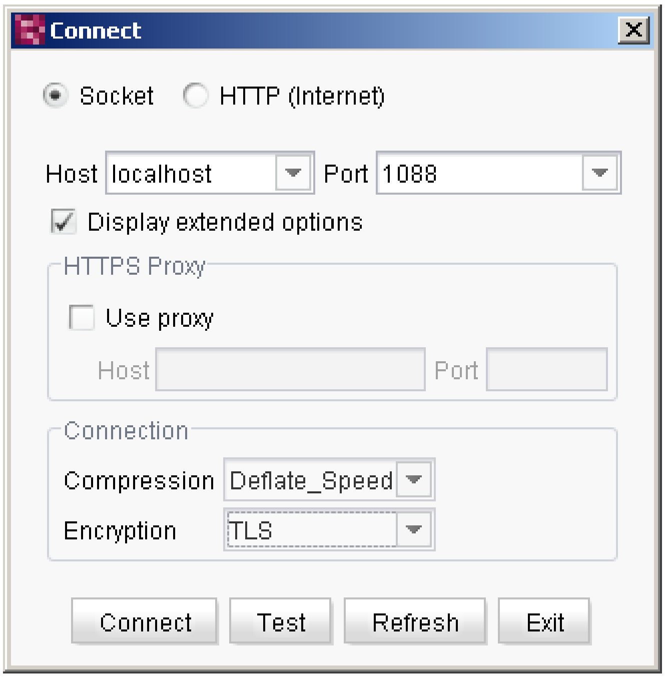
Just click on the respective entry in the administrator area on the FirstSpirit start page to start the client (see section 6.3 page 170).

6.5 Start FirstSpirit client as JAVA application

! *This function is released for FirstSpirit Version 4.1 and higher only. Screenshots are therefore displayed in the new "LightGray" look & feel. The display can differ slightly in the "Classic" look & feel.*

If the FirstSpirit server is not accessed via the internet, but instead, for example, by means of a command line, parameters for communication between JavaClient and the FirstSpirit server can be configured in the following connection dialog, similar to the connection settings in Chapter 6.3.5.1 from page 173.

6.5.1 Socket mode



Connect

● Socket ○ HTTP (Internet)

Host: localhost Port: 1088

☑ Display extended options

HTTPS Proxy

□ Use proxy

Host: Port:

Connection

Compression: Deflate_Speed

Encryption: TLS

Figure 6-8: Connection dialog for the JavaClient – Socket mode

Host: Server name or IP address of the FirstSpirit server with which the Client is to connect with the web start.

Port: Port number of the FirstSpirit server.

The following parameters can be set if the "Display advanced options" checkbox is selected:

Use proxy: If this checkbox is selected, the "Host" and "Port" fields below it can be used to define a proxy server, which is to be used for communication between JavaClient and the FirstSpirit server.

Compression: Compression for communication between FirstSpirit clients and the server for the currently logged in user:

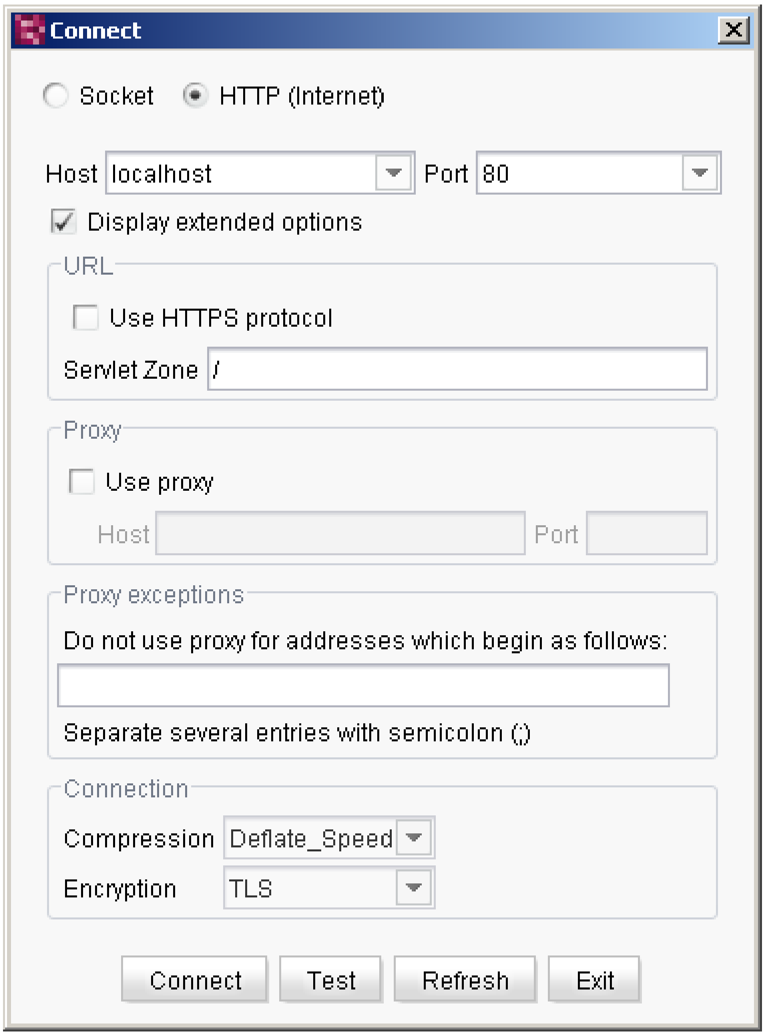
- **None**: No compression on transferring data between the client and server.
- **Deflate**: Use the deflate algorithm with standard compression for the transfer of data between client and server.
- **Deflate_Speed**: Use the deflate algorithm with fastest compression for the transfer of data between client and server.
- **Deflate_Best**: Use the deflate algorithm with the best compression for the transfer of data between client and server.

Encryption: Encryption for communication between FirstSpirit clients and the server for the currently logged in user:

- **None**: No encryption for the transfer of data between client and server.
- **TLS**: Use of the TLS protocol for the transfer of data between the client and server.
- **DH_Arc4**: Use of the DH ARC4 encryption algorithm for the transfer of data between the client and server.

A description of the settings is given in Chapter 6.3.5.1 page 173.

6.5.2 HTTP (Internet) mode



Connect

○ Socket ● HTTP (Internet)

Host localhost Port 80

☑ Display extended options

URL

□ Use HTTPS protocol

Servlet Zone /

Proxy

□ Use proxy

Host Port

Proxy exceptions

Do not use proxy for addresses which begin as follows:

Separate several entries with semicolon (;)

Connection

Compression Deflate_Speed

Encryption TLS

Figure 6-9: Connection dialog for the JavaClient – HTTP mode

A different host name and port number to those used for socket mode can be given for HTTP mode. The following additional parameters can also be configured:

Use HTTPS protocol: If this checkbox is selected, communication between the client and server is encrypted over the HTTPS protocol.

Servlet zone: Path to the servlet directory. The path must always begin with "/".

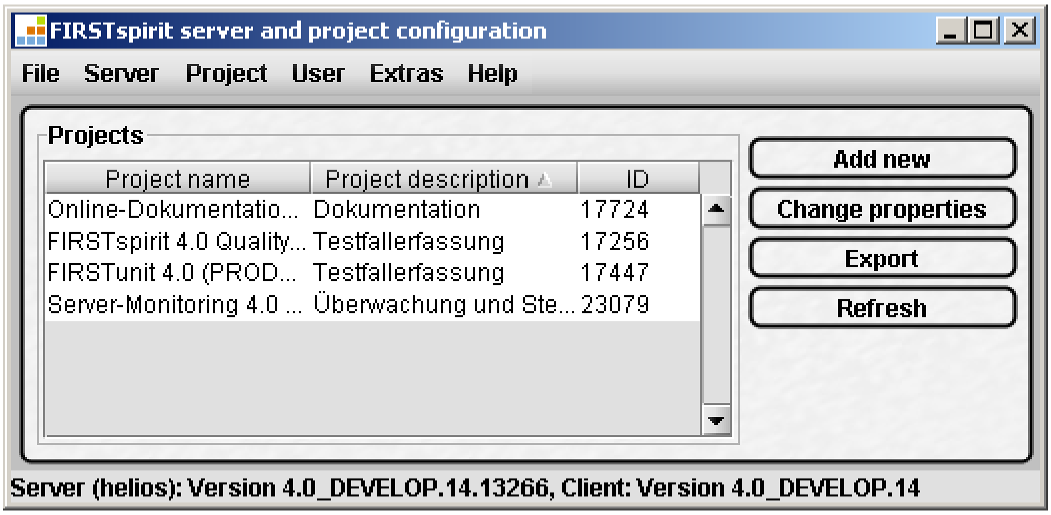
Do not use a proxy for addresses which begin as follows: Details of the domain which is to be opened directly, i.e. not over a proxy connection, e.g. addresses in your own company network. Several addresses must be separated by semicolons.

In both cases (socket and HTTP), the "Test" button can be used to test the connection between JavaClient and the FirstSpirit server using the settings made. If a connection cannot be established (i.e. the test fails), the configuration must be changed. If the test was successful, the connection can then be established using the "Connect" button.

A description of the settings is given in Chapter 6.3.5.1 *page* 173.

7 FirstSpirit Server and Project Configuration

The FirstSpirit Server and Project Configuration is a Java application with a convenient, swing-based user interface which supports the FirstSpirit administrator for general, administrative FirstSpirit tasks. For example, the user interface can be used to create and configure new FirstSpirit projects. Besides the general tasks, it provides extensive functions. The Server and Project Configuration can, e.g., be used to define users or integrate existing identity management systems, such as LDAP or Active Directory. Analogue to the JavaClient, the Server and Project Configuration is started and updated via Java Web Start.



Project name	Project description	ID
Online-Dokumentatio...	Dokumentation	17724
FIRSTspirit 4.0 Quality...	Testfallerfassung	17256
FIRSTunit 4.0 (PROD...	Testfallerfassung	17447
Server-Monitoring 4.0 ...	Überwachung und Ste...	23079

Server (helios): Version 4.0_DEVELOP.14.13266, Client: Version 4.0_DEVELOP.14

Figure 7-1: Server and Project Configuration

7.1 Server and project administrators

Task sharing in FirstSpirit also refers to the administrators, who are distinguished according to their permissions. Each project which is created on the server requires one or more project administrators.

Project administrators can:

- Change the properties of their project
- Export their project
- Clean up their project

In contrast, the **Server administrator** can:

- Create / export / delete new projects
- Create users
- Change the properties of all the projects
- Define project administrators
- Install und uninstall editor and function components
- Execute special server operations

7.2 Menu bar elements

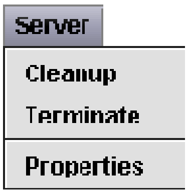
The individual entries in the Server and Project Configuration menu bar are described below.

7.2.1 File

7.2.1.1 Terminate

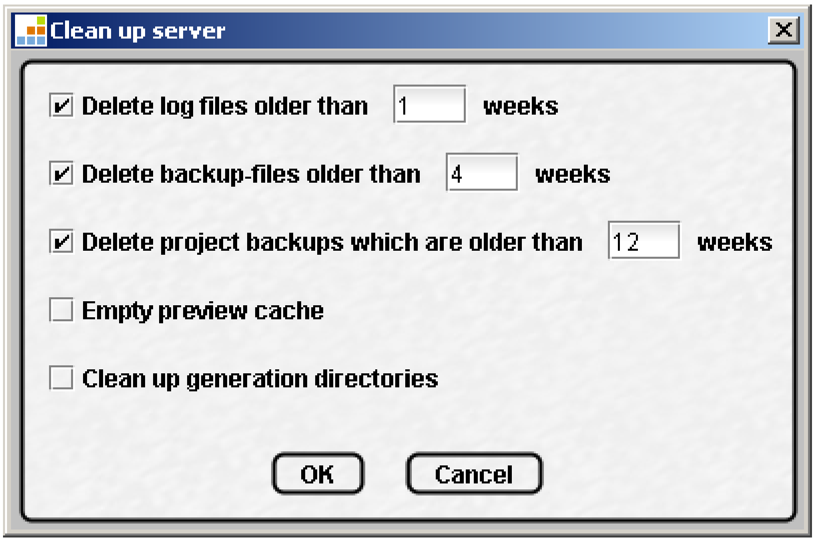
This function terminates the Server and Project Configuration.

7.2.2 Server



7.2.2.1 Clean up

This function cleans up, e.g., server backup and log files. The following clean up actions can be executed via the dialogue “Clean up server”:



Clean up server

☑ Delete log files older than 1 weeks

☑ Delete backup-files older than 4 weeks

☑ Delete project backups which are older than 12 weeks

☐ Empty preview cache

☐ Clean up generation directories

Figure 7-2: Clean up server

Delete log files: Deletes all files which are older than the specified number of weeks from the server log directory.

Delete backup files: Deletes the backup files for pages and media which are older than the specified number of weeks. The deleting of backup files refers to all server project directories.

Delete project backups: Deletes all (automatically created) project backups which are older than the specified number of weeks. This affects all the compressed export files located in the subdirectory “backup” directly underneath the FirstSpirit Server root directory. These are created in the schedule entry planning (see section 7.5.9.5 page 348). (It is also possible to transfer the directory to another hard drive (see section 4.3.1.8 page 46).)

! *Projects are not deleted in the project export directory via the function “Delete project backups”.*

Empty preview cache: Removes all the generated preview pages from the preview cache. The generated pages are stored in the directory “preview_cache” underneath the FirstSpirit Server directory “web/fs4root/” after requesting a preview. The directory can be changed via the configuration file fs-server.conf (see section 4.3.1.9 page 48).

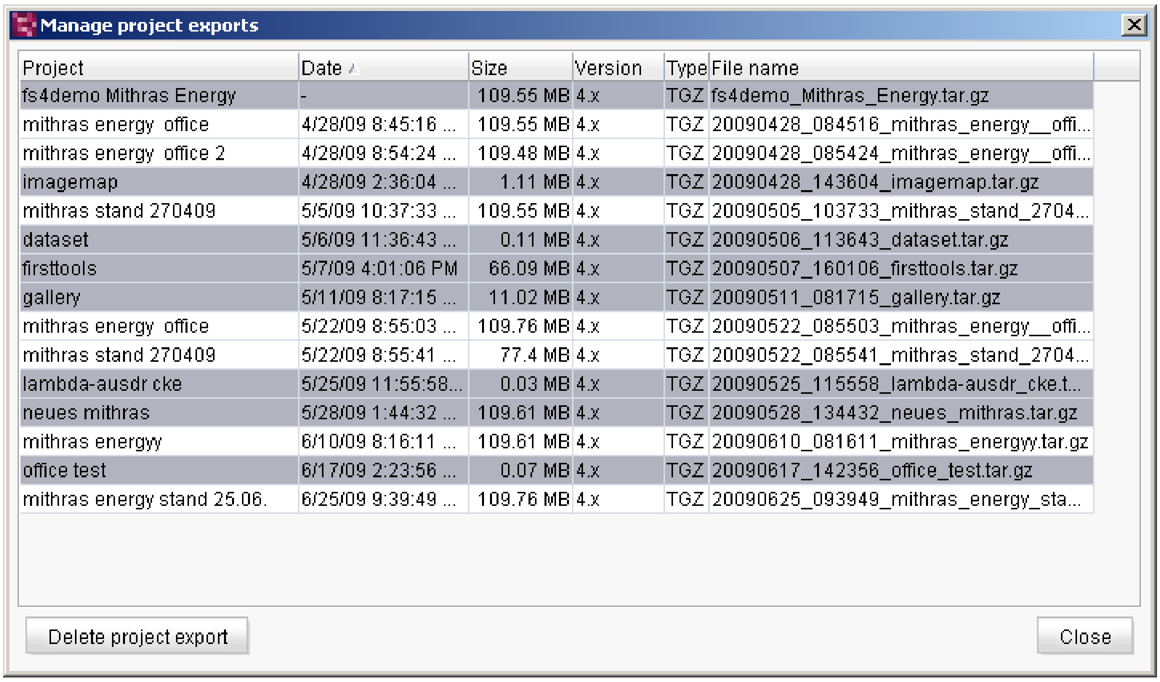
Clean up generation directories: This function deletes data which is no longer required by deleted or deactivated projects. For example, directories and their content created during generations and deployments:

- Clean up of the directories “project_projectID” in the FirstSpirit Server subdirectory “web/fs4staging”.
- Clean up of the directories “project_projectID_partial_deployment” in the FirstSpirit Server subdirectory “web/fs4staging”.
- Clean up of the directories in which the project previews have been stored (directory “project_projectID” in the directory “preview_cache” of the subdirectory “web/fs4staging”). The preview directory as well as the corresponding paths can be configured via the configuration file fs-server.conf (see section 4.3.1.9 page 48).

Renew search index of all projects (from V4.1): From FirstSpirit Version 4.1, a search (full text search of the Lucene search index) can be made using the JavaClient's selection dialogs. For this, the search index must be calculated for the project. If this option is selected, the indexing for the search is carried out for all projects on the server. This indexing only takes into account the current state of the elements in the project. Historic data (e.g. changes to a project within a specific period) is not calculated. Depending on the number and size of the projects, recalculation of the search index can take a very long time. (The individual projects can be indexed using the BeanShell console within JavaClient.)

Delete unused project directories (from V4.2): In some cases projects can not be deleted completely, e.g. the operating system still has access to files in the project’s directories. Such projects can no more be accessed via the application for the Server and Project Configuration. The assiciated project directories can be deleted definitely by using this option from FirstSpirit version 4.2.

Manage project exports (from V4.1) Click the button to open the "Manage Project Exports" dialog:



Project	Date	Size	Version	Type	File name
fs4demo Mithras Energy	-	109.55 MB	4.x	TGZ	fs4demo_Mithras_Energy.tar.gz
mithras energy office	4/28/09 8:45:16 ...	109.55 MB	4.x	TGZ	20090428_084516_mithras_energy__offi...
mithras energy office 2	4/28/09 8:54:24 ...	109.48 MB	4.x	TGZ	20090428_085424_mithras_energy__offi...
imagemap	4/28/09 2:36:04 ...	1.11 MB	4.x	TGZ	20090428_143604_imagemap.tar.gz
mithras stand 270409	5/5/09 10:37:33 ...	109.55 MB	4.x	TGZ	20090505_103733_mithras_stand_2704...
dataset	5/6/09 11:36:43 ...	0.11 MB	4.x	TGZ	20090506_113643_dataset.tar.gz
firsttools	5/7/09 4:01:06 PM	66.09 MB	4.x	TGZ	20090507_160106_firsttools.tar.gz
gallery	5/11/09 8:17:15 ...	11.02 MB	4.x	TGZ	20090511_081715_gallery.tar.gz
mithras energy office	5/22/09 8:55:03 ...	109.76 MB	4.x	TGZ	20090522_085503_mithras_energy__offi...
mithras stand 270409	5/22/09 8:55:41 ...	77.4 MB	4.x	TGZ	20090522_085541_mithras_stand_2704...
lambda-ausdr cke	5/25/09 11:55:58...	0.03 MB	4.x	TGZ	20090525_115558_lambda-ausdr_cke.t...
neues mithras	5/28/09 1:44:32 ...	109.61 MB	4.x	TGZ	20090528_134432_neues_mithras.tar.gz
mithras energyy	6/10/09 8:16:11 ...	109.61 MB	4.x	TGZ	20090610_081611_mithras_energyy.tar.gz
office test	6/17/09 2:23:56 ...	0.07 MB	4.x	TGZ	20090617_142356_office_test.tar.gz
mithras energy stand 25.06.	6/25/09 9:39:49 ...	109.76 MB	4.x	TGZ	20090625_093949_mithras_energy_sta...

Figure 7-3: Manage project exports

The dialog is used to list all existing project exports in tabular form. These can be selected in the overview and if necessary deleted using the "Delete project exports" button.

7.2.2.2 Terminate

This function shuts down the server.

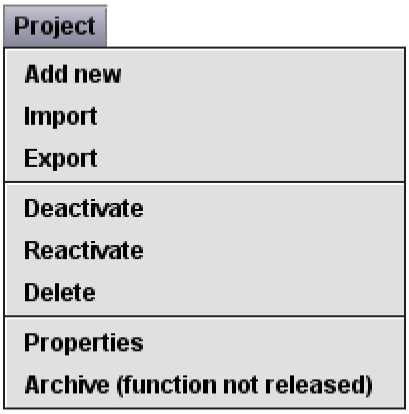
A restart is NOT possible via the Server and Project Configuration!

7.2.2.3 Properties

Call this function to open a window in which a number of server properties can be set.

See chapter 7.3 page 210 ff for a detailed description of these server properties.

7.2.3 Project



- Add new project (see section 7.2.3.1)
- Import project (see section 7.2.3.2)
- Export project (see section 7.2.3.3)
- Deactivate project (see section 7.2.3.4)
- Reactivate project (see section 7.2.3.5)
- Delete project (see section 7.2.3.6)
- Project properties (see section 7.2.3.7)
- Project archive (see section 7.2.3.8)

7.2.3.1 Add new

This function creates a new project on the server.

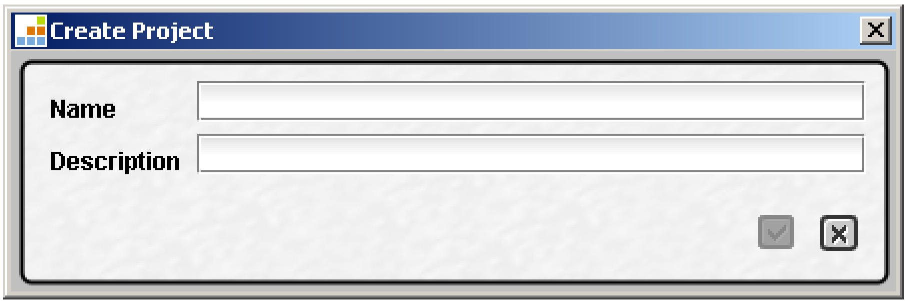
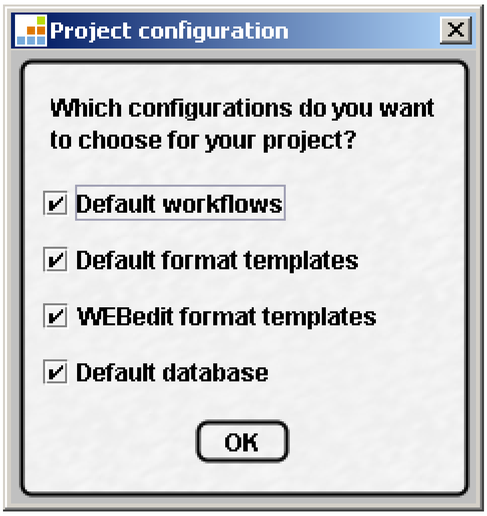


Figure 7-4: Project – Add new

Name: The name specified here is listed in the root of the FirstSpirit-Client tree. The specified project name has to be unique.

Description: The description specified here is displayed in the project selection list after login.

After entering the project name and project description, the Project Configuration is set in the next dialogue.



Project configuration

Which configurations do you want to choose for your project?

☑ Default workflows

☑ Default format templates

☑ WEBedit format templates

☑ Default database

Figure 7-5: Project configuration

Check the boxes to accept the respective configurations for the project.

Default format templates: Check this box to import the default format templates provided by FirstSpirit into the format templates node (in the Template-Store of the project). Editors are thus able to format a text, e.g. in the input component DOM editor, as bold or italic.

Default workflows: Check this box to import the default workflows provided by FirstSpirit into the workflows node (in the Template-Store of the project). There are two integrated workflows in FirstSpirit:

1. “Task”: Workflow for general editing of tasks in the project.
2. “Request release”: Workflow for releasing objects which have been newly created or changed in the project.

WebEdit format templates: Check this box to import the WebEdit format templates provided by FirstSpirit into the format templates node (in the Template-Store of the project). These format templates are not absolutely necessary when using WebEdit in projects. Nonetheless, the utilisation of format templates is usually recommended, e.g. if a project works with frames or if certain WebEdit elements, e.g. “Quick Edit”, are to be used.

Default database: Check this box to activate the default database (Derby) provided with FirstSpirit for the project. The default layer can be used for a database schema in the FirstSpirit-Client. This automatically sets write access to the database for this project (see section 7.4.12 page 281).

The Derby-DBMS integrated in FirstSpirit is not suitable for productive operation and should therefore only be used for tests.

Confirming the configuration via the “OK” button ensures that the new project is immediately accepted in the project list and can be further edited via the “Change properties” button (see section 7.3.16 page 243 ff).

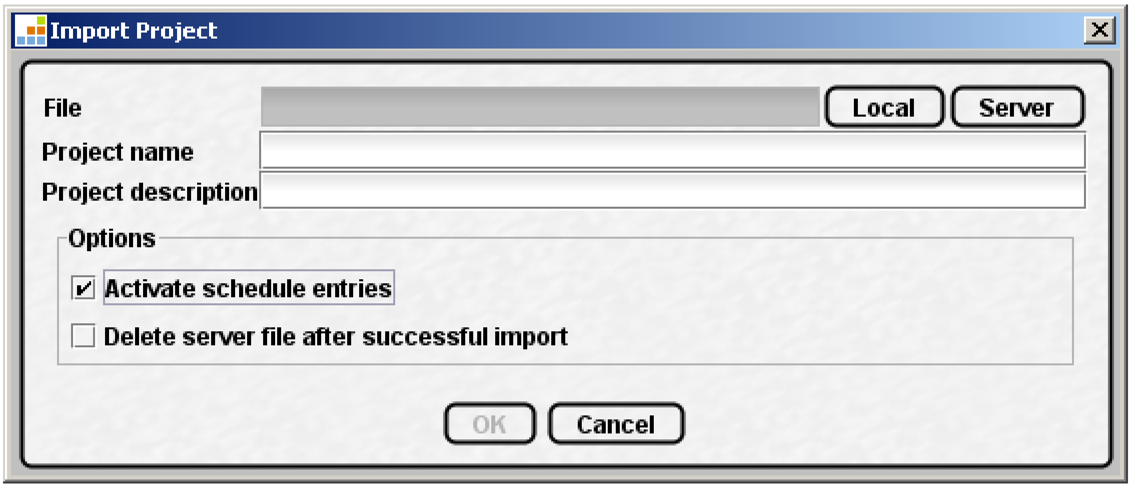
Add new fulfils the same function.

This menu item is only available to server administrators!

7.2.3.2 Import

Transfer exported projects back to the server via the project import.

This menu item is only available to server administrators!



Import Project

File [Local] [Server]

Project name

Project description

Options

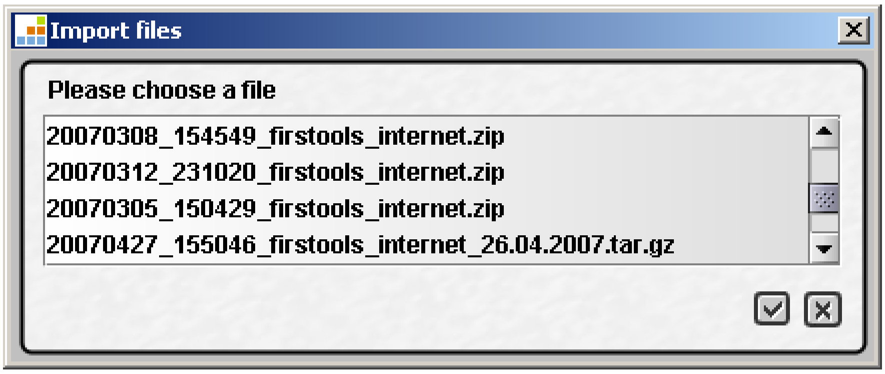
☑ Activate schedule entries

□ Delete server file after successful import

[OK] [Cancel]

Figure 7-6: Project – Import

File: Select a compressed project export file via the “Local” and “Server” button. Click on **Local** to browse the file system of the computer.
Click on **Server** to display a list of the compressed files from the export directory of the server. Subsequently select the desired project from this list (see Figure 7-7):



Please choose a file

20070308_154549_firstools_internet.zip
20070312_231020_firstools_internet.zip
20070305_150429_firstools_internet.zip
20070427_155046_firstools_internet_26.04.2007.tar.gz

Figure 7-7: Import project – Import file list

Project name: Enter a unique name in this field for the project which is to be imported. This name is displayed on the project root of the JavaClient.

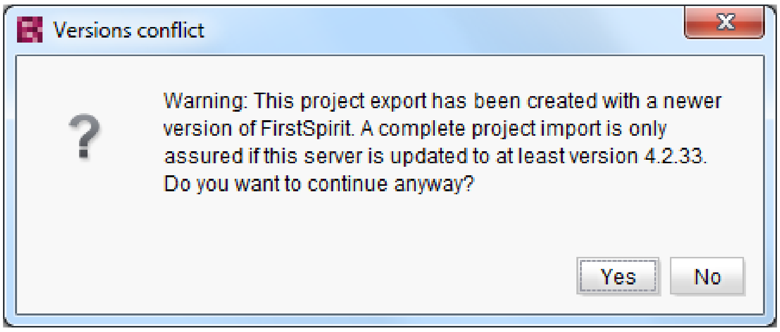
Project description: Enter a project description in this field for the project which is to be imported.

<u>Options Activate schedule entries:</u>
If this box is **checked**, all schedule entries remain in the status at the time of project export. This means that active schedule entries remain active and are executed at the set time. This option should, therefore, only be activated in full knowledge of the entries and with caution, since it could, e.g., result in unwanted deployments. If the box is **unchecked**, the schedule entries for a project remain, but are deactivated.

Delete server file after successful import: If this box is **checked**, the export file of the project is deleted on the server after project import. It is then no longer available in the import file list (see Figure 7-7).

Click on “Yes“ to start the import.

If the project which is to be imported was created on a FirstSpirit server of a higher version after an analyse of the file the following warning will be displayed:



Warning: This project export has been created with a newer version of FirstSpirit. A complete project import is only assured if this server is updated to at least version 4.2.33. Do you want to continue anyway?

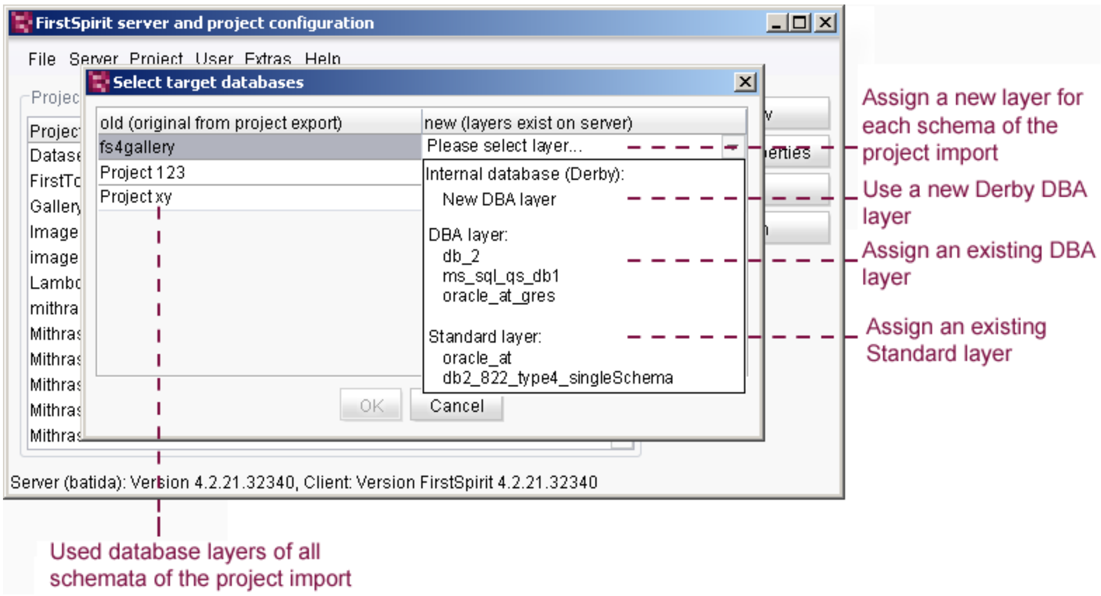
Figure 7-8: Version conflict during import

Click “No“ to cancel the importing process, click “Yes“ to continue the import.

> ! *The import of projects which have been created by a FirstSpirit server with a higher version (e.g. export from a FirstSpirit server of the version 4.2, import to a FirstSpirit server of the version 4.1) is not a warranted prodct property! Technically such an export/import can be executed, but can lead to unforeseeable problems, especially in the case of an export from 4.2 to 4.1, because of the numerous newly implemented input components, methods and incompatible changes in the software behaviour.*

If database contents are used in the project, these contents have to be mapped onto a new database layer prior to project import. On importing a project it is now possible:

- Up to FirstSpirit Version 4.2: To assign a new database layer or one which already exists on the server to all a project's schemata
- Since FirstSpirit Version 4.2: To assign a new database layer or one which already exists on the server to each individual schema of a project (see Figure 7-9):


FirstSpirit server and project configuration
File Server Project User Extras Help
Select target databases
OK
Cancel
Internal database (Derby):
New DBA layer
DBA layer:
db_2
ms_sql_qs_db1
oracle_at_gres
Standard layer:
oracle_at
db2_822_type4_singleSchema

old (original from project export)	new (layers exist on server)
fs4gallery	Please select layer...
Project 123	
Project xy	

Server (batida): Version 4.2.21.32340, Client: Version FirstSpirit 4.2.21.32340

Assign a new layer for each schema of the project import

Use a new Derby DBA layer

Assign an existing DBA layer

Assign an existing Standard layer

Used database layers of all schemata of the project import

Figure 7-9: Selection of the destination for databases (Version 4.2)

Up until FirstSpirit Version 4.2, when a project was imported it was only possible to assign one database layer for all its schemata.

Since FirstSpirit Version 4.2, each schema of the export file is now displayed in the "Selection of the destination for databases" dialog (left-hand area). The new assignment of a layer for the imported project can be selected from the right-hand area. For each individual schema, either:

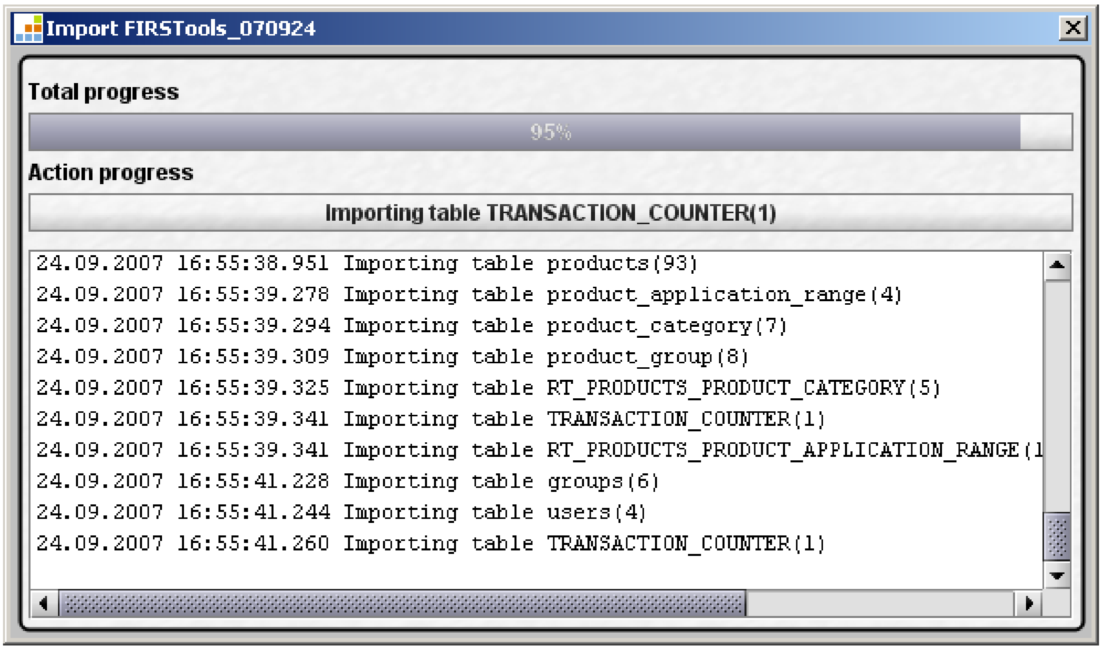
- a new Derby DBA layer,
- a default layer or
- a DBA layer can be selected.

! *The Derby DBMS contained in FirstSpirit is not suitable for productive use and should therefore be used for tests only.*

The choice of layer types decides whether FirstSpirit users can independently create new schemata in the project (possible for DBA layers) or not (is prevented for default layers)(for a further description, see Chapter 4.8.4.2 page 140).

! *Before FirstSpirit Version 4.2, the following terms were used:*
Multi-Project Layer (sic): since FirstSpirit Version 4.2 corresponds to the term "Default Layer" (standard layer)
Single-Project Layer (sic): since FirstSpirit Version 4.2 corresponds to the term "DBA Layer"

Depending on the project size, the import process can take several minutes. The total progress and the individual actions are displayed in the import dialogue.



Importing table TRANSACTION_COUNTER(1)

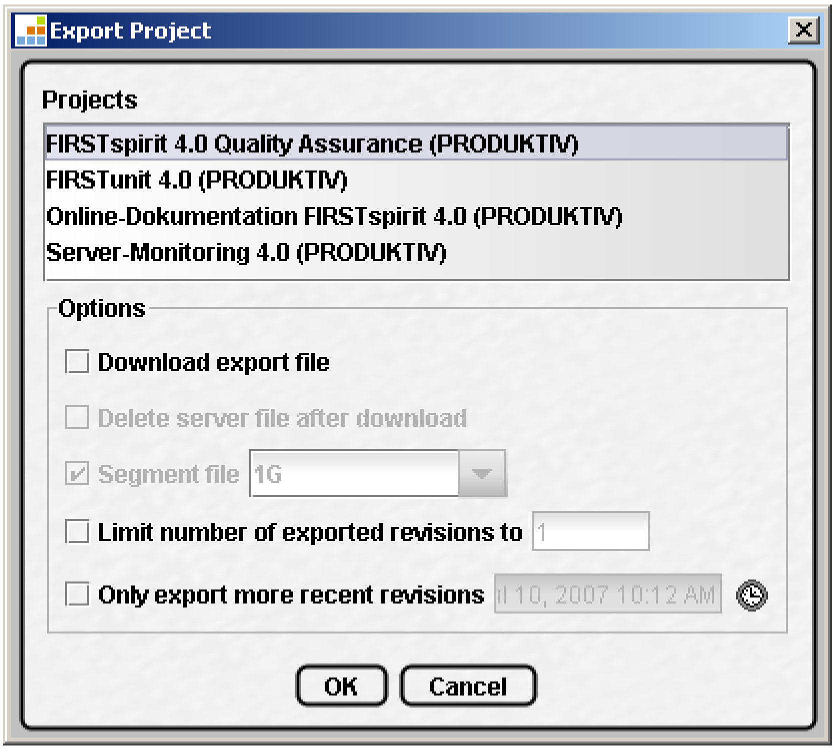
```
24.09.2007 16:55:38.951 Importing table products(93)
24.09.2007 16:55:39.278 Importing table product_application_range(4)
24.09.2007 16:55:39.294 Importing table product_category(7)
24.09.2007 16:55:39.309 Importing table product_group(8)
24.09.2007 16:55:39.325 Importing table RT_PRODUCTS_PRODUCT_CATEGORY(5)
24.09.2007 16:55:39.341 Importing table TRANSACTION_COUNTER(1)
24.09.2007 16:55:39.341 Importing table RT_PRODUCTS_PRODUCT_APPLICATION_RANGE(1
24.09.2007 16:55:41.228 Importing table groups(6)
24.09.2007 16:55:41.244 Importing table users(4)
24.09.2007 16:55:41.260 Importing table TRANSACTION_COUNTER(1)
```

Figure 7-10: Import project – Progress display

The project import won't be cancelled, if errors occur during the import (e.g. because of a failed database sync). An error dialogue will be displayed after the import operation. The corresponding exceptions can be found in the import log. The project will initially be deactivated after import.

7.2.3.3 Export

Use this function to create an export file for an individual project.



Export Project

Projects

- FIRSTspirit 4.0 Quality Assurance (PRODUKTIV)
- FIRSTunit 4.0 (PRODUKTIV)
- Online-Dokumentation FIRSTspirit 4.0 (PRODUKTIV)
- Server-Monitoring 4.0 (PRODUKTIV)

Options

□ Download export file

□ Delete server file after download

☑ Segment file 1G

□ Limit number of exported revisions to 1

□ Only export more recent revisions il 10, 2007 10:12 AM

Figure 7-11: Project - Export

All current server projects are listed at the top of the window. Select the project to be exported from these projects.

Download export file: If this option has been activated, the export file is also stored in the file system of the local computer.

After confirming the settings via **OK**, a window opens in which the target location for the export file can be selected.

Delete server file after download: This option can only be activated if the export file is downloaded locally. If this option has been activated, the export file on the server is deleted again immediately after download to the local computer.

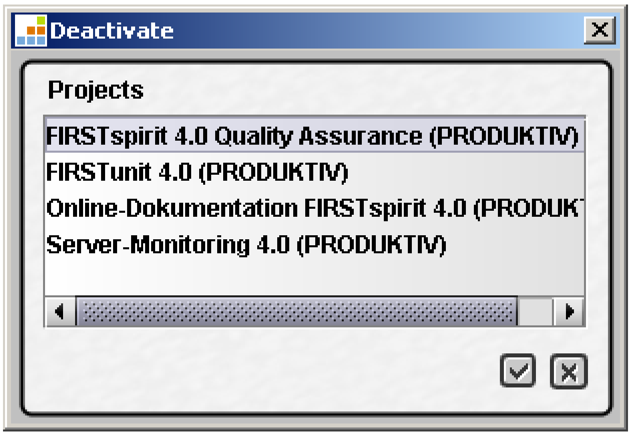
Segment file: Use this option to limit the maximum size of the export file. A corresponding number of export files is then created. (This option is required for extremely large projects to ensure they can still be copied onto external data carriers.)

Limit number of exported revisions to: It is possible to specify the number of revisions created during the export process. If the number of revisions is limited to 1, only the current project status is exported (also see section 9.13 page 436).

Only export more recent revisions: It is possible to set the maximum age of the revisions to be exported. If the current date has been set, only the current project state is exported.

7.2.3.4 Deactivate

If a project is to be temporarily removed from the project list (without deleting it completely), it can be deactivated here.



FIRSTspirit 4.0 Quality Assurance (PRODUKTIV)
FIRSTunit 4.0 (PRODUKTIV)
Online-Dokumentation FIRSTspirit 4.0 (PRODUK
Server-Monitoring 4.0 (PRODUKTIV)

Figure 7-12: Project - Deactivate

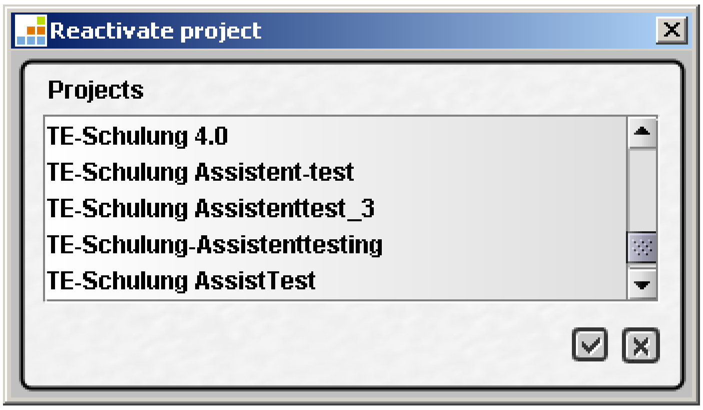
This is achieved by selecting the project to be deactivated from the displayed list and clicking on ☑ to confirm selection. A multiple selection is possible by simultaneously pressing the SHIFT or CTRL key. If there are still open sessions on a project, a warning will be displayed prior to deactivation. However, the project can still be deactivated.

A project must be deactivated in order to delete a project!

This menu item is only available to server administrators!

7.2.3.5 Reactivate

Use this function to re-release deactivated projects for editing. A selection window displays currently deactivated projects.



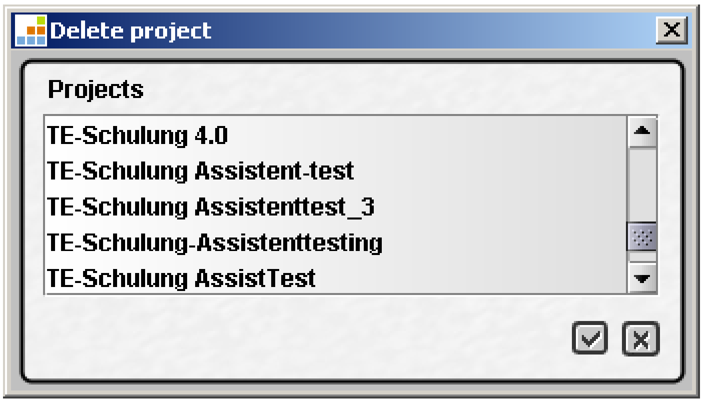
- TE-Schulung 4.0
- TE-Schulung Assistent-test
- TE-Schulung Assistenttest_3
- TE-Schulung-Assistenttesting
- TE-Schulung AssistTest

Figure 7-13: Project - Reactivate

This menu item is only available to server administrators!

7.2.3.6 Delete

Use this function to irrevocably delete projects from the server. For security reasons, only projects which have been deactivated can be deleted. These projects are displayed in a selection list.



- TE-Schulung 4.0
- TE-Schulung Assistent-test
- TE-Schulung Assistenttest_3
- TE-Schulung-Assistenttesting
- TE-Schulung AssistTest

Figure 7-14: Project - Delete

It is also possible to select several projects simultaneously from the list for deleting.

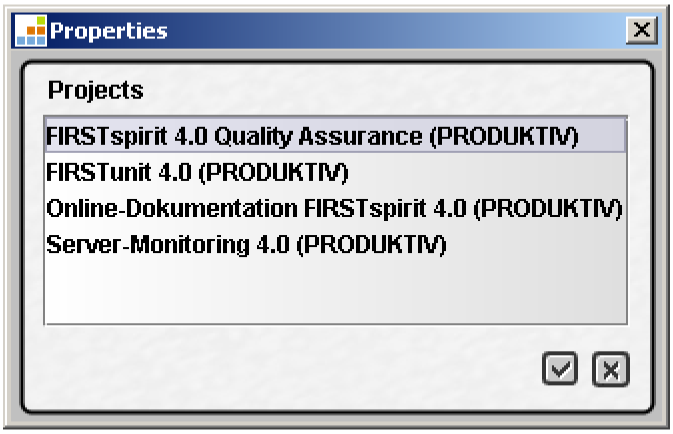
A confirmation prompt will appear prior to deleting a project.

Deleting a project means that all project data is completely deleted from the hard drive of the server. This process cannot be undone.

This menu item is only available to server administrators!

7.2.3.7 Properties

Call this function to display a window in which a project has to be selected first.



FIRSTspirit 4.0 Quality Assurance (PRODUKTIV)
FIRSTunit 4.0 (PRODUKTIV)
Online-Dokumentation FIRSTspirit 4.0 (PRODUKTIV)
Server-Monitoring 4.0 (PRODUKTIV)

Figure 7-15: Project - Properties

The project settings open after selecting a project. See the project settings for detailed information (section 7.3.16 page 243 ff).

7.2.3.8 Archive (from V4.1)

This function is released for FirstSpirit Version 4.1 and higher only. Screenshots are therefore displayed in the new "LightGray" look & feel. The display can differ slightly in the "Classic" look & feel.

For information about the use of this function in comparison with the use of the module “FirstSpirit EnterpriseBackup“ see also FirstSpirit Release Notes 4.2, Chapter “Long-term archiving and backup in FirstSpirit“.

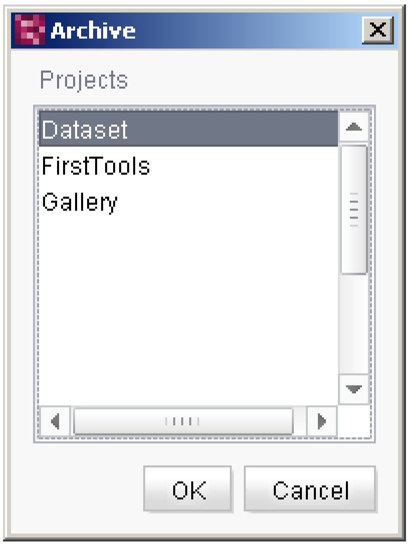
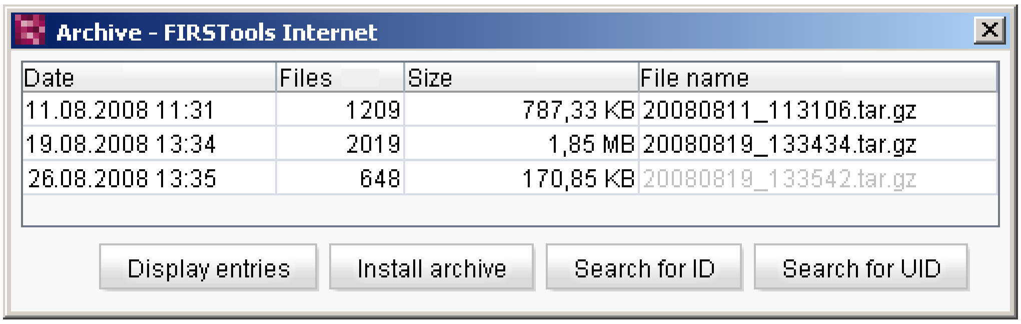


Figure 7-16: Project – Archive

The "Archive" function can be used to display data from archive files created during an archiving schedule (Chapter 7.5.9.1 page 328):



Date	Files	Size	File name
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

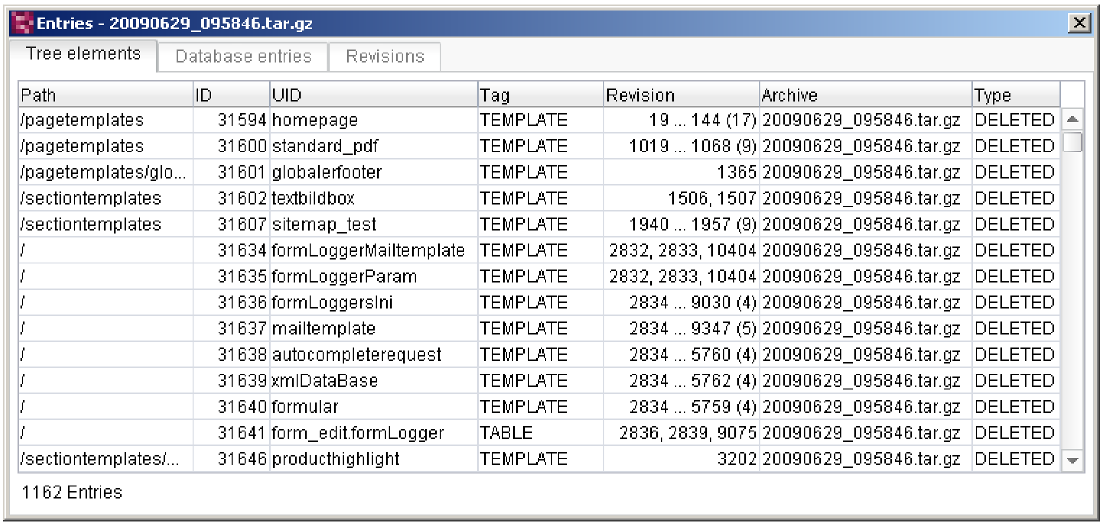
Figure 7-17: Archive list

After selecting a project from Figure 7-16, all the available archive files are displayed with the date on which they were created, the number of files (archived elements), the file size of the archive and the file name. Deleted archive files are shown in grey.

Display entries : Use this button to display all the entries of an archive file. They are displayed on three tabs, separated by tree elements, database entries and revision information:

▪ Tree elements	(see Chapter 7.2.3.8.1 page 200)
▪ Database entries	(see Chapter 7.2.3.8.2 page 201).
▪ Revisions	(see Chapter 7.2.3.8.3 page 202)

7.2.3.8.1 Archived tree elements (from V4.1)



Path	ID	UID	Tag	Revision	Archive	Type
/pagetemplates	31594	homepage	TEMPLATE	19 ... 144 (17)	20090629_095846.tar.gz	DELETED
/pagetemplates	31600	standard_pdf	TEMPLATE	1019 ... 1068 (9)	20090629_095846.tar.gz	DELETED
/pagetemplates/glo...	31601	globalerfooter	TEMPLATE	1365	20090629_095846.tar.gz	DELETED
/sectiontemplates	31602	textbildbox	TEMPLATE	1506, 1507	20090629_095846.tar.gz	DELETED
/sectiontemplates	31607	sitemap_test	TEMPLATE	1940 ... 1957 (9)	20090629_095846.tar.gz	DELETED
/	31634	formLoggerMailtemplate	TEMPLATE	2832, 2833, 10404	20090629_095846.tar.gz	DELETED
/	31635	formLoggerParam	TEMPLATE	2832, 2833, 10404	20090629_095846.tar.gz	DELETED
/	31636	formLoggersIni	TEMPLATE	2834 ... 9030 (4)	20090629_095846.tar.gz	DELETED
/	31637	mailtemplate	TEMPLATE	2834 ... 9347 (5)	20090629_095846.tar.gz	DELETED
/	31638	autocompleterequest	TEMPLATE	2834 ... 5760 (4)	20090629_095846.tar.gz	DELETED
/	31639	xmlDataBase	TEMPLATE	2834 ... 5762 (4)	20090629_095846.tar.gz	DELETED
/	31640	formular	TEMPLATE	2834 ... 5759 (4)	20090629_095846.tar.gz	DELETED
/	31641	form_edit.formLogger	TABLE	2836, 2839, 9075	20090629_095846.tar.gz	DELETED
/sectiontemplates/...	31646	producthighlight	TEMPLATE	3202	20090629_095846.tar.gz	DELETED

1162 Entries

Figure 7-18: Archived tree elements

The following data is displayed in this list for each archived tree element from the FirstSpirit stores:

Path: Path to the element for which data was archived

ID / UID: ID or reference name of the archived element

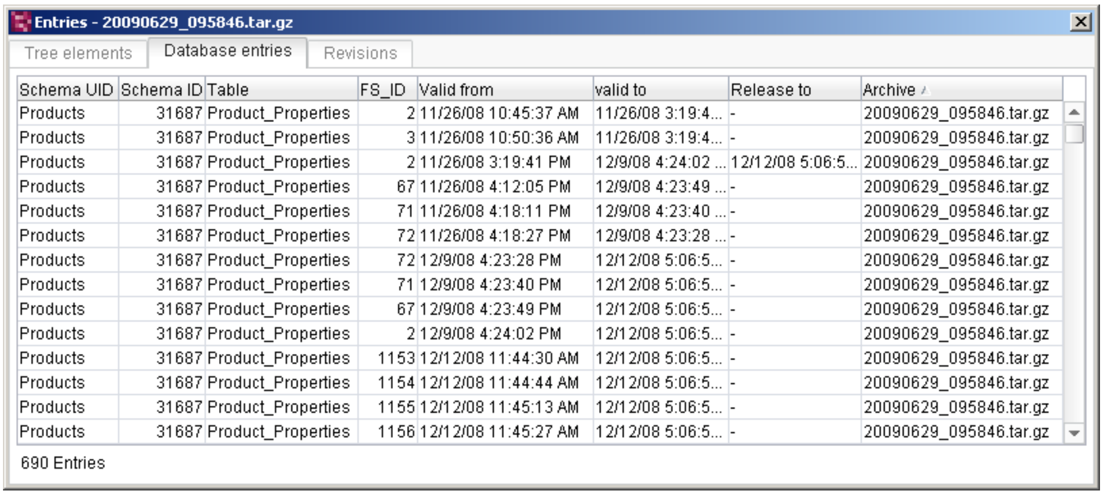
Tag: Type of archived element (e.g. medium, template, workflow)

Revision: Revision number/s of the archived element. Up to three revision numbers are completely displayed, if more than three revision numbers exist the first and last revision are displayed and the number of archived revisions is shown in brackets after this.

Archive: Name of the file in which the element was archived

Type: Shows the reason for archiving (`OLD_VERSION`, for example, means that an older and therefore no longer required state of an object was archived, `DELETED` means that a deleted object was archived).

7.2.3.8.2 Archived database entries (from V4.1)



Schema UID	Schema ID	Table	FS_ID	Valid from	valid to	Release to	Archive
Products	31687	Product_Properties	2	11/26/08 10:45:37 AM	11/26/08 3:19:4...	-	20090629_095846.tar.gz
Products	31687	Product_Properties	3	11/26/08 10:50:36 AM	11/26/08 3:19:4...	-	20090629_095846.tar.gz
Products	31687	Product_Properties	2	11/26/08 3:19:41 PM	12/9/08 4:24:02 ...	12/12/08 5:06:5...	20090629_095846.tar.gz
Products	31687	Product_Properties	67	11/26/08 4:12:05 PM	12/9/08 4:23:49 ...	-	20090629_095846.tar.gz
Products	31687	Product_Properties	71	11/26/08 4:18:11 PM	12/9/08 4:23:40 ...	-	20090629_095846.tar.gz
Products	31687	Product_Properties	72	11/26/08 4:18:27 PM	12/9/08 4:23:28 ...	-	20090629_095846.tar.gz
Products	31687	Product_Properties	72	12/9/08 4:23:28 PM	12/12/08 5:06:5...	-	20090629_095846.tar.gz
Products	31687	Product_Properties	71	12/9/08 4:23:40 PM	12/12/08 5:06:5...	-	20090629_095846.tar.gz
Products	31687	Product_Properties	67	12/9/08 4:23:49 PM	12/12/08 5:06:5...	-	20090629_095846.tar.gz
Products	31687	Product_Properties	2	12/9/08 4:24:02 PM	12/12/08 5:06:5...	-	20090629_095846.tar.gz
Products	31687	Product_Properties	1153	12/12/08 11:44:30 AM	12/12/08 5:06:5...	-	20090629_095846.tar.gz
Products	31687	Product_Properties	1154	12/12/08 11:44:44 AM	12/12/08 5:06:5...	-	20090629_095846.tar.gz
Products	31687	Product_Properties	1155	12/12/08 11:45:13 AM	12/12/08 5:06:5...	-	20090629_095846.tar.gz
Products	31687	Product_Properties	1156	12/12/08 11:45:27 AM	12/12/08 5:06:5...	-	20090629_095846.tar.gz

690 Entries

Figure 7-19: Archived information on database entries

If database entries were archived by the archiving schedule (see Chapter 7.5.9.1 page 328), they are displayed in the list with the following information:

Schema ID / UID: ID or reference name of the schema from which the archived information on the respective data set originates

Table: Name of the table from which the archived information on the respective data set originates

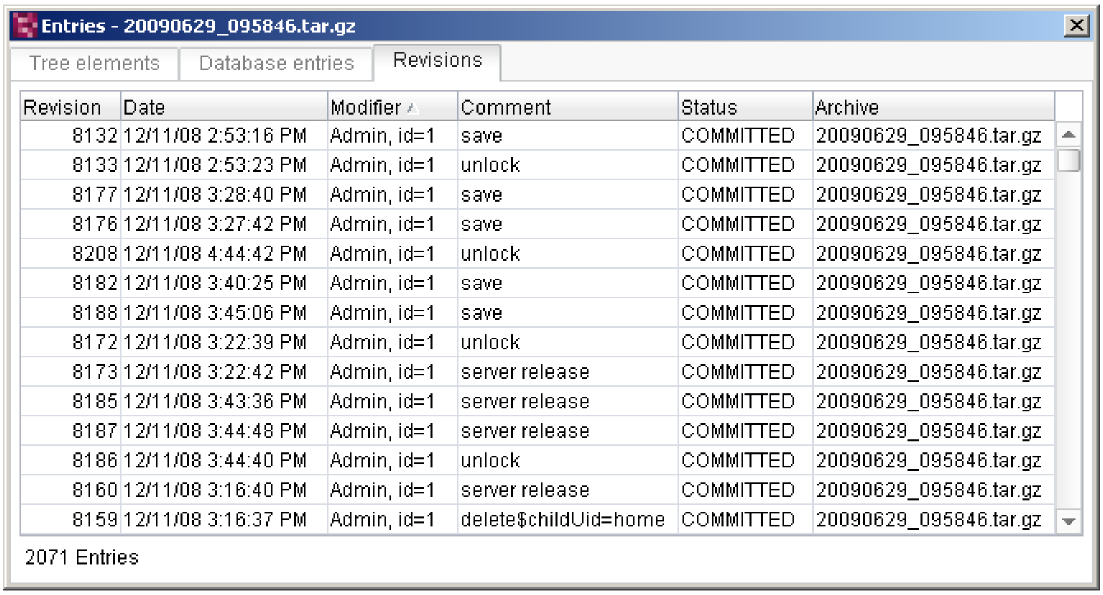
FS_ID: ID of the data set, on which information was archived. Entries with the same FS_ID indicate that changes have been made to the respective data set.

Valid from / valid to: analogous to revisions in tree elements, this data gives the period during which an unchanged version of the data set existed. If changes have been made to a data set, the different versions of the data set can be identified on the basis of the time period.

Release to: gives the date/time, up until which the data set was released

Archive: Name of the file in which the information on the database entry is archived

7.2.3.8.3 Archived revisions (from V4.1)



Revision	Date	Modifier	Comment	Status	Archive
8132	12/11/08 2:53:16 PM	Admin, id=1	save	COMMITTED	20090629_095846.tar.gz
8133	12/11/08 2:53:23 PM	Admin, id=1	unlock	COMMITTED	20090629_095846.tar.gz
8177	12/11/08 3:28:40 PM	Admin, id=1	save	COMMITTED	20090629_095846.tar.gz
8176	12/11/08 3:27:42 PM	Admin, id=1	save	COMMITTED	20090629_095846.tar.gz
8208	12/11/08 4:44:42 PM	Admin, id=1	unlock	COMMITTED	20090629_095846.tar.gz
8182	12/11/08 3:40:25 PM	Admin, id=1	save	COMMITTED	20090629_095846.tar.gz
8188	12/11/08 3:45:06 PM	Admin, id=1	save	COMMITTED	20090629_095846.tar.gz
8172	12/11/08 3:22:39 PM	Admin, id=1	unlock	COMMITTED	20090629_095846.tar.gz
8173	12/11/08 3:22:42 PM	Admin, id=1	server release	COMMITTED	20090629_095846.tar.gz
8185	12/11/08 3:43:36 PM	Admin, id=1	server release	COMMITTED	20090629_095846.tar.gz
8187	12/11/08 3:44:48 PM	Admin, id=1	server release	COMMITTED	20090629_095846.tar.gz
8186	12/11/08 3:44:40 PM	Admin, id=1	unlock	COMMITTED	20090629_095846.tar.gz
8160	12/11/08 3:16:40 PM	Admin, id=1	server release	COMMITTED	20090629_095846.tar.gz
8159	12/11/08 3:16:37 PM	Admin, id=1	delete$childUid=home	COMMITTED	20090629_095846.tar.gz

2071 Entries

Figure 7-20: Archived revisions

If the "System data" option was enabled in the objects area of the archiving schedule (see Chapter 7.5.9.1 page 328), the archived revisions are displayed in this list with the following information:

Revision: Number of the archived revision

Date: Date of the revision

Modifier: Name of the user who made the change

Comment: Automatically issued comment

Status: gives the processing status of the revision (e.g. `COMMITTED` for completed revisions, `UNCOMMITTED` for revisions which have been created but not completed)

Archive: Name of the file in which the revision was archived

Install archive : Use this button to reinstall the selected archive with all its elements back into the project. The archive file is not deleted by its installation and can be installed again at a later date.

In case of more than one archive file they must be reinstalled in the correct order. Otherwise there my result gaps in the version history. For this purpose, the archive files must be installed in reversed order, in descending order of date. I.e. the last created archive file should be reinstalled at first, e.g. (vgl. Figure 7-17)

1. Installation of archive file of 19.08.2008, 13:34
2. Installation of archive file of 11.08.2008, 11:31

> ! *Deleted archives cannot be installed.*

If the archive files are not reinstalled in this order this can lead to “IllegalArgumentException“ error messages in the form "unknown revision id: 18, latest: 46" if the version history is opened on an affected object.

Search for ID : Use this button to search through the archives for IDs of the archived elements. All archives that exist for the project are searched through during the search.

Search for UID : This button can be used to browse through the archives for the UIDs of the archived elements.

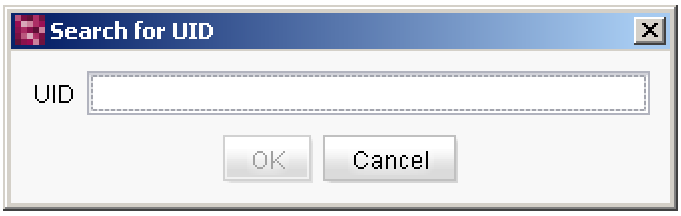
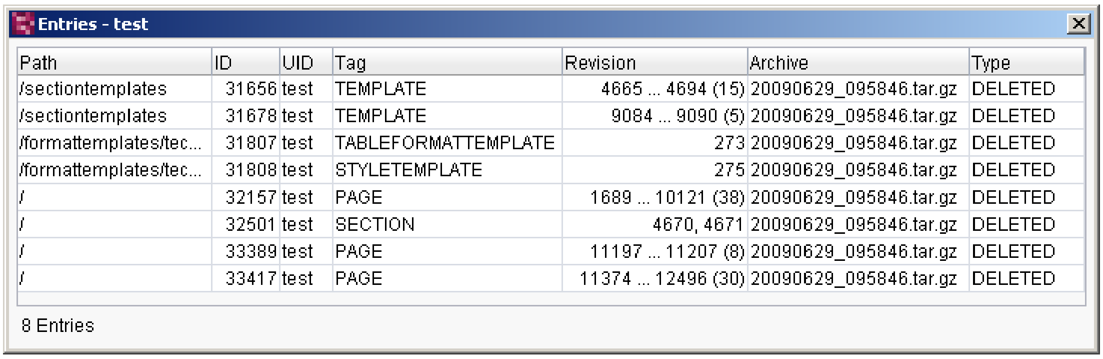


Figure 7-21: Search for UID

All archives that exist for the project are searched through during the search. The results are listed in the following window:

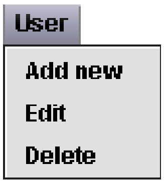


Path	ID	UID	Tag	Revision	Archive	Type
/sectiontemplates	31656	test	TEMPLATE	4665 ... 4694 (15)	20090629_095846.tar.gz	DELETED
/sectiontemplates	31678	test	TEMPLATE	9084 ... 9090 (5)	20090629_095846.tar.gz	DELETED
/formattemplates/tec...	31807	test	TABLEFORMATTEMPLATE	273	20090629_095846.tar.gz	DELETED
/formattemplates/tec...	31808	test	STYLETEMPLATE	275	20090629_095846.tar.gz	DELETED
/	32157	test	PAGE	1689 ... 10121 (38)	20090629_095846.tar.gz	DELETED
/	32501	test	SECTION	4670, 4671	20090629_095846.tar.gz	DELETED
/	33389	test	PAGE	11197 ... 11207 (8)	20090629_095846.tar.gz	DELETED
/	33417	test	PAGE	11374 ... 12496 (30)	20090629_095846.tar.gz	DELETED

8 Entries

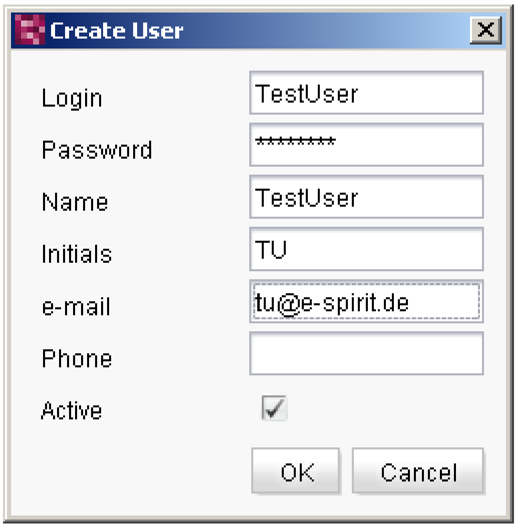
Figure 7-22: Results list for UID search for "test"

7.2.4 User



7.2.4.1 Add new

Use this function to create a new user on the server. A window appears in which user data can be entered.



Create User

Login	TestUser
Password	********
Name	TestUser
Initials	TU
e-mail	tu@e-spirit.de
Phone	
Active	☑

Figure 7-23: User – Add new

Login: New user’s login name (required field).

Password: New user’s password (required field).

*In the exceptional case in which LDAP, Kerberos or NTLM are **not** used for authentication, only ASCII characters in passwords are checked on signing on.*

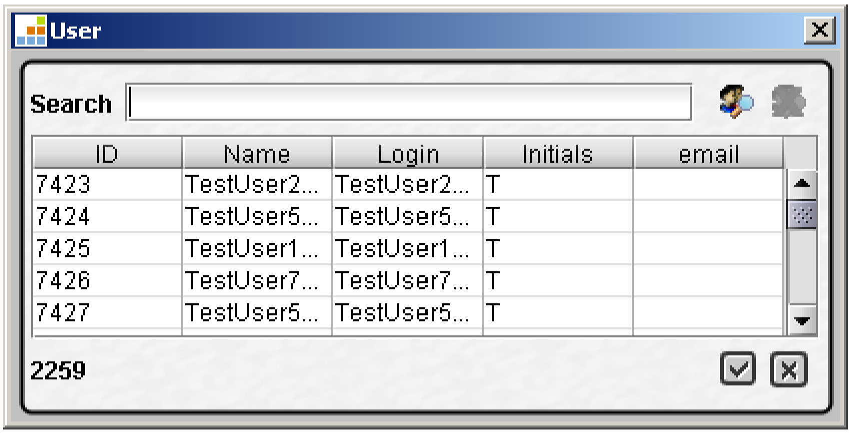
Further information on the new user, i.e. real **Name**, **Initials**, **Email** address and **Phone** number, is optional.

Special case "external user": If users are identified via a third-party system, they are automatically created as FirstSpirit users during the initial login process. The required user attributes are imported from the third-party system (the password is pre-allocated with a random value in FirstSpirit). Users are subsequently displayed in the FirstSpirit user list (see Figure 7-24).

Active (from V4.1): If this option is not active the new user is created directly in the system but can not authenticate himself (for “Deactivating users“ see chapter 7.2.4.2 page 205).

7.2.4.2 Edit

Use this function to change the aforementioned user entries at a later date. A sorted list of registered users is displayed. (Click on any column header to change the sort sequence.)



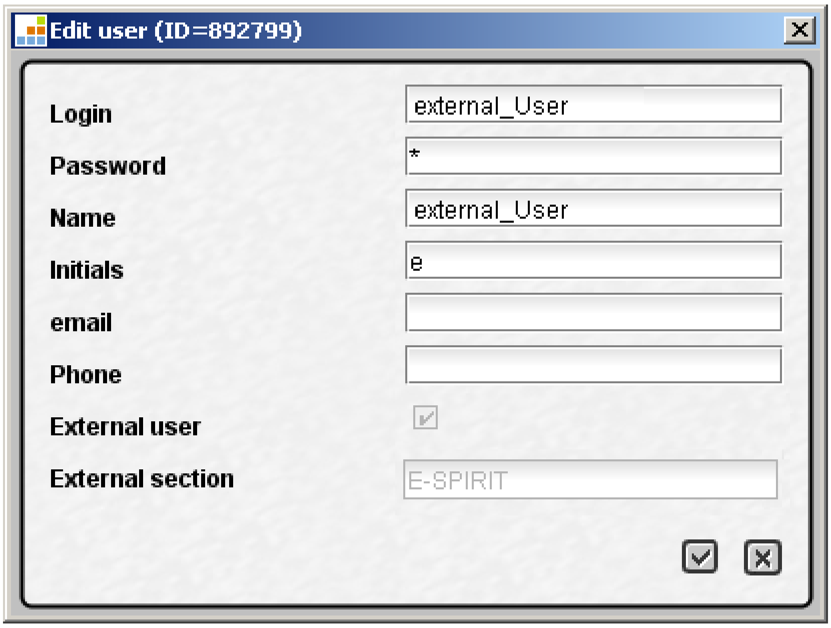
ID	Name	Login	Initials	email
7423	TestUser2...	TestUser2...	T	
7424	TestUser5...	TestUser5...	T	
7425	TestUser1...	TestUser1...	T	
7426	TestUser7...	TestUser7...	T	
7427	TestUser5...	TestUser5...	T	

2259

Figure 7-24: Edit user

Search: Use this search function to search for words or parts of a word in the Name and Login columns. Click on 🔍 to start the search and click on the button next to it to stop the search.

After selecting a user from the list, the edit window opens for changing user data.



Edit user (ID=892799)

Login: external_User

Password: *

Name: external_User

Initials: e

email:

Phone:

External user: ☑

External section: E-SPIRIT

Figure 7-25: Edit user

If the user has been manually created as a FirstSpirit user, all the entries, including user name and password, can be changed. The user is identified internally via a unique user ID. This ensures that user information is retained (e.g. allocation to a project) even if user attributes (e.g. name) are changed.

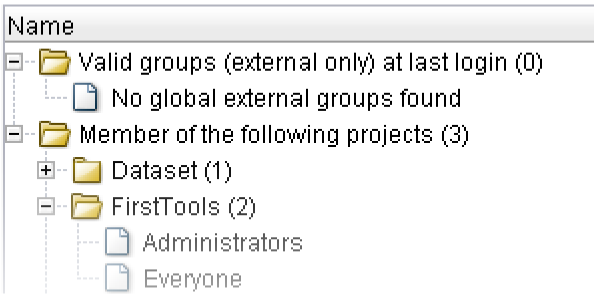
If the user has been automatically created as a FirstSpirit user who is identified via a third-party system, the user attributes cannot be changed via FirstSpirit. External users are also displayed if the “External user” box has been checked.

Active (from V4.1): From FirstSpirit Version 4.1, users can now be deactivated without finally removing the users from FirstSpirit. In this case the users are retained in the system, however, they can no longer authenticate themselves (even with authentication via external systems, e.g. SSO). The deactivate users are shown in grey within the FirstSpirit editing environment and in server monitoring. Deactivated users can be reactivated at any time using this dialog. All user information (e.g. the user's original ID) and all project assignments are retained and can be directly reused (unlike delete and renewed creation of a user).

Special case "External user": The check box is *activated* if the user is a user who has been automatically created on the server from a third party system (e.g. from the LDAP) (see section 7.4.7.2 page 272).
If the check box is *deactivated*, the user has been created manually (see Figure 7-23).

External section: The LDAP section on which the user is registered is displayed here. If login occurs via the WindowsLoginModule, the domain is displayed as an external section here.

Group membership: The group membership of the user is displayed in the area at the bottom of the window. The assignment occurs according to the projects and can not be edited in the dialogue. For changing the group membership of a user see Chapter 7.4.8.5 page 277. This information is also available via the FirstSpirit Server Monitoring (see Chapter 8.5 page 394).



Name
- Valid groups (external only) at last login (0)
  - No global external groups found
- Member of the following projects (3)
  - Dataset (1)
  - FirstTools (2)
    - Administrators
    - Everyone

Figure 7-26: Group membership of user

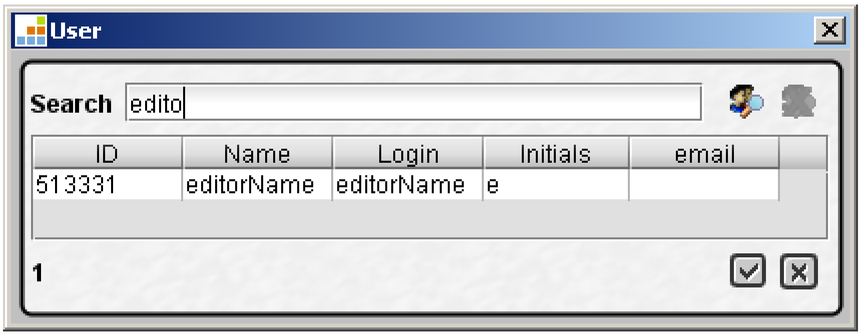
Further information on the new user, i.e. real **Name**, **Initials**, **Email** address and **Phone** number, is optional. These fields might be automatically filled in via user attributes for external users.

The system-internal user ID is automatically assigned and cannot be changed. The permission management for all users can be executed via the JavaClient context menu.

This menu item is only available to server administrators!

7.2.4.3 Delete

Use this function to delete a user from the list.



Search: edito

ID	Name	Login	Initials	email
513331	editorName	editorName	e	

Figure 7-27: Delete user

Search: Use this search function to search for words or parts of a word in the Name and Login columns. Click on 🔍 to start the search.

After selecting a user from the list, a confirmation prompt appears and the user is subsequently deleted. A distinction is made between:

a. Manually created FirstSpirit users:
If users have been manually created, they are automatically deleted from all FirstSpirit projects and the server via the “Delete user” function.

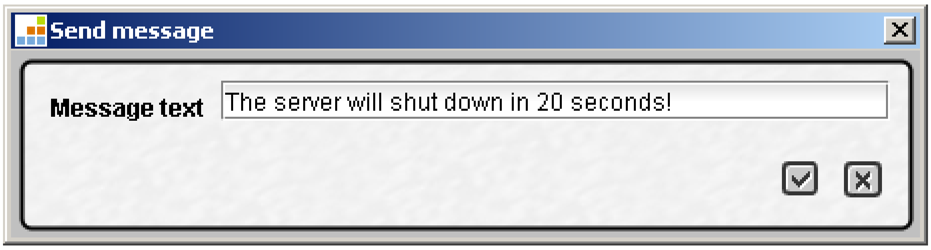
b. Automatically created (external) users:
If the users are external users, they are automatically deleted from all FirstSpirit projects and the server via the “Delete user” function. If the users have not been deleted in the third-party system, they are recreated as new FirstSpirit users with new user ID during a new login.

This menu item is only available to server administrators!

7.2.5 Extras

7.2.5.1 Send message

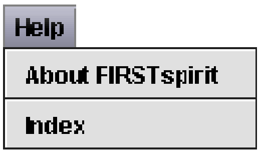
Here, it is possible to edit a message which is transmitted to all active clients on the server as a pop-up window.



Message text: The server will shut down in 20 seconds!

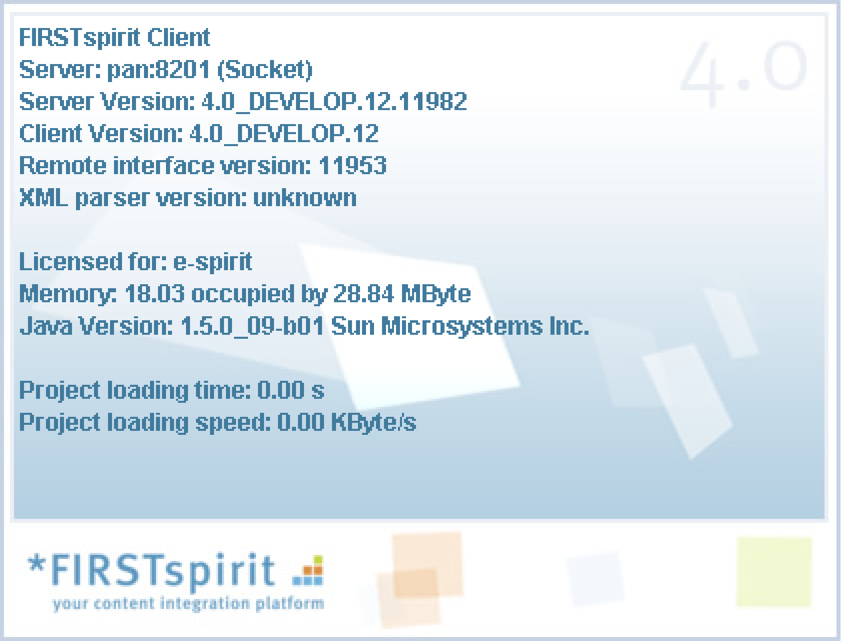
This is particularly useful if the server is to be shut down. This means, editors have enough time to save their work, ensuring data is not lost.

7.2.6 Help



7.2.6.1 About FirstSpirit

Use this function to obtain an overview of certain specifications of the currently used FirstSpirit version:



FIRSTspirit Client
Server: pan:8201 (Socket)
Server Version: 4.0_DEVELOP.12.11982
Client Version: 4.0_DEVELOP.12
Remote interface version: 11953
XML parser version: unknown

Licensed for: e-spirit
Memory: 18.03 occupied by 28.84 MByte
Java Version: 1.5.0_09-b01 Sun Microsystems Inc.

Project loading time: 0.00 s
Project loading speed: 0.00 KByte/s

7.2.6.2 Index

Select this function to open the “Manual for Administrators” (.pdf) in the browser. Chapter “Server and Project Configuration” is displayed automatically.

7.3 Server properties

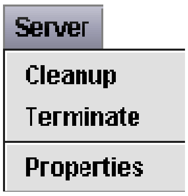
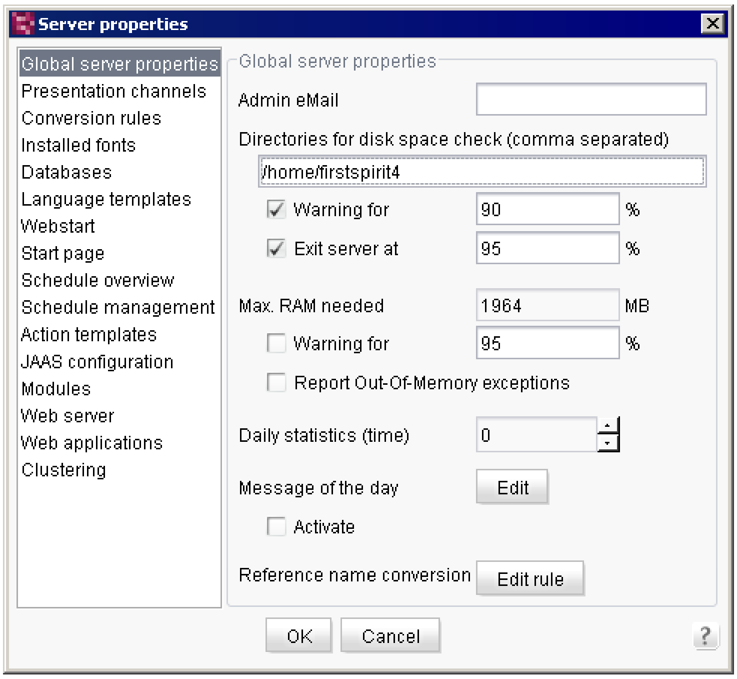


Figure 7-28: Server menu – Server properties

Configure the FirstSpirit Server properties via menu item “Server properties”.

7.3.1 Global server properties


Server properties

Global server properties
Presentation channels
Conversion rules
Installed fonts
Databases
Language templates
Webstart
Start page
Schedule overview
Schedule management
Action templates
JAAS configuration
Modules
Web server
Web applications
Clustering

Global server properties

Admin eMail

Directories for disk space check (comma separated)

/home/firstspirit4

☑ Warning for 90 %

☑ Exit server at 95 %

Max. RAM needed 1964 MB

□ Warning for 95 %

□ Report Out-Of-Memory exceptions

Daily statistics (time) 0

Message of the day Edit

□ Activate

Reference name conversion Edit rule

OK Cancel ?

Figure 7-29: Server properties – Global server properties (from 4.2R4)

The first menu item “Global server properties” contains the following configuration options:

Admin eMail: The email address of the responsible server administrator can be entered in this field. Critical errors which concern the subsequently configured fields can be sent by email to the address specified here. If an email address has not been specified, the respective error messages are only output in the log output of the server.

Max. disk space (up to 4.2R4): The maximum disk space of the server (in MB) can be set in this field. A critical threshold value (in %) can also be defined. If the corresponding box on the left is checked, a warning is written into the server log as soon as the critical value of the maximum available memory space is exceeded (an email can be optionally sent to the server administrator).

Directories for disk space check (comma separated) (from 4.2R4): With version 4.2R4 several measures have been implemented for being able to better control already used or still free disk space and thus to avoid possible data loss.
For this purpose, enter the directories in this field which are to be controlled regarding the disk space which is still available. By default, the base directory of the FirstSpirit server is specified (see also `fs-server.conf`, parameter `hdd.directories`, Chapter 4.3.1.16 page 59). If FirstSpirit is distributed across several mount points, all mount points of the file systems which are used by FirstSpirit, must be specified here. If there is more than one directory, they must be separated by comma. The free disk space in the specified directories is checked within an interval of five minutes.

> ! *For technical reasons this controlling of the disk space is only possible with JDK 1.6.*

Warning for (from 4.2R4): Enter the percentage in this field, from which a warning e-mail is to be sent to the server administrator. Subject and content of the e-mail: "FIRSTspirit server 'MYSERVER' disk space warning: /home/fs/firstspirit4, 32,01 GB free, 42,52 GB used".
The percentage results from the relation of the disk space of the indicated directory which is still free and which is already occupied. By default, the value is set to 90%, i.e. the server administrator will get a warning e-mail if 90% of the disk space available are occupied (see also `fs-server.conf`, parameter `hdd.limit`, Chapter 4.3.1.16 page 59). If more than one directory is specified, an e-mail will be sent if the percentage in **one** of the directories is exceeded. The value you enter here should be lower than the value for "Exit server at" (see below). Warning e-mails are sent every 12 hours maximal.
The checkbox can be used to activate or deactivate the sending of warning e-mails (see also `fs-server.conf`, parameter `hdd.limit.active`, Chapter 4.3.1.16 page 59). By default, the sending is activated for the indicated percentage.

Exit server at (from 4.2R4): Enter the percentage in this field, at which the FirstSpirit server is to be shut down and an e-mail is to be sent to the server administrator. Subject and content of the e-mail: "FIRSTspirit server 'MYSERVER' disk space shutdown limit reached: /home/fs/firstspirit4, 463.2 MB free, 5.54 GB

used".
The percentage results from the relation of the disk space of the indicated directory which is still free and which is already occupied. By default, the value is set to 95%, i.e. the server will be shut down if 95% of the disk space available are occupied (see also `fs-server.conf`, parameter `hdd.shutdown`, Chapter 4.3.1.16 page 59). If more than one directory is specified, the server will be shut down, if the percentage in **one** of the directories is exceeded. The value you enter here should be higher than the value for "Warning for" (see above).
The checkbox can be used to activate or deactivate the shut down of the server (see also `fs-server.conf`, parameter `hdd.shutdown.active`, Chapter 4.3.1.16 page 59). By default, the shut down of the server at the indicated percentage is activated.

Max. RAM needed: The maximum RAM requirement (in MB) for the virtual machine of the server can be set in this field. This value is transferred during server start-up. A critical threshold value (in %) can also be defined. If the corresponding box on the left is checked, a warning is written into the server log as soon as the critical value of the maximum available RAM is exceeded (an email can be optionally sent to the server administrator). If the box “Report Out-Of-Memory exceptions” is also checked, an email is sent to the system administrator if an Out-Of-Memory exception occurs on the server.

Example:

```
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 this case, the server is started with: `-Xmx=900m` (HEAP STORAGE max + NON_HEAP STORAGE max) and the following values are set in the Server and Project Configuration in the field "Max. RAM needed":

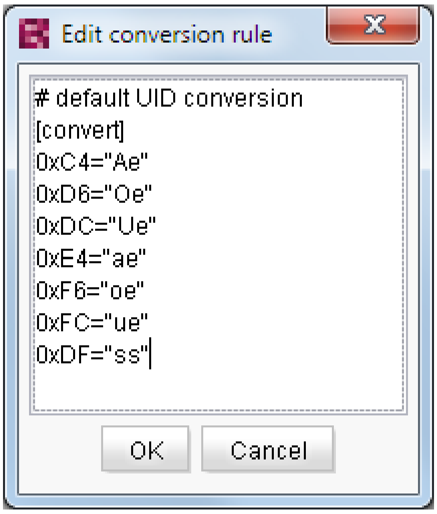
- 90% (HEAP STORAGE used / HEAP STORAGE max=91%, i.e. more than 90%)

Daily statistics (Time): The start time of daily statistic execution can be set here.

Message of the day: A text which appears as a message during start-up can be entered for each project language via **Edit**. The message will only appear in the

FirstSpirit JavaClient when opening a project if the “Activate” box has been checked.

Reference name conversion: With FirstSpirit Version 4.2R4, a server-wide rule set can be defined, according to which invalid characters are to be transformed on creating new FirstSpirit objects or changing the reference name. With clicking the button "Edit rule" the following window opens:



```
# default UID conversion
[convert]
0xC4="Ae"
0xD6="Oe"
0xDC="Ue"
0xE4="ae"
0xF6="oe"
0xFC="ue"
0xDF="ss"
```

Figure 7-30: Reference name conversion

Several rules are already defined here as a default, namely for the umlauts "ä", "ö" and "ü" in lower and upper case and for "ß". Each rule must be positioned in a separate line and consists of two values, separated by an equals sign:

- **on the left** the special character to be transformed, in ASCII code (hexadecimal)
- **on the right** the valid character(s), into which the special character is to be transformed when used in reference names, in double inverted commas.

The default rule set transforms umlauts used in reference names into 2 characters (vowel in lower case + "e"), "ß" is transformed into "ss".

If the characters in this dialog are not given in the correct code or with the correct formatting, an error message is displayed in the following form when the rule set is saved:
"The conversion rule format is incorrect: Error parsing line 11:...".

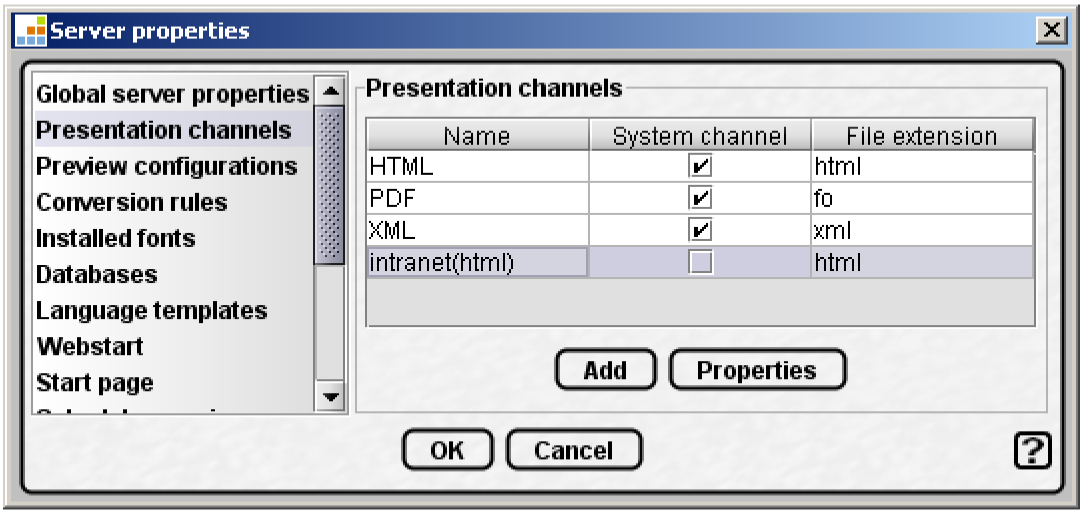
Rules can only be defined for single characters, not for character strings, and only language-independent, i.e. each rule applies to all languages. If more than one definition is given for a special character (ASCII code), the lowest definition is always applied. Special characters for which no rule is deposited continue to be removed directly on entering a reference name.

When defining conversion rules for the symbol . (dot), it should be taken into account that it is used in FirstSpirit to generate reference names for table and link templates and is also converted if a corresponding rule definition exists.

The rules supplied can be changed or deleted and new rules can be added. They are not reset or supplemented during a FirstSpirit update. Comments can be entered, they must be introduced by a #.

Click "OK" to save the rules. If the dialog below it with the Server properties is closed with "OK", the rules are immediately applied; the server or project do not have to be restarted first.

7.3.2 Presentation channels



Presentation channels

Name	System channel	File extension
HTML	☑	html
PDF	☑	fo
XML	☑	xml
intranet(html)	□	html

Figure 7-31: Server properties – Presentation channels

The presentation channels of the server are defined in this area. The template sets available on the server are based on the presentation channels defined here.

Add: Click on “Add” to open a dialogue window for defining a new presentation channel (see Figure 7-32).

Properties: Highlighting a presentation channel from the list activates the “Properties” button. Click on the button to open a dialogue window for editing the properties of the presentation channel (see Figure 7-32).

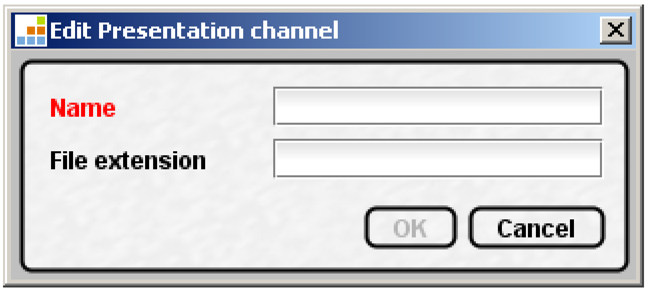


Figure 7-32: Server properties – Edit presentation channels

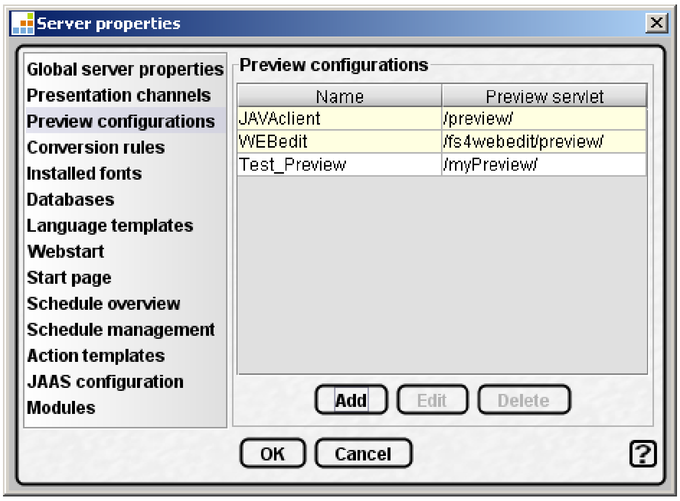
Name: Name of the presentation channel. The presentation channel is displayed with this name in the project and in the presentation channel list after saving.

System channel: The check box indicates whether the presentation channel is a system channel. These presentation channels cannot be edited or deleted.

File extension: The file extension of the presentation channel is stated in this field, e.g. “html” for the HTML presentation channel.

About supporting Apache FOP see chapter 7.8 page 368.

7.3.3 Preview configurations



Preview configurations

Name	Preview servlet
JAVAclient	/preview/
WEBedit	/fs4webedit/preview/
Test_Preview	/myPreview/

Figure 7-33: Server properties – Preview settings

Preview settings can be configured in this area. The default preview settings for the JavaClient and WebClient have been configured by default. The default settings are yellowed out and cannot be edited or deleted.

Name: Unique name for the configuration. Configuration can be carried out for the project preview under this name in the project properties (see section 7.4.13 page 282).

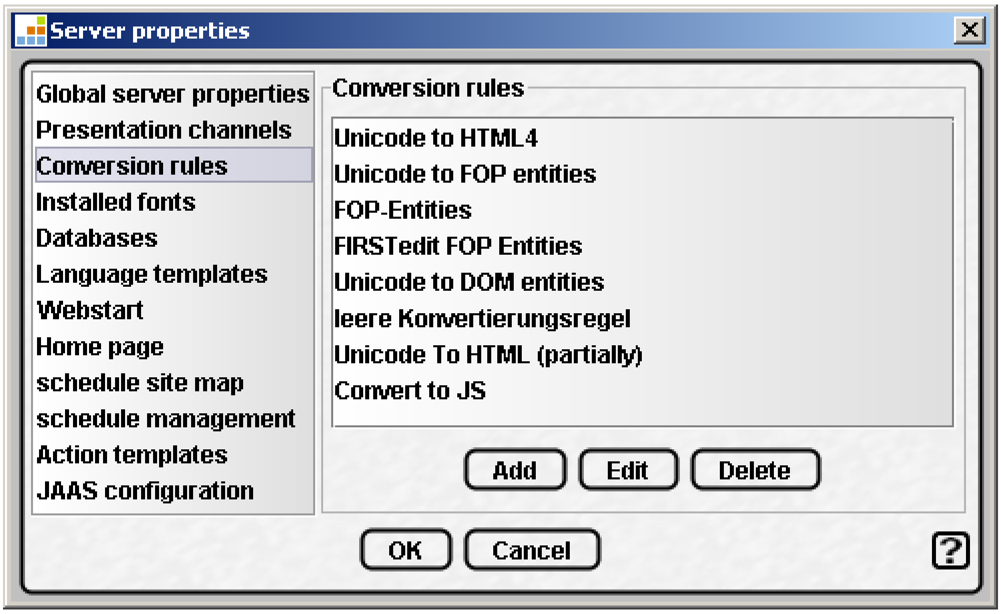
Preview servlet: Path specification to the directory in which the preview servlet is located.

Add: Click on this button to create a new preview configuration. To achieve this, a unique name for the new configuration and the path to the directory in which the preview servlet is located have to be created.

Edit: Click on this button to edit a preview configuration. The default settings are yellowed out and cannot be edited.

Delete: Click on this button to edit a preview configuration. The default settings are yellowed out and cannot be deleted.

7.3.4 Conversion rules



Unicode to HTML4
Unicode to FOP entities
FOP-Entities
FIRSTedit FOP Entities
Unicode to DOM entities
leere Konvertierungsregel
Unicode To HTML (partially)
Convert to JS

Figure 7-34: Server properties – Conversion rules

The conversion rules of the server are defined in this area. The conversion rules defined here can be selected within the format templates of the project from a fold list for each presentation channel. Default conversion rules are, for example, "Unicode to HTML4" for the HTML output channel and "FOP-Entities" for the PDF output channel. One conversion rule can be set for each output channel.

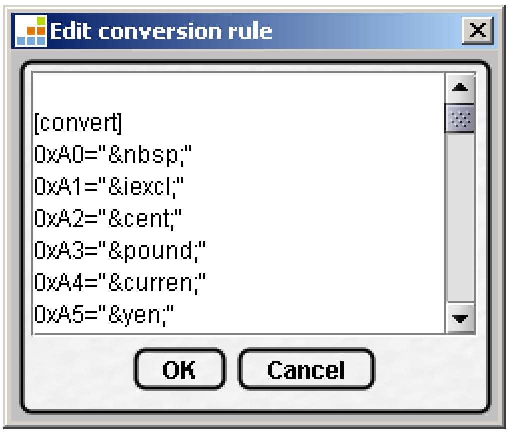
Conversion rules are used to convert entered characters. A conversion rule consists of two parts:

- “convert” is always evaluated if a conversion rule has been selected for a format template in the client.
- if “quote” is to be additionally evaluated, the respective toggle has to be activated in the format template.

For further information on format templates see the “FirstSpirit Manual for Developers".

Add: A selection dialogue opens in which it is possible to browse the local file structure of the computer. A new conversion rule has to be created as a test file in advance.

Edit: An existing conversion rule can be edited via the dialogue “Edit conversion rule”.

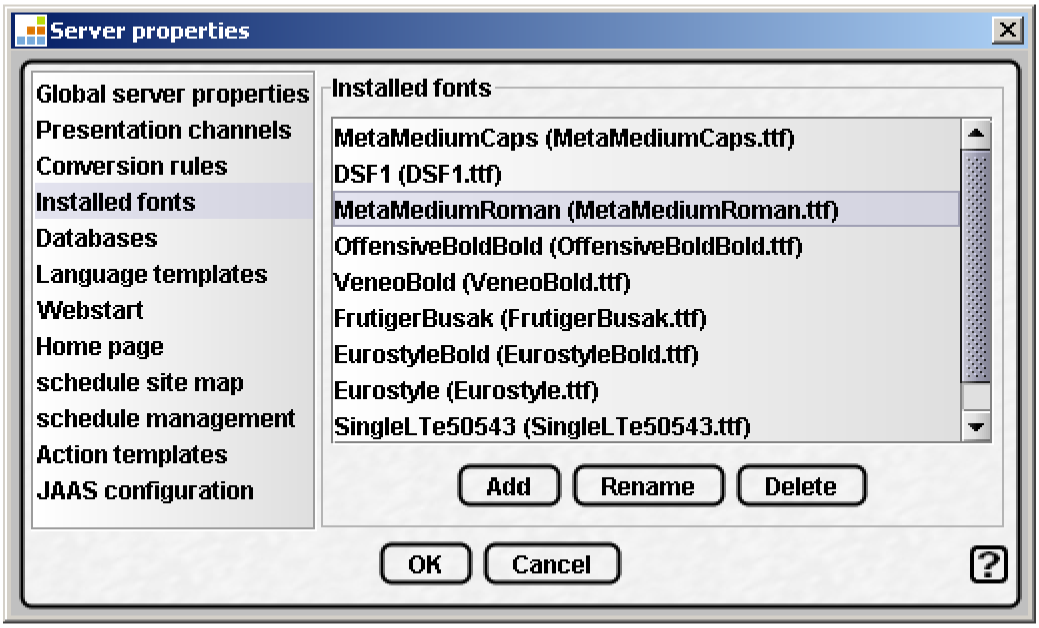


```
[convert]
0xA0="&nbsp;"
0xA1="&iexcl;"
0xA2="&cent;"
0xA3="&pound;"
0xA4="&curren;"
0xA5="&yen;"
```

Figure 7-35: Server properties – Edit conversion rule

Delete: Click on this button to delete existing conversion rules from the server. A confirmation prompt appears prior to deleting.

7.3.5 Installed fonts


Server properties

Global server properties
Presentation channels
Conversion rules
Installed fonts
Databases
Language templates
Webstart
Home page
schedule site map
schedule management
Action templates
JAAS configuration

Installed fonts

MetaMediumCaps (MetaMediumCaps.ttf)
DSF1 (DSF1.ttf)
MetaMediumRoman (MetaMediumRoman.ttf)
OffensiveBoldBold (OffensiveBoldBold.ttf)
VeneoBold (VeneoBold.ttf)
FrutigerBusak (FrutigerBusak.ttf)
EurostyleBold (EurostyleBold.ttf)
Eurostyle (Eurostyle.ttf)
SingleLTe50543 (SingleLTe50543.ttf)

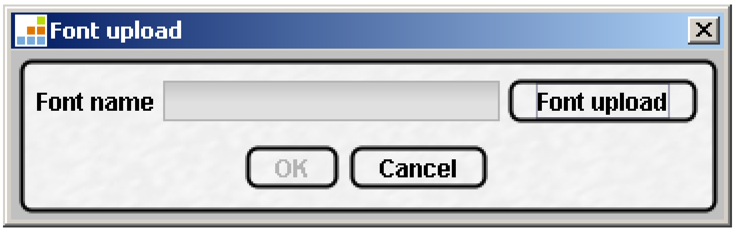
Figure 7-36: Server properties – Install fonts

The True Type Fonts (TTF) installed on the server are listed here. These fonts can be accessed in a template, e.g. to provide images with a text in the respective font:

```
$CMS_REF(cmsFont(font:"MetaMediumCaps",size:12,color:"#666666",jus
tify:"right",valign:"center",xoffset:-4,yoffset:-
2,media:"version_from",bounds:"image",text:#content.toString))$
```

The reference name via which the font can be referenced and the file name (in brackets) under which the font has been stored in the file system are displayed (see Figure 7-36).

Add: If another font is to be added, it must have been installed on the operating system of the server computer. Installing the font on the operting system of the client computer is insufficient. In order to add a new font on the server, a unique reference name of the new font has to be specified first. Click on “Font upload” to select a True Type Font from the file system.


Font upload

Font name

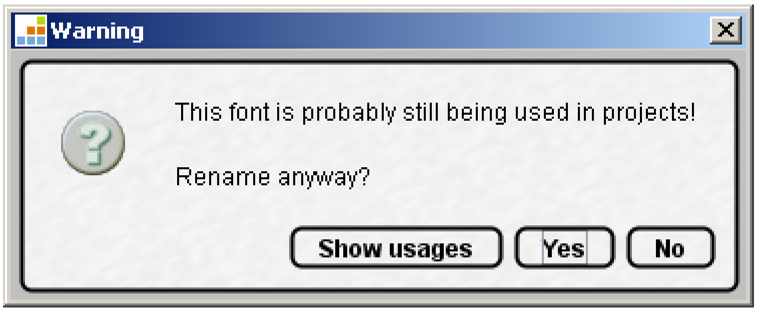
Figure 7-37: Create new font

Confirm the selection via “OK” to create the new font.

Rename: Renaming a font displayed in the list.

Delete: Deleting the font highlighted in the list. The font is removed from the server directory `data/fonts` and no longer displayed in the list of installed fonts.

The renaming or deleting of a font can lead to errors in the templates which use this font. Therefore, a confirmation prompt always appears.



Warning

This font is probably still being used in projects!

Rename anyway?

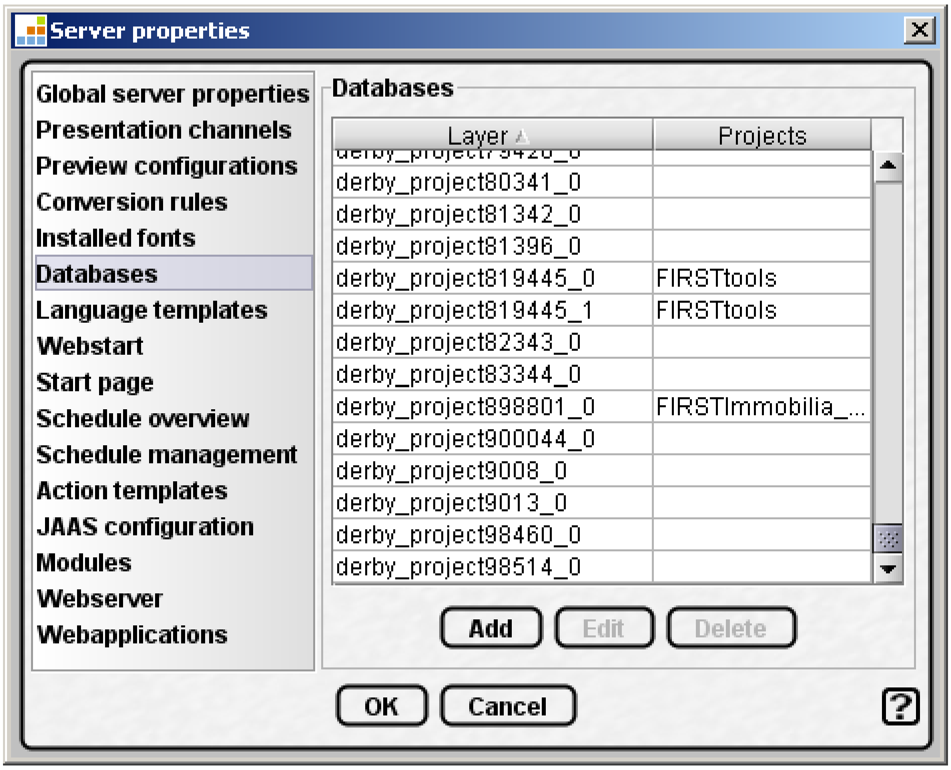
Figure 7-38: Warning prior to deleting or renaming fonts

OK Click on this button to rename or delete the font; this may lead to errors in the projects which use this font. In case of doubt, use the “Show usages” button to determine these projects.

Show usages Click on this button to determine utilisation of a font in the projects. The projects are displayed in a list. A message appears if no utilisation is found.

Cancel Click on the button to close the dialogue. The affected font is neither deleted nor renamed.

7.3.6 Databases



Layer	Projects
derby_project80341_0	
derby_project81342_0	
derby_project81396_0	
derby_project819445_0	FIRSTtools
derby_project819445_1	FIRSTtools
derby_project82343_0	
derby_project83344_0	
derby_project898801_0	FIRSTImmobilia_...
derby_project900044_0	
derby_project9008_0	
derby_project9013_0	
derby_project98460_0	
derby_project98514_0	

Figure 7-39: Server properties – Databases

All the database layers integrated on the server are listed in this area, i.e. all connections to a database available on the server. A database layer defined here is selected in the project properties for a project (see section 7.4.12 page 281) and can be subsequently selected in the Template-Store of the project when defining a database schema (for further information on using database layers in the project see the “FirstSpirit Manual for Developers (Basics)”).

Add Click on this button to add a new database layer on the server. A window opens in which only the name of the new configuration has to be entered.

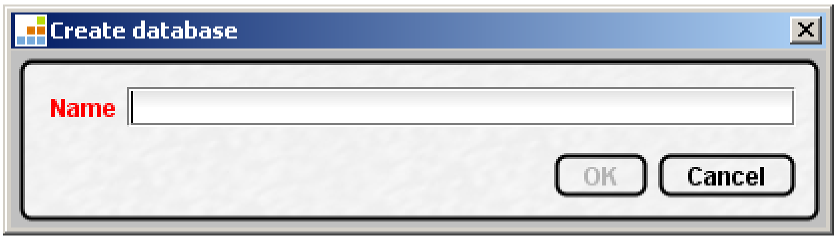


Figure 7-40: Create a new database layer

A window subsequently appears for database configuration; the basic structure for a

MySQL database has already been entered in this window. The database configuration now has to be adapted to the local requirements (see section 7.3.6.1 ff). (For detailed information on database connection see 0 page 102 and on database configuration see section 4.8.4 page 137.)

> ! *FirstSpirit uses JDBC[24] for database access. An appropriate driver has to be added to the class path of the server for each database or the JDBC driver files have to be integrated as FirstSpirit module (see also Chapter 4.8.1 page 124).*

Edit Click on this button to edit the existing configuration. A window with the existing configuration opens (see Figure 7-41). It is distinguished between:

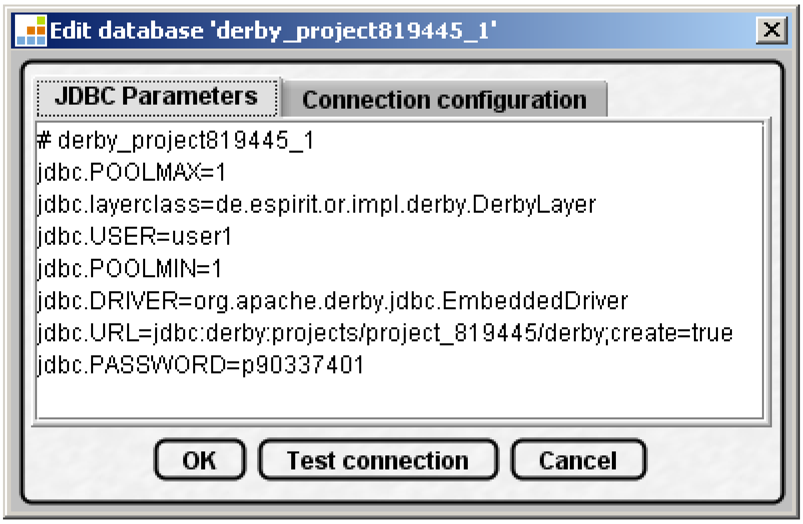
- JDBC parameter configuration (see section 7.3.6.1 page 222)
- Connection configuration (see section 7.3.6.2 page 222)

Delete Click on this button to delete an existing database layer from the FirstSpirit Server. If the layer represents an embedded Derby database, click on this button to delete the respective database in addition to the layer. (For all other layers, the respective database has to be manually deleted by the database administrator.)

> ! *A confirmation prompt appears prior to deleting databases currently used in a project.*

[24] Java Database Connectivity

7.3.6.1 Configure JDBC parameters



```
# derby_project819445_1
jdbc.POOLMAX=1
jdbc.layerclass=de.espirit.or.impl.derby.DerbyLayer
jdbc.USER=user1
jdbc.POOLMIN=1
jdbc.DRIVER=org.apache.derby.jdbc.EmbeddedDriver
jdbc.URL=jdbc:derby:projects/project_819445/derby;create=true
jdbc.PASSWORD=p90337401
```

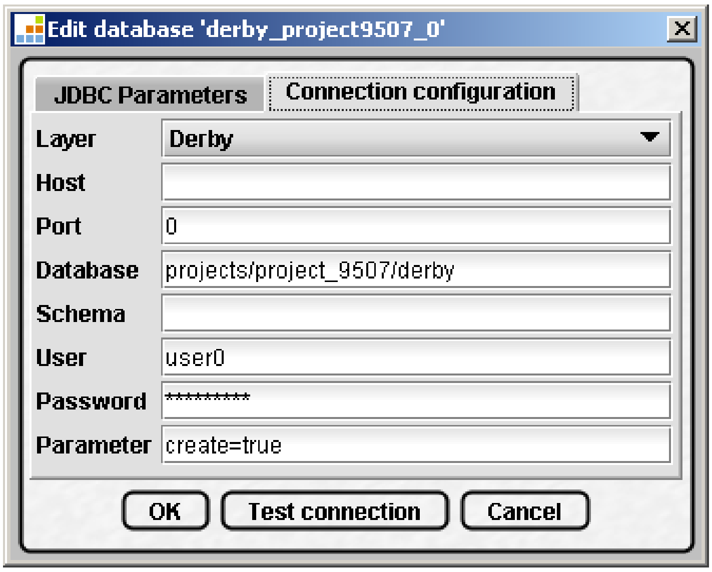
Figure 7-41: Configure database connection (JDBC)

JDBC parameter configuration for the database connection.

The configuration can be changed and subsequently saved by clicking on “OK”. Before saving a change, always test the connection to the database. See section 4.8.1 page 124 for detailed information on configuring the database.

Test connection: Once configuration of the new database has been completed, click on this button to test whether all the specifications have been entered correctly.

7.3.6.2 Test connection configuration



Layer: Derby
Host:
Port: 0
Database: projects/project_9507/derby
Schema:
User: user0
Password: *********
Parameter: create=true

Figure 7-42: Configure database connection (connection)

Connection parameter configuration for the database connection.

Layer: Über die Combobox wird die Klasse angegeben, die den Datenbanklayer für dieses spezielle Datenbanksystem implementiert (vgl. Parameter `layerclass` section 4.8.4.1 page 138).

Host: Host name for the bind address of the database server. When specifying a Derby layer, this field should remain empty.

Port: Port number for the bind address of the database server. When specifying a Derby layer, this field should remain empty.

Database: Name of the database in which the data is to be stored. This value is pre-allocated by default with the name of the database layer.

Schema: A distinction is made between:

- Standard layers: An existing DB schema in the layer configuration is specified via parameter `jdbc.SCHEMA`. The name of the schema has to be identical with the name of the schema in the database. In this case, all database contents from FirstSpirit are written into this schema.
- DBA layers: Parameter `jdbc.SCHEMA` is not defined. In this case, each FirstSpirit schema creates an independent database schema for the respective database contents.

User: A database user's valid login name. The FirstSpirit Server uses this account to establish a connection to the database during runtime. (see parameter `jdbc.user` section 4.8.4.1 page 138)

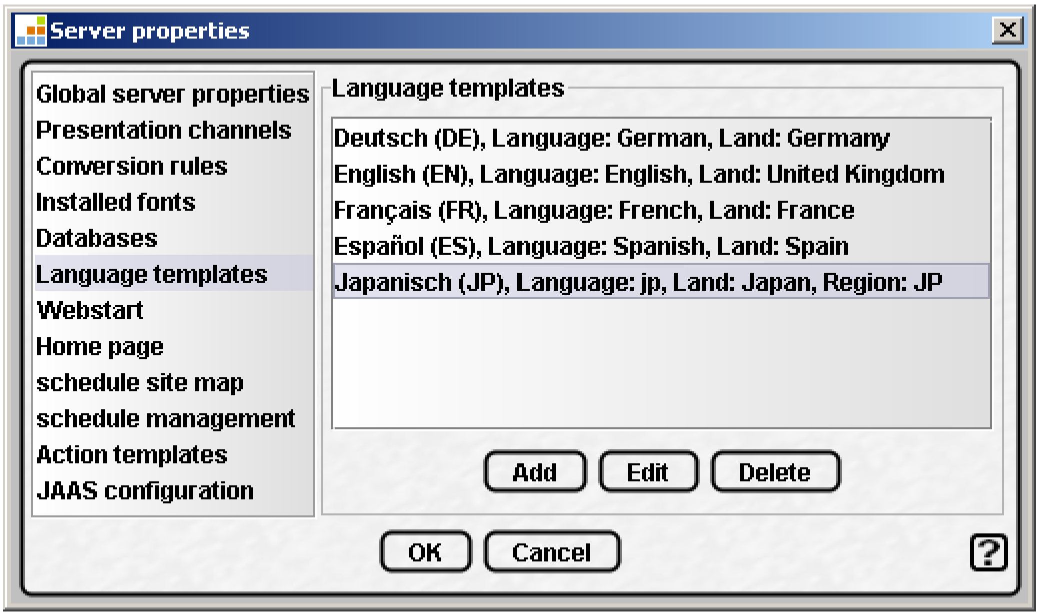
Password: Valid password for login under `<database>.jdbc.USER`. (see parameter `jdbc.password` section 4.8.4.1 page 138)

Parameters: Input option for further database-specific parameters.

The configuration can be changed and subsequently saved by clicking on "OK". Before saving a change, always test the connection to the database. See section 4.8.1 page 124 for detailed information on configuring the database.

Test connection: Once configuration of the new database has been completed, click on this button to test whether all the specifications have been entered correctly.

7.3.7 Language templates



Deutsch (DE), Language: German, Land: Germany
English (EN), Language: English, Land: United Kingdom
Français (FR), Language: French, Land: France
Español (ES), Language: Spanish, Land: Spain
Japanisch (JP), Language: jp, Land: Japan, Region: JP

Figure 7-43: Server properties – Language templates

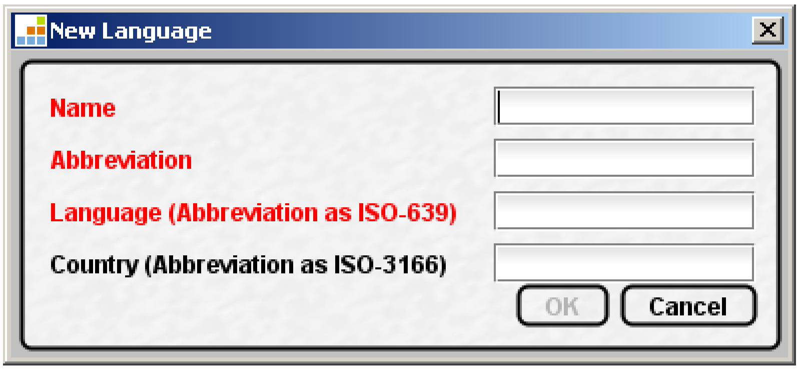
FirstSpirit supports the multilingual output of a publication. If a new language is required in a FirstSpirit project, the language must be created on the server first.

Add: Use this button to integrate a new language into the system. Click on this button to open the "New language" dialogue window (see Figure 7-44). Once a newly created language has been defined, it is available for all projects. However, it has to be integrated into the respective project prior to utilisation (see section 7.4.5 page 265). Only then is the language tab displayed in the editorial environment and ready to be filled with content.

Edit: Click on this button to edit an existing language (see Figure 7-44).

Delete: Click on this button to delete an existing language.

Contents which have already been entered in a language for a project will not be lost if the corresponding language template is edited or deleted. These contents remain as long as the language remains in the project properties of the project (see section 7.4.5 page 265), since they work with a copy of the language template. After deleting a language template in the server properties, it is not possible to re-add the respective language to a project in the project properties.


New Language

Name

Abbreviation

Language (Abbreviation as ISO-639)

Country (Abbreviation as ISO-3166)

Figure 7-44: Server properties – Create new language

The red font colour indicates a required field. If the entry is valid, the font changes to black.

Name: Enter a name for the language in this field. The language name can consist of any characters.

Abbreviation: Enter an abbreviation for the language name in this field. The language is displayed with this designation in the language tabs of the FirstSpirit editorial environment. The value indicated here must be used also in templates for the attribute *lang* (see FirstSpirit Online Documentation, attribute *lang* in all input componentes, Section *Template development / Forms*). The following characters can be used for this as often as required: -, _, *0-9* and *A-Z*. The abbreviation of the language can be oriented e.g. to the ISO norms 3166 or 639. For languages which are spoken in several countries, e.g. British Englisch and American English, composed abbreviations can be used, e.g. *EN-GB* or *EN_GB* for British and *EN-US* or *EN_US* for American English.

The values for language and (optionally) country determine the later locale of the language:

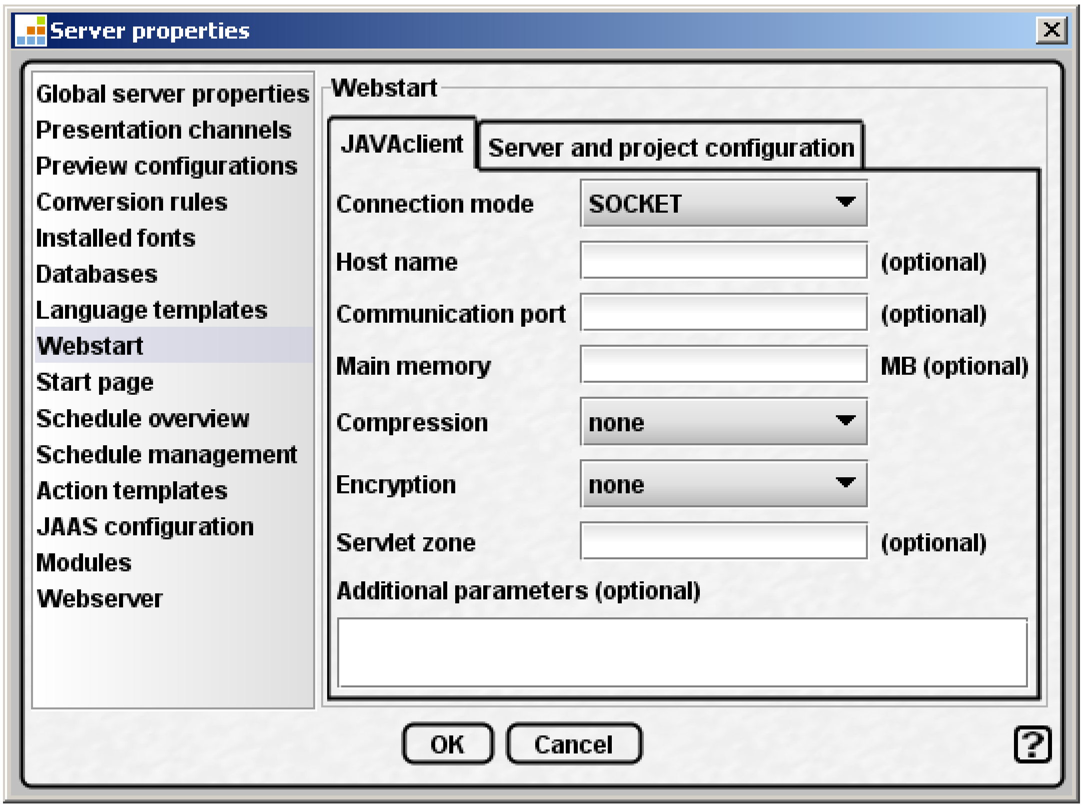
Language: Enter the 2 letter abbreviation for the new language in compliance with ISO-639 in this field, e.g. *de* for German or *en* for English. Only lower case letters can be entered. (First part of the LOCALE)

Country: Enter the 2 letter abbreviation for the country of the new language in compliance with ISO 3166 in this field, e.g. *DE* for Germany or *GB* for England. Only capital letters can be entered. (Second part of the LOCALE)

For example, it is possible to make a distinction between en_US (English in the USA) and en_GB (English in the UK).

From FirstSpirit Version 4.2 for each language template language-dependent display names can be defined. For this, the respective input fields are shown in the form area “Display names“. Language-dependent display names can be defined for all editorial languages of a project.

7.3.8 Webstart



Connection mode: SOCKET
Host name: (optional)
Communication port: (optional)
Main memory: MB (optional)
Compression: none
Encryption: none
Servlet zone: (optional)
Additional parameters (optional)

Figure 7-45: Server properties – Configure Webstart (quick start)

The quick start entries in FirstSpirit 4.0 are configured via JNLP files. The default configurations (default JNLP files) can be defined via menu item “Webstart”.

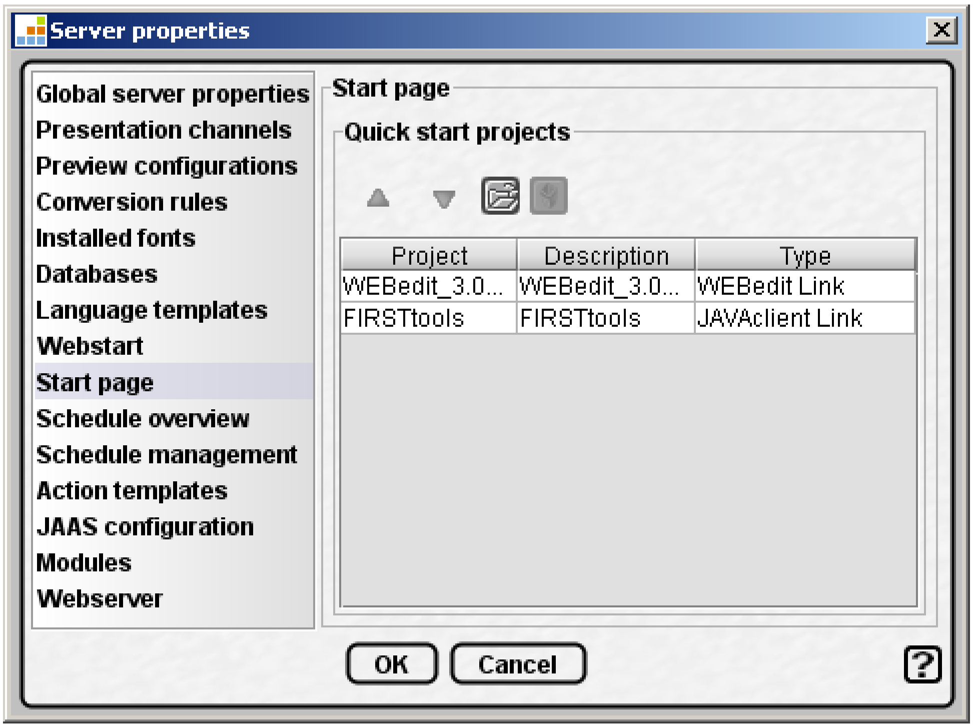
JavaClient tab: Configuration for starting the JavaClient and the quick start entries on the start page. The parameters configured here affect all the JavaClient type quick start entries of the start page JAVA unless other parameters have been explicitly defined for the entry in the “Start page” area.

The individual quick start entries can be configured via the “Start page” dialogue. This overwrites the universal values in the “Webstart” area (see section 7.3.9 page 227).

The configuration options are the same as for the user-specific Webstart configuration (see section 6.3.5.1 page 173).

Server and Project Configuration tab: Configuration for the Server and Project Configuration of the start page. The parameters are the same as for the quick start entries in the “JavaClient” tab.

7.3.9 Start page

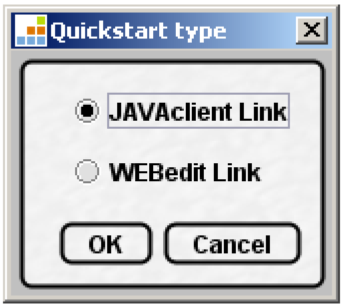


Project	Description	Type
WEBedit_3.0...	WEBedit_3.0...	WEBedit Link
FIRSTtools	FIRSTtools	JAVAclient Link

Figure 7-46: Server properties – Configure start page

Quick start projects: Select projects from all the projects on the server via the icon. These projects are subsequently shown in a tabular overview and displayed below the quick start entries on the FirstSpirit start page (see section 6.3.1 page 171). Click on the icon to delete previously highlighted projects in the table. Use the arrow keys to sort the entries.

Click on a “quick start entry” on the start page to directly open the linked project without having to select it in the project selection list first. Depending on the configuration, the JavaClient as well as the WebClient (only for projects with WebEdit configuration) can be linked for a project.



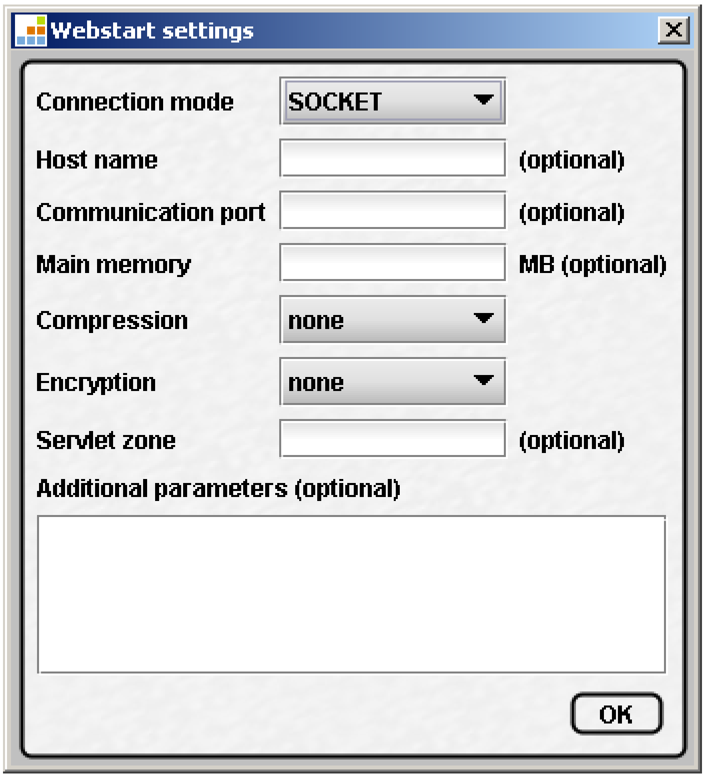
Quickstart type

● JAVAclient Link

○ WEBedit Link

Figure 7-47: Server properties – Create quick start links

Depending on the selected setting, the selected project opens in either the JavaClient or the WebClient. To improve distinction, the quick start entries have either the logo JAVA or WEB. Further connection settings (settings for Java Web Start) can be defined for quick start entries which open in the JavaClient. Double-click on the respective quick start entry to open the dialogue “Webstart settings” (see Figure 7-48). The configuration options are the same as for the user-specific Webstart configuration (see section 6.3.5.1 page 173).



Webstart settings

Connection mode: SOCKET

Host name: (optional)

Communication port: (optional)

Main memory: MB (optional)

Compression: none

Encryption: none

Servlet zone: (optional)

Additional parameters (optional)

Figure 7-48: Webstart settings for project links (only JavaClient)

Evaluation according to priority: The initial configuration is carried out via configuration file `fs-server.conf` (see section 4.3.1.1 page 34). The parameters defined therein can be overwritten by the Server and Project Configuration, i.e. under the menu item “Webstart” for all Java Web Start applications (via the JNLP files) (see section 7.3.8 page 226) and under “Start page” in the properties of the individual quick start entries. The evaluation according to priority is:

1. Quick start properties
2. JNLP file
3. fs-server.conf

If connection settings on the start page are defined and activated (see Chapter 6.3.5.1 page 173), these will be evaluated in addition preferentially.

If the connection settings are deactivated, the `fs.url` parameter of the `fs-server.conf` (beginning with "`fs.url.`", see Chapter 4.3.1.1 page 34) will not be overwritten by the respective parameters, which are defined in the Server properties in the areas "Webstart" and "Start page".

7.3.10 Schedule entry overview

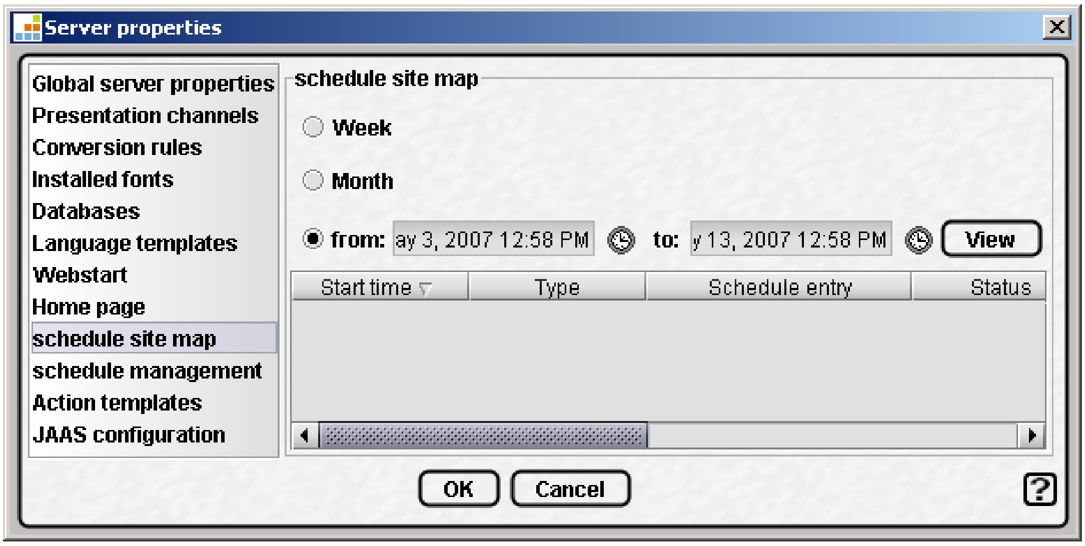
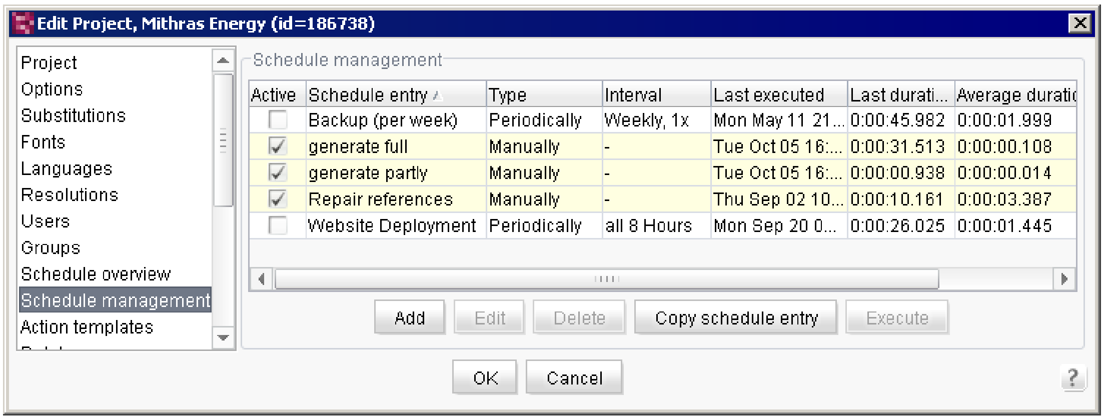


Figure 7-49: Server properties – Schedule entry overview

See section 7.5.1 page 311 ff for the schedule entry overview.

7.3.11 Schedule entry management

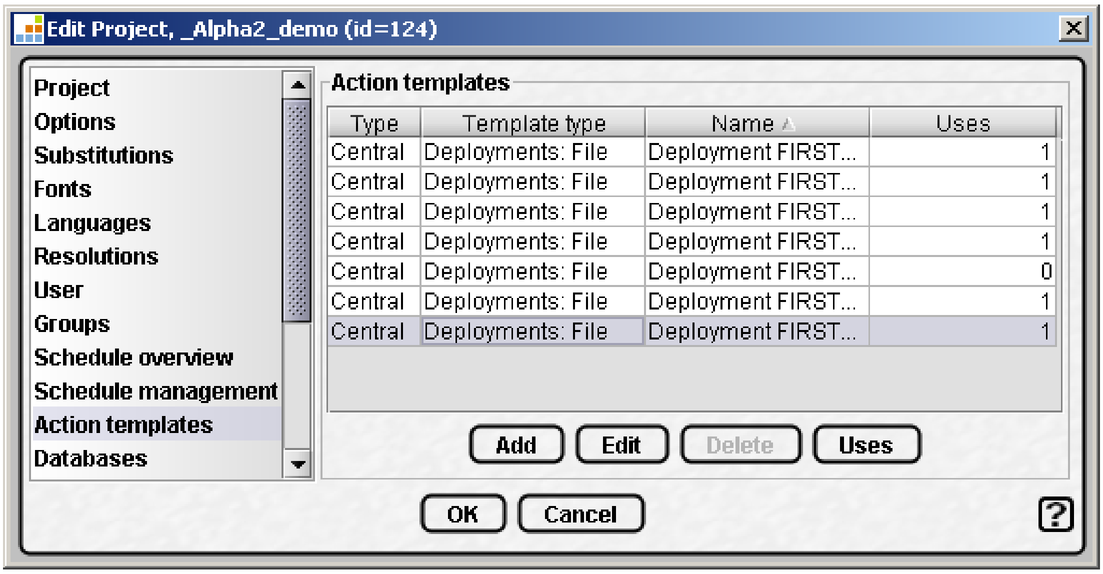


Active	Schedule entry	Type	Interval	Last executed	Last durati...	Average duratio
☐	Backup (per week)	Periodically	Weekly, 1x	Mon May 11 21...	0:00:45.982	0:00:01.999
☑	generate full	Manually	-	Tue Oct 05 16:...	0:00:31.513	0:00:00.108
☑	generate partly	Manually	-	Tue Oct 05 16:...	0:00:00.938	0:00:00.014
☑	Repair references	Manually	-	Thu Sep 02 10...	0:00:10.161	0:00:03.387
☐	Website Deployment	Periodically	all 8 Hours	Mon Sep 20 0...	0:00:26.025	0:00:01.445

Figure 7-50: Server properties – Schedule entry management

See section 7.5.2 page 314 ff for the schedule entry management.

7.3.12 Action templates

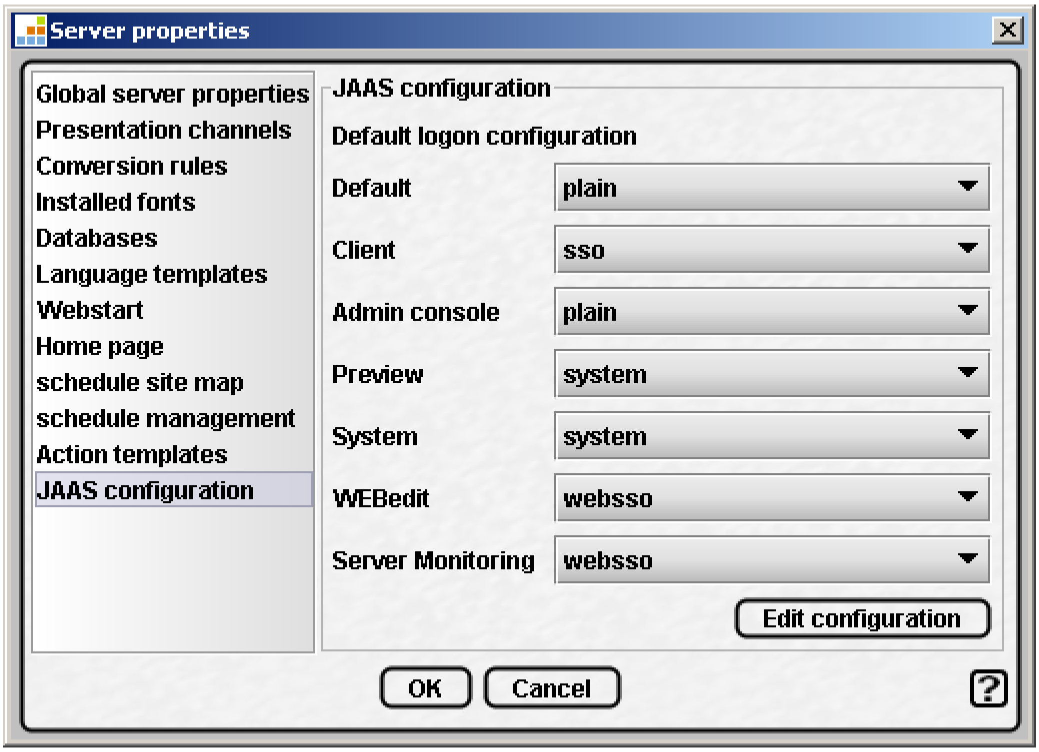


Type	Template type	Name	Uses
Central	Deployments: File	Deployment FIRST...	1
Central	Deployments: File	Deployment FIRST...	1
Central	Deployments: File	Deployment FIRST...	1
Central	Deployments: File	Deployment FIRST...	1
Central	Deployments: File	Deployment FIRST...	0
Central	Deployments: File	Deployment FIRST...	1
Central	Deployments: File	Deployment FIRST...	1

Figure 7-51: Server properties – Action management

See section 7.5.3 page 317 ff for the action templates.

7.3.13 JAAS[25] configuration



JAAS configuration

Default logon configuration

Default: plain

Client: sso

Admin console: plain

Preview: system

System: system

WEBedit: websso

Server Monitoring: websso

Figure 7-52: Server properties – JAAS configuration

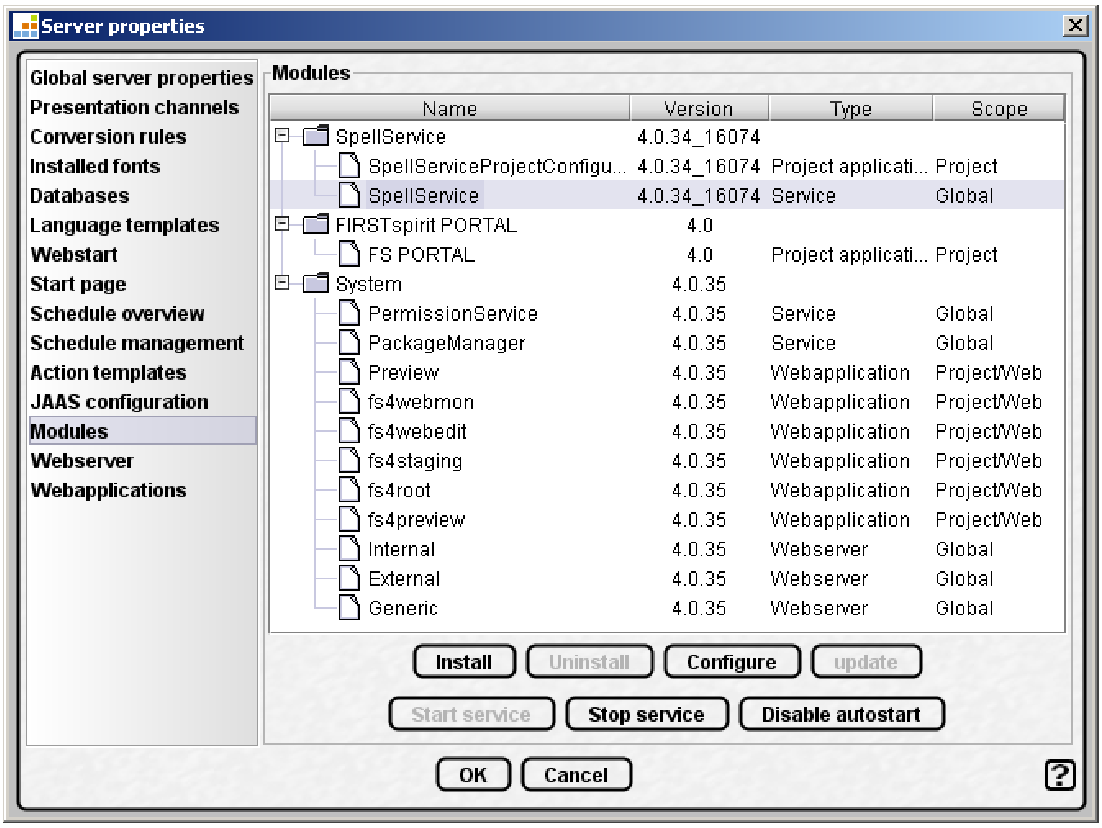
Important configuration settings for the login process can be defined and adapted in this area. The settings defined here are stored in configuration file `fs-jaas.conf` (see section 4.3.4 page 72). If the file is changed via the Server and Project Configuration, the file is rewritten and automatically reloaded.

It is possible to define the respective login process for the various application options (e.g. client login or preview request) via the fold lists. This mapping is saved in configuration file `fs-server.conf` in the JAAS area (see section 4.3.1.6 page 43).

Edit configuration: Click on this button to display an editor for configuration file `fs-jaas.conf`. The file contains an example configuration for all available login modules. Additional modules can be defined via the editor. Confirm via “OK” to save and automatically load the changes in `fs-jaas.conf`.

[25] Java Authentication and Authorization Service

7.3.14 Modules



Name	Version	Type	Scope
SpellService	4.0.34_16074		
SpellServiceProjectConfigu...	4.0.34_16074	Project applicati...	Project
SpellService	4.0.34_16074	Service	Global
FIRSTspirit PORTAL	4.0		
FS PORTAL	4.0	Project applicati...	Project
System	4.0.35		
PermissionService	4.0.35	Service	Global
PackageManager	4.0.35	Service	Global
Preview	4.0.35	Webapplication	ProjectWeb
fs4webmon	4.0.35	Webapplication	ProjectWeb
fs4webedit	4.0.35	Webapplication	ProjectWeb
fs4staging	4.0.35	Webapplication	ProjectWeb
fs4root	4.0.35	Webapplication	ProjectWeb
fs4preview	4.0.35	Webapplication	ProjectWeb
Internal	4.0.35	Webserver	Global
External	4.0.35	Webserver	Global
Generic	4.0.35	Webserver	Global

Figure 7-53: Server properties – Modules

FirstSpirit modules can be installed and configured in this area. The system module with the default components is initially available. The modules are displayed in alphabetical order (by the name of the modules).

> ! *When installing and updating modules which, themselves or because of services which are depending directly or indirectly to these modules, are the basis for data, these data are not available to processes which access these data (generations, clients...) until the restart of these processes.*

Name: Name of the module or of a module component. A 🗀 icon labels the modules in the dialogue. Each module has one or more components which are displayed below the module. A 🗋 icon labels the components. There are various types of components (see "Type").

Version: Version number of the module or component installed on the FirstSpirit
Type: Component type. The following components are distinguished:

- Library: A library is an unconfigured collection of classes, packed in one or more

JAR files. They are available in the clients, scripts and other modules on the FirstSpirit Server after installation.

- Editor: An editor is a combination of GUI and render components. Using the editor it is possible to exend the FirstSpirit-Client by further input possibilities. (Example CMS_INPUT_PERMISSION – input component for defining permissions. This input component works together with the relevant service which loads the group definitions from the server and makes them available.)
- Service: A service is a server component which can be addressed via a public interface consisting of input components or scripts. (Examples for this are the spell check or the service for the permission input component CMS_INPUT_PERMISSION).
- Web application: A web application defines JSP tags and servlets which can be used and called in projects. (Personalisation and search are examples for web applications).
- Web server: A web server component provides functions for installing and uninstalling web applications. (Examples for this are the internal web server control or Tomcat support) (see section 7.3.15 page 238).

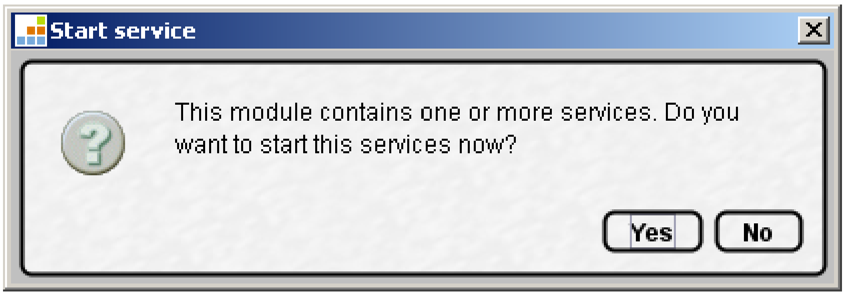
Visible: Components are only available in a certain area after installation. The following areas are distinguished here:

- Global: Global (system-wide) components are available on the server after installation; these components are, therefore, also available in all scripts and components in the FirstSpirit-Clients. (Example: All services; e.g. permission service)
- Project: After installation, project-local components can be added to the desired projects via their project properties (see section 7.4.18 page 291). These components can be subsequently configured. The configuration option depends on the installed component.
- Web: After installation, web-local components can be added to the individual web areas (preview, staging, live) in the desired projects (see section 7.4.18 page 289). The configuration option depends on the installed component. The components can be configured in various ways for the respective projects.

Install: Click on the button to open a file selection dialogue. The fsm file to be installed can be selected here (e.g. the module for the FirstSpirit Personalisation). The successfully installed file is subsequently displayed in the dialogue “Server properties” (see Figure 7-53).

If the installed component contains a service, a dialog is also displayed with which the Autostart option can be configured for the service. If the dialog is confirmed with "Yes" the service is automatically started with each restart of the server (cf.

activate/deactivate autostart):

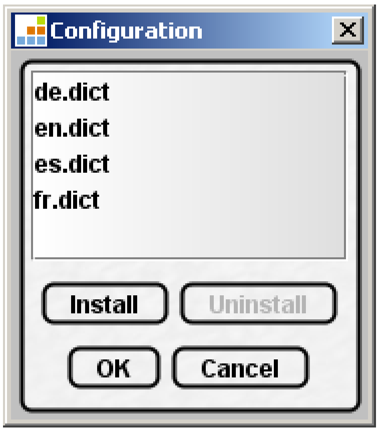


This module contains one or more services. Do you want to start this services now?

Figure 7-54: Install autostart for services

Uninstall: Select the desired module in the overview of installed modules and click on the button to uninstall. The system module cannot be uninstalled.
Only modules which are not used in a project can be uninstalled. When trying to uninstall a module which is still being used, an error message appears with a list of all the projects which currently use this module. Remove the module from these projects first for uninstallation (see section 7.4.18 page 289).

Configure: Click on this button to configure the selected component. The configuration option depends on the selected component (see "Spell Service" example). In addition, from FirstSpirit Version 4.1 permissions can be defined for a module (see chapter 7.3.14.1 page 236):



de.dict
en.dict
es.dict
fr.dict

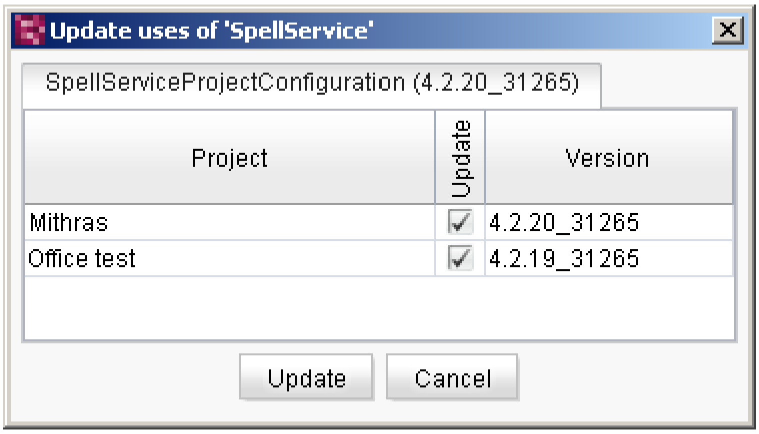
Figure 7-55: Spell Service configuration example

Update (up to and including V4.2R2): If a later version of the module is available, click on the button to update the module on the FirstSpirit Server. The original configuration is retained during the update. However, it might be necessary to adapt the configuration. The new module version is not automatically transferred to the projects. If modules have been added to a project or web area, adaptations within the project might be necessary during an update. Therefore, project updates have to be carried out manually (see section 7.4.17 page 289 and section 7.4.18 page 289).

From FirstSpirit Version 4.1, the uses within the projects can be updated using the "Update uses" button, see below). **From Version 4.2R4** modules can only be installed (button "Install", see above).

> ❗ *After the update of modules with dependencies to other modules with services, these services must restarted manually. (Buttons "Stop service " / "Start service" or via the Server Monitoring, see Chapter 8.6.2.4 page 402.)*

Updating uses (from V4.1): Updating modules is made even more convenient from FirstSpirit Version 4.1. The "Update Uses" button is now available to update a module which contains a project or web application. When the button is clicked, a dialog box opens for updating the project and/or web applications for all projects which have used this application to date.



Update uses of 'SpellService'

SpellServiceProjectConfiguration (4.2.20_31265)

Project	Update	Version
Mithras	☑	4.2.20_31265
Office test	☑	4.2.19_31265

Figure 7-56: Update all uses of a module (from V4.1)

The updating therefore no longer has to be carried out individually for each product via the project properties, but instead can be controlled centrally via the server properties.

Start service: Click on the button to start a service. The service can also be started via Server Monitoring (see section 8.6.2.4 page 402).

> ❗ *After the update of modules with dependencies to other modules with services, these services must restarted manually.*

Stop service: Click on the button to stop a service. A service can also be stopped via Server Monitoring (see section 8.6.2.4 page 402).

Enable/disable autostart: Click on the button to enable or disable an automatic service start. If the option autostart is enabled, the service will be automatically started after a server restart. If the option autostart is disabled, the service must be manually started after a server restart. This option can initially be defined during the installation or update operation of services.

7.3.14.1 Trustworthy modules (from V4.1)

The development of external modules or components for FirstSpirit has been supported since Version 4.0[26]. This concept is enhanced in **FirstSpirit Version 4.1** to include security aspects.

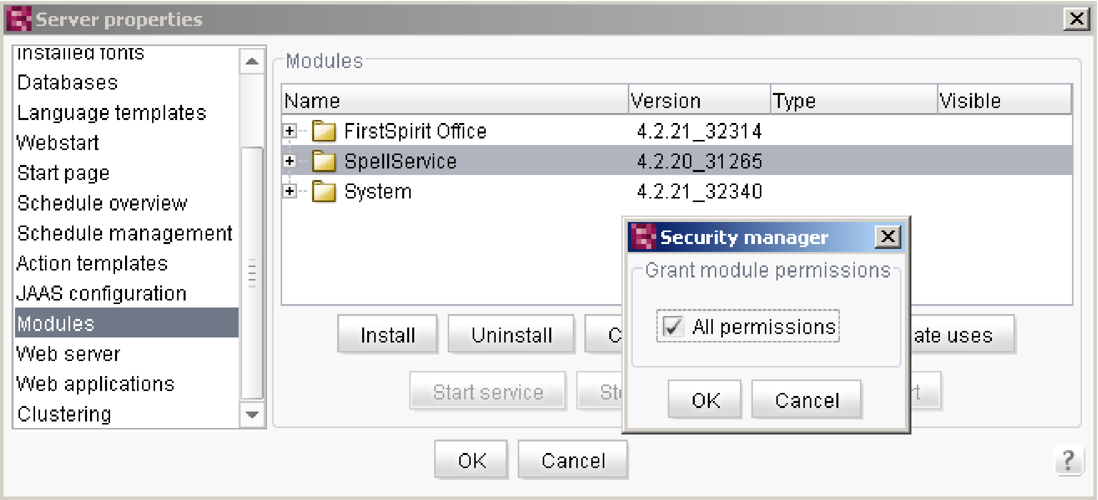
FirstSpirit JavaClient and the FirstSpirit server and project configuration are implemented via a `jnlp` file, i.e. via Java Web Start. This results in restrictions in the use of several functions for modules not signed by e-Spirit or classes in the Jar archives. Java programs usually run in a "sandbox". This means they do not have full access to the computer (and its resources), on which they are run. Access to local resources such as files, clipboard, network, etc. is provided via a security manager.

The internal FirstSpirit modules are signed with the "e-Spirit AG" key. This in turn is part of the internal FirstSpirit security policy. Furthermore, the key is confirmed by a root authority which in turn is known to the Java certificate manager.

External components or modules which access security-relevant functions can be conveniently configured in Version 4.1 via the FirstSpirit server and project configuration. Each installed module (with the exception of the FirstSpirit system modules) can be optionally equipped with privileges or rights to local system resources. For example, it is possible to trust a module which performs security-relevant operations, e.g. access to the clipboard (java.awt.AWTPermission ClipboardAccess). This module can be assigned permissions for performing the operations. This takes place internally via the FirstSpirit security manager/class loader.

The configuration interface for setting the module permissions in the FirstSpirit server and project configuration is as follows:

[26] For concept and development see “Entwicklerhandbuch für Komponenten“ for FirstSpirit Version 4.0



Name	Version	Type	Visible
FirstSpirit Office	4.2.21_32314		
SpellService	4.2.20_31265		
System	4.2.21_32340		

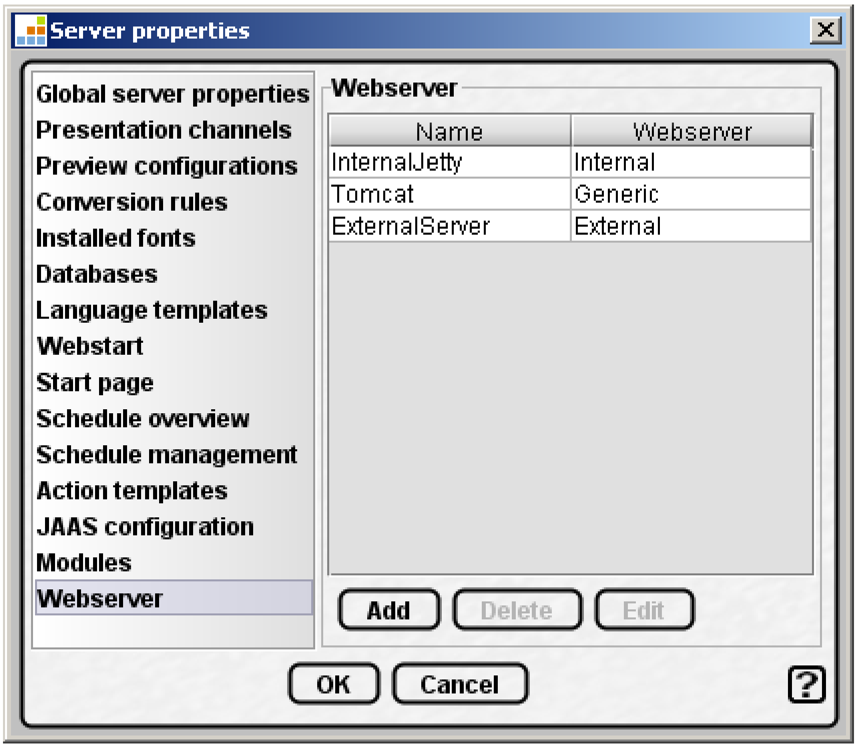
☑ All permissions

Figure 7-57: Security manager – Assign all rights for this module

If an external component or module is classified as being trustworthy, it is not possible to ensure that the access protection mechanisms of FirstSpirit are fully effective. Any malfunctions which occur can no longer be clearly assigned so that fault diagnosis is made difficult or impossible. Therefore, within the scope of FirstSpirit product maintenance, system configurations with external components or modules classified as trustworthy are not accepted.

For further information, see "FirstSpirit Developer Manual for Components".

7.3.15 Web server



Name	Webserver
InternalJetty	Internal
Tomcat	Generic
ExternalServer	External

Figure 7-58: Server properties – Web server

The web server control for installing and uninstalling within the project areas can be inserted in this area (see section 7.4.17 page 289).

The control for the internal web server Jetty is available by default. The respective entry is displayed in the table. This web server control cannot be changed or deleted and is automatically available after FirstSpirit installation.

In addition to this default entry, it is possible to add and configure any generic or external web server control in this area.

Name: Unique name under which the entity of the web server has been added.

Web server: Three different web server types are available in the default module "System" (see "Module" dialogue in section 7.3.14 page 232):

- Internal (internal web server): A control (installation or uninstallation of web applications) is available for the internal Jetty after FirstSpirit installation.
- External (external web server): This is an external web server not supported by FirstSpirit. Therefore, installing or uninstalling of web applications on the web server as well as each configuration setting have to be carried out manually.
- Generic (local web server): FirstSpirit enables the simple connection of web

servers as long as it is possible to implement the control (installation or uninstallation of web applications) via a BeanShell script (see section 7.3.15.2 page 240) (for further information on supported web servers see the FirstSpirit V4.x Technical Datasheet).

Add Click on the button to add a new generic web server entity or a new external web server entity to the FirstSpirit Server:

- Internal web server: The "InternalJetty" is available by default, i.e. it does not have to be added.
- Generic web server (see section 7.3.15.1 page 239)
- External web server (see section 7.3.15.3 page 242)

Delete Click on the button to remove an added web server entity. The default entity "InternalJetty" cannot be removed.

Configure Click on the button to open the configuration dialogue for the web server.

- Internal web server: This configuration cannot be edited.
- Generic web server: If it is a generic web server, the required functions for installing or uninstalling can be configured here (see section 7.3.15.2 page 240).
- External web server: A configuration for external web servers is not supported and has to be carried out manually (see section 7.3.15.4 page 242)

7.3.15.1 Add a generic web server

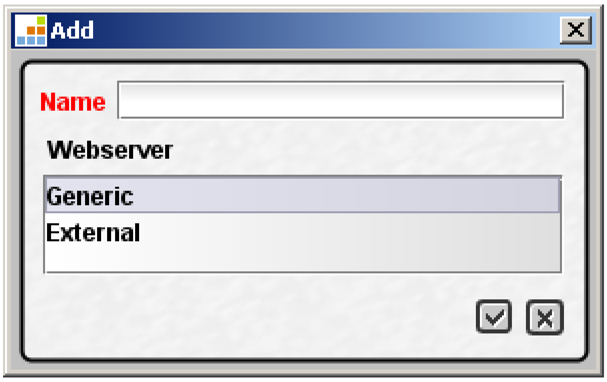


Figure 7-59: Add a generic web server

Name: Assign a unique name for the respective web server entity here.

Web server: A new generic web server entity can be added by selecting "Generic".

! *After adding a new generic web server entity, the installation and uninstallation function has to be implemented (see section 7.3.15.2 page 240).*

7.3.15.2 Configure a generic web server

The following functions are required for embedding a new web server on the FirstSpirit Server:

1. Install
2. Check installation state
3. Uninstall

Click on "Configure" to open the following dialogue:

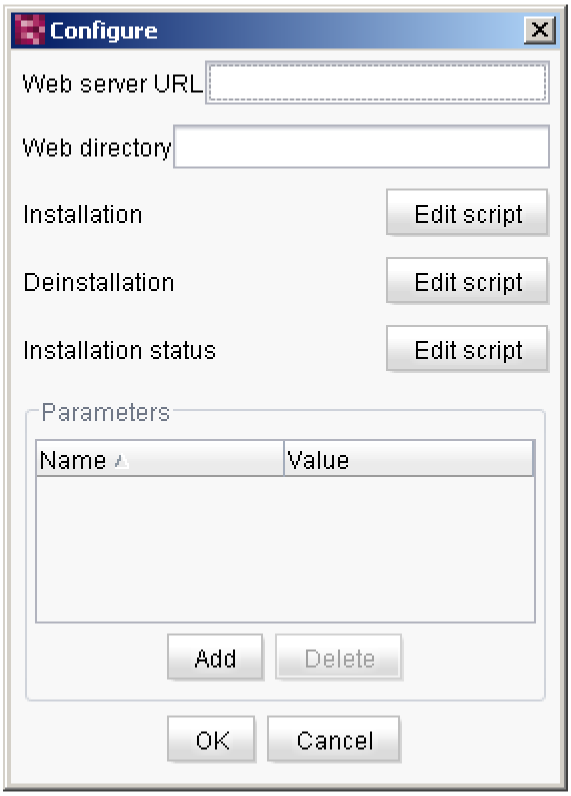


Figure 7-60: Configure web server functions

Web server URL: The URL of the new web server can be entered in this field. The URL is, e.g., required to adapt the links for the FirstSpirit web applications on the start page.

Web directory: Specifying the path to the web directory of the generic web server for use of the FirstSpirit web applications (e.g. fs4staging). If the generic web server is used for example for the web application fs4staging, the project files are generated into the specified directory of the generic web server. The path is also required for executing the configured scripts (for example for installation or deinstallation).

Edit script: All functions can be realised via a BeanShell script. Click on “Edit script” (in front of the respective function) to open the script dialogue.

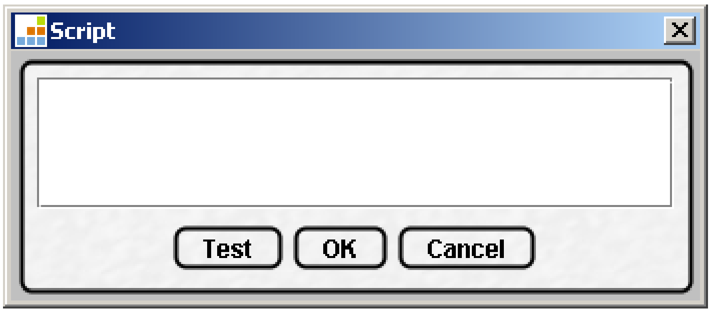


Figure 7-61: Add or edit a BeanShell script

The desired functions can be executed script-controlled for each web server supported by FirstSpirit. The required parameters for the scripts can be added in the “Parameters” area. Click on the “Test” button to test the function prior to saving. After saving, the function is available in the “Web applications” or “Web components” area:

- Server properties / Web applications (see section 7.3.16 page 243
- Project properties / Web components (see section 7.4.18 page 289).

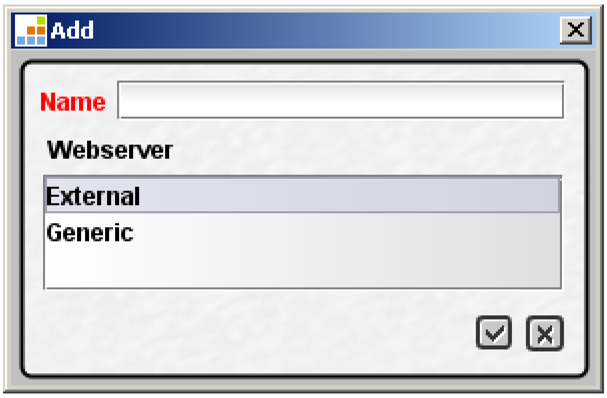
For executing the scripts the web directory can be specified which contains the generic web server (see “Web directory“).

Add parameters: Parameters which will be subsequently available in all script contexts (from installation to uninstallation) can be added in the “Parameters” area (see Figure 7-60) (example: Path specification).

Delete parameters: Click on the button to delete a highlighted parameter from the list.

> ! *Scripts may no longer work or have to be adapted after deleting parameters.*

7.3.15.3 Add an external web server



Add

Name

Webserver

External

Generic

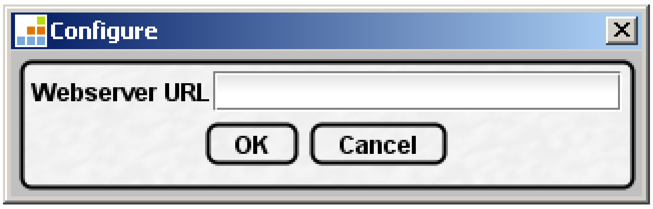
Figure 7-62: Add an external web server

Name: Assign a unique name for the respective web server entity here.

Web server: A new external web server entity can be added by selecting the "External" entry.

After adding a new external web server entity, it is possible to specify the URL of the web server (see section 7.3.15.4 page 242). Further support (installation or uninstallation) is not offered and has to be carried out manually.

7.3.15.4 Configure an external web server



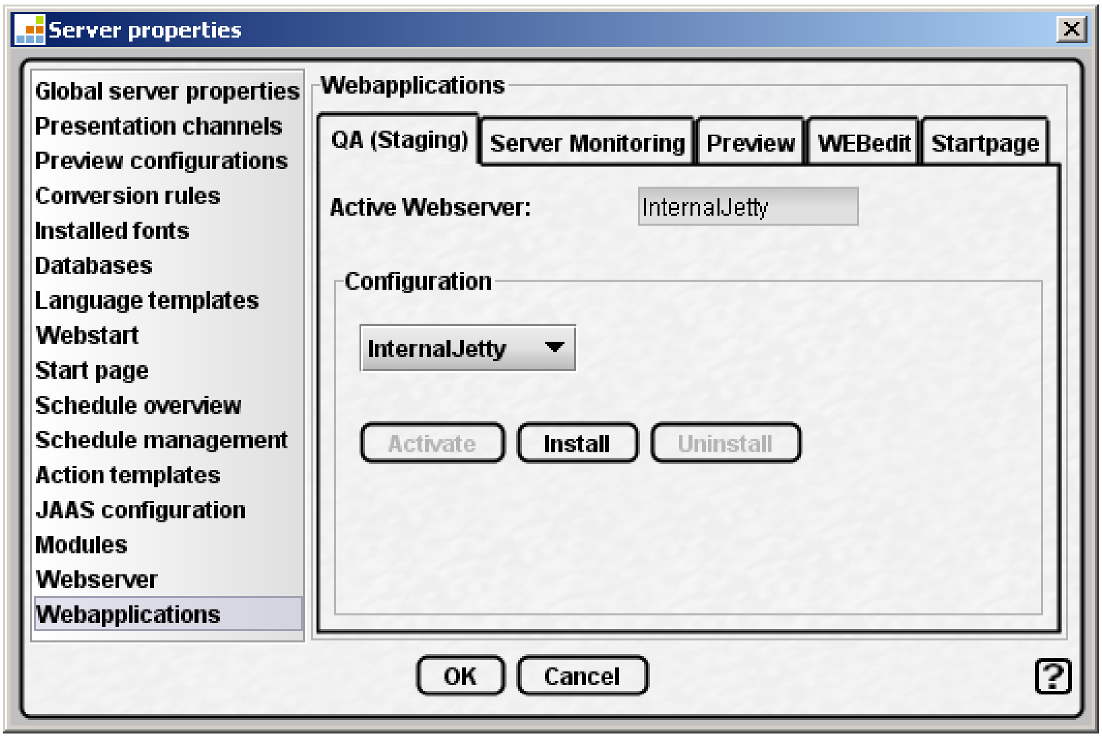
Configure

Webserver URL

Figure 7-63: Configure an external web server

Web server URL: The URL of the new web server can be entered in this field. For example, the URL is required to adapt the links for the FirstSpirit web applications on the start page.

7.3.16 Web applications



Active Webserver: InternalJetty

InternalJetty

Figure 7-64: Server properties – Web applications

FirstSpirit web applications can be configured in this area. From Version 4.2R4 and higher, the server properties can be used to define and configure the customer's own web applications, which are then available in all projects on the server ("global"). In this way, e.g. the customer's own web applications can be installed for the FirstSpirit AppCenter (see *FirstSpirit Release Notes Version 4.2R4*, Chapter 3 "The FirstSpirit AppCenter").

7.3.16.1 Installing global web applications

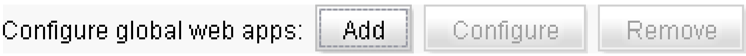
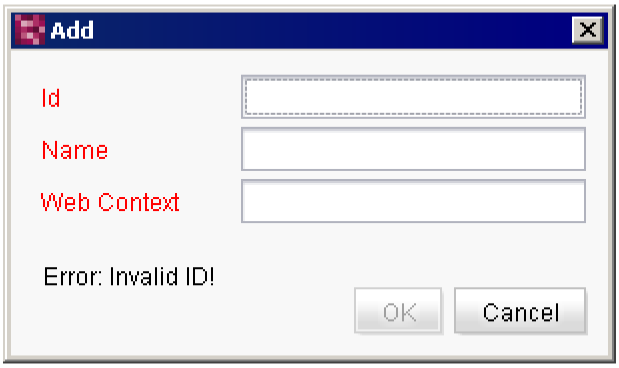


Figure 7-65: Configuring global web applications

The "Add" button can be used to install a new web application on the server:



Add

Id

Name

Web Context

Error: Invalid ID!

Figure 7-66: Add new web application

Id: A unique identifier for the web application must be entered here. A subdirectory with this identifier will be created on the server for the web applications. Only upper case, lower case letters, numbers and underscores may be used. This identifier can no longer be changed after it has been saved.
Using the FirstSpirit Developer-API you can find out the to a global web application by means of the Id of a web application via the interface `de.espirit.firstspirit.agency.LegacyModuleAgent`.

Name: A name must be given here, which is used as the display name. This can be changed later if and when necessary.

Web Context: A so-called context name must be entered here, which forms part of the URL to the web application. It may not be the same as the name of the existing standard FirstSpirit web applications (i.e. *fs4root*, *fs4preview*, *fs4staging*, *fs4webedit*, *fs4webmon)*.

7.3.16.2 Configuration of web applications

Each web application is displayed and configured on an own tab. The default web applications are:

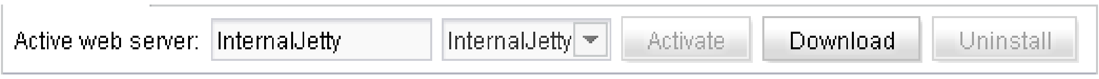
Start page / Preview: Configuration for the FirstSpirit start page and preview (see chapter 6 page 168).

QA (Staging): Configuration for the generated project contents QA (Staging).

WEBedit: Configuration for the editorial system WebEdit.

Server Monitoring: Configuration for FirstSpirit Server Monitoring (see chapter 8 page 376)

- For each web application a web server can be set:


Active web server: InternalJetty InternalJetty

All web servers which have been configured in the “Web server” area can be selected (cf. Chapter 7.3.15 page 238).

Depending on the web server type, various steps are required for changing a web server:

▪ Internal web server	(see section 7.3.16.3 page 246)
▪ Generic web server	(see section 7.3.16.4 page 246)
▪ External web server	(see section 7.3.16.5 page 247)

Activate Click on the button to change the configuration of the web application to the selected web server. This web server is subsequently displayed as an active web server for the application. The links to the web applications (e.g. on the start page) are changed automatically to the new web server.

Install Click the button to install the web application on the selected web server. If the button is deactivated, the web application has already been installed. If the selected web server is an external web server or a generic web server (without the necessary script function), the "Download" button is displayed instead (see chapter 7.3.16.5 page 247).

Download Click the button to download a WAR file of the application, which must then be manually installed on the web server. The button is only displayed for the configuration of external web servers or the configuration of generic web servers (without the required script functionality).

Uninstall Click the button to uninstall the web application on the selected web server. If the button is deactivated, the web application has not yet been installed. If a web application is uninstalled, the corresponding entry is also removed from the configuration file fs-server.conf (in FirstSpirit Version 4.2.437 and higher – in earlier versions the entry is commented out).

- Web components available on the server can be added, removed, configured and updated for each web application:
  Add Delete Configure Update
- The `web.xml` file can be manually edited for each web application: web.xml

The web components can be grouped together to form a web application and installed on the web server or downloaded as a WAR file.

The function is analogous to the that of web components in the project properties.

See also Chapter 7.4.18 page 291.

7.3.16.3 Configure an internal web server for a web application

The control for the internal web server Jetty is available by default and cannot be changed. After FirstSpirit installation, the internal web server is activated for all FirstSpirit web applications.

If a different web server (e.g. Tomcat) has been activated for a FirstSpirit web application, the configuration can be reset to the internal web server as follows:

1. Select “InternalJetty” in the combo box.
2. The Install button is activated. Click on the button to install the web application on the internal web server.
3. The Activate button is activated after installation. Click on the button to change the web application configuration to the internal web server which is subsequently displayed as the active web server for the application. The links to the web applications (e.g. on the start page) are automatically changed to the new web server.
4. Click on OK to confirm and save all configuration changes.

7.3.16.4 Configure a generic web server for a web application

A generic web server can only be selected if a respective web server entity has been added to the server (see section 7.3.15.1 page 239). A generic web server control is not available by default, but can be realised via scripts (see section 7.3.15.2 page 240). If these scripts are not available, please proceed as described for the external web server (see section 7.3.16.5).

Carry out the following steps if a generic web server (e.g. Tomcat) is to be activated for a FirstSpirit web application:

1. Select the entry for the desired generic web server in the combo box.
2. The Install button is activated. If the respective function has been provided via a script, click on the button to install the web application on the generic web server.
3. The Activate button is activated after installation. Click on the button to change the web application configuration to the generic web server which is subsequently displayed as the active web server for the application. The links to

the web application (e.g. on the start page) are automatically changed to the new web server. The URL specified under “Web server URL” is used for this task (see Figure 7-60).

4. Click on OK to confirm and save all configuration changes.

7.3.16.5 Configure an external web server for a web application

An external web server can only be selected if a respective web server entity has been added to the server (see section 7.3.15.3 page 242). FirstSpirit does not support an external web server control. This has to be carried out manually (see section 7.3.15.4 page 242).

Carry out the following steps if an external web server is to be activated for a FirstSpirit web application:

1. Select the entry for the desired external web server in the combo box.
2. The “Download” button is activated. Click on the button to download a WAR file of the application.
3. The WAR file has to be installed manually on the external web server. Installation either takes place manually via the administration user interface of the external web server or automatically from the file system of the web server. An independent virtual server should be created as start URL to ensure that the FirstSpirit start page can be defined as root application in the web application `/fs4root` and can, e.g., be reached directly via http://fs4.yourdomain.net. An URL with a prefix, e.g. http://appserver.yourdomain.net/fs4, is also possible.
   The WAR file has to be re-installed after each FirstSpirit Server update!
4. After installation, click on Activate to change the web application configuration to the external web server. The external web server is subsequently displayed as the active web server for the application. The links to the web application (e.g. on the start page) are automatically changed to the new web server. The URL specified under “Web server URL” is used for this task (see Figure 7-63).
5. Click on OK to confirm and save all configuration changes.

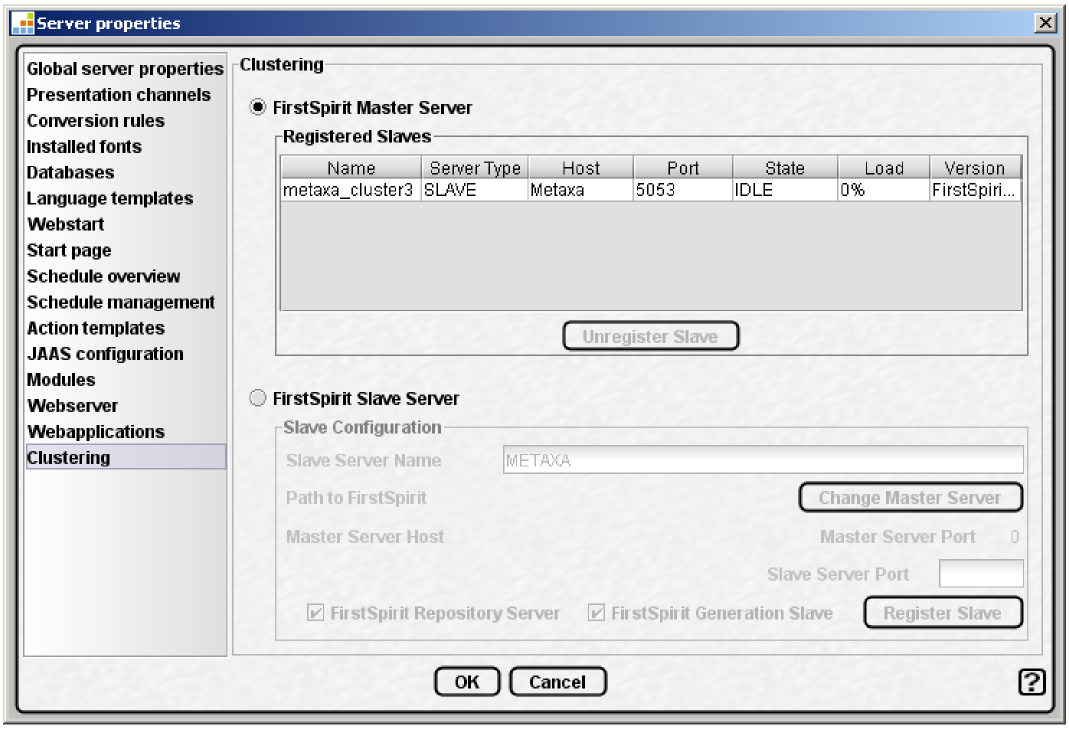
7.3.17 Clustering (from V4.1)

> *This functionality is released with FirstSpirit Version 4.1.*

In our case, clustering is the load distribution for the generation of projects to other FirstSpirit servers ("horizontal scalability").

For details of the "Clustering" concept and the description of the architecture, see chapter 7.6, page 351.

The configured FirstSpirit server within a cluster are listed in the FirstSpirit server and project configuration, in the "Clustering" area within the server properties. Under Windows this area serves also for the first installation of the clusters. Under Unix the cluster is installed and configure only by means of the configuration files of the FirstSpirit Server.



● FirstSpirit Master Server

Name	Server Type	Host	Port	State	Load	Version
metaxa_cluster3	SLAVE	Metaxa	5053	IDLE	0%	FirstSpiri...

○ FirstSpirit Slave Server

Slave Server Name: METAXA

Master Server Port: 0

☑ FirstSpirit Repository Server ☑ FirstSpirit Generation Slave

Figure 7-67: Server properties – Clustering

- Configuration of the Master Server under Unix: Chapter 7.3.17.1
- Configuration of the Slave Server under Unix: Chapter 7.3.17.2
- Configuration of the Master Server under Windows: Chapter 7.3.17.3

- Configuration of the Slave Server under Windows: Chapter 7.3.17.4

7.3.17.1 Configuring FirstSpirit Master server under Unix

First, a FirstSpirit server is installed as described in the installation manual By way of example, `/opt/firstspirit4` is assumed to be the installation path and `fs4host` the host name. Following installation, the FirstSpirit server is shut down in order to define the FirstSpirit server as the master server with the configuration changes described in the following and to enable data to be sent to the FirstSpirit slave server.

The master server contains the complete functional scope of a normal FirstSpirit server and should be designed to be fail-safe with respect to the hardware, as all FirstSpirit clients use the master server as the terminal point of the network connections. The slave servers alone cannot be used independently. Conversely however, even if slave servers have failed, all FirstSpirit functions are available through the master server alone without configuration changes during operation. The generation schedules are then automatically executed on the master instead of on the slave, which only results in a higher workload for the master.

The following line must be added in the file `/opt/firstspirit4/conf/fs-server.conf`:

```
HOST=fs4host
```

A distributed file system is necessary to transfer data to the slave server and to write log files from the slave servers to the master server. NFS can be used for this under Unix. The NFS server should be run on the same system as the FirstSpirit master server, so that the master server can locally write into the file system. Solaris is used in this example. The NFS configuration can be adopted accordingly for other UNIX systems. The differentiation between directories with read only and read/write access increases operational reliability and data throughput, as only the log files are distributed writable via NFS. The central repository data of the master server is distributed with read access only.

The following lines must be added into the `/opt/dfs/dfstab` file, in order to release several selected directories of the master server via NFS:

```
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
```

If increased security requirements exist, the NFS release should be limited to exclusive access of the FirstSpirit slaves by means of the authentication method supported by the operating system (IP address, Kerberos, …).

The NFS server must now be informed of the changed configuration, e.g. using `shareall`.

The FirstSpirit master server can now be started using `fs4 start` (or `svcadm enable fs4`). The availability and workload/capacity utilisation of the slave servers can be monitored on the FirstSpirit homepage using FirstSpirit server monitoring in the "Clustering" area (see Chapter 8.6.6 page 415) .

The following log files of the master are used to analyse errors if problems occur during and after configuration:

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

7.3.17.2 Configuring FirstSpirit Slave servers under Unix

First, a FirstSpirit server is installed as described in the installation manual It is necessary to ensure that the Unix User ID used for the FirstSpirit installation is the same as the one used on the master server. Otherwise the release and writability of the common NFS directories are more complicated. By way of example, `/slave/firstspirit4` is assumed to be the installation path and `fs4slave` the host name. Following installation, the FirstSpirit server is shut down in order to define the FirstSpirit server as the slave server with the configuration changes described in the following and to enable data to be sent to the FirstSpirit master server.

Several JAR files and start scripts of the master server are used directly via NFS to simplify subsequent updating to a new FirstSpirit version. After updating the master servers, the slave servers therefore only have to be restarted, without having to manually exchange files.

For the configuration, the existing `wrapper.mainclass` and `#include` parameters are changed in the `/slave/firstspirit4/conf/fs-wrapper.conf` file:

```
wrapper.java.mainclass=de.espirit.firstspirit.server.ClusterHost

#include ../conf/fs-wrapper-license.slave.conf
```

and the following parameters are to be added, with continuation of the consecutive numbering of the existing entries named `wrapper.java.additional`:

```
wrapper.java.additional.12=-Dcmsroot=/import/fs4master

wrapper.java.additional.13=-Dnode=Generierungs-Slave

wrapper.java.additional.14=-DinitialPort=1088
```

The port number given for `initialPort` defines the TCP port which the slave server users to receive the master server's control data.

The `/slave/firstspirit4/conf/fs-server.conf` file automatically created during the FirstSpirit default installation is not required for the slave server and should be deleted:

```
rm /slave/firstspirit4/conf/fs-server.conf
```

The configuration of the NFS client under Solaris is shown here as an example. This configuration can be adopted accordingly under other Unix systems.

First, the mount points on which the master files are provided are created:

```
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
```

The following entries are now necessary in the `/etc/vfstab` file in order to use the directories provided by the master server:

```
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
```

The NFS client must now be informed of the changed configuration, e.g. using `mount –F nfs –a`.

The FirstSpirit slave server can now be started using `fs4 start` (or `svcadm enable fs4`). The availability and workload/capacity utilisation of the slave servers can be monitored on the FirstSpirit homepage using FirstSpirit server monitoring in the "Clustering" area (see Chapter 8.6.6 page 415) .

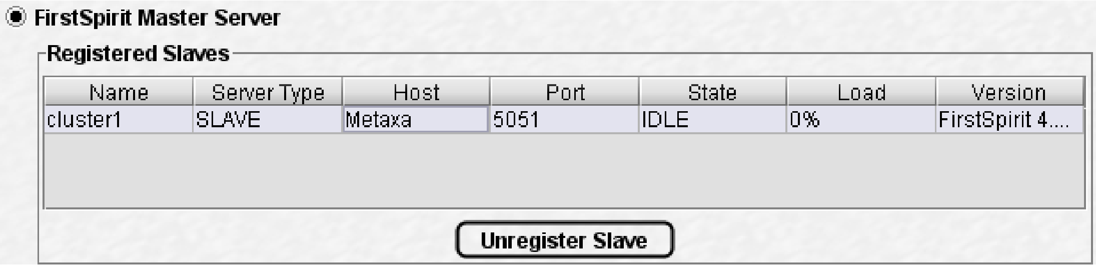
The following log files of the slave are used to analyse errors if problems occur during and after configuration:

```
/slave/firstspirit4/log/fs-wrapper.log
/import/fs4master/log/fs4slave/fs-server.log
```

! *"Execute on cluster" must be enabled under Properties in the respective schedule configuration so that a slave server is used for the project generation (see Chapter 7.6.5 page 354).*

7.3.17.3 Configuring FirstSpirit Master server under Window

If the "FirstSpirit Master Server" radio button is selected (cf. Figure 7-67), the server is the FirstSpirit Master Server. This centrally manages all FirstSpirit projects and distributes the tasks, where possible, to other FirstSpirit servers. The available servers (e.g. Generation slaves) are displayed in the "Registered Slaves" area (see Figure 7-68):



● FirstSpirit Master Server

Registered Slaves

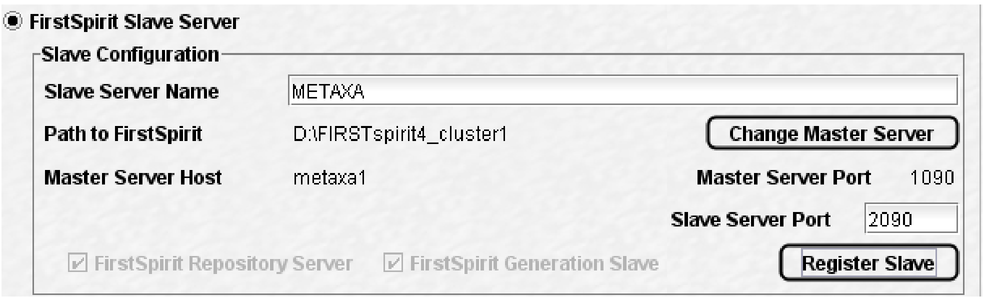
Name	Server Type	Host	Port	State	Load	Version
cluster1	SLAVE	Metaxa	5051	IDLE	0%	FirstSpirit 4....

Figure 7-68: Configuration as FirstSpirit Master server

Unregister Slave Click the button to remove a "Registered Slave" from the selection again.

7.3.17.4 Configuring FirstSpirit Slave servers under Windows

The FirstSpirit Server can be defined as a so-called "Slave Server" in the bottom part of the dialog. To this end, a FirstSpirit Master server must be disclosed in the "Slave Configuration" area. Following successful configuration the FirstSpirit Server is available as the member of a cluster.



● FirstSpirit Slave Server

Slave Configuration

Slave Server Name: METAXA

Path to FirstSpirit: D:\FIRSTspirit4_cluster1

Master Server Host: metaxa1

Master Server Port: 1090

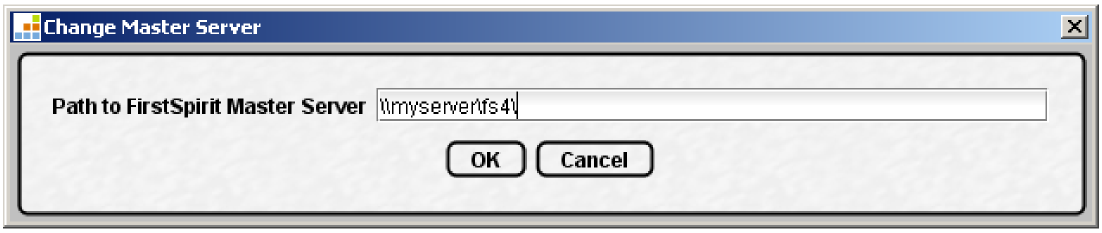
Slave Server Port: 2090

☑ FirstSpirit Repository Server ☑ FirstSpirit Generation Slave

Figure 7-69: Configuration as FirstSpirit Slave server

Slave server name: Name of the FirstSpirit slave.

Path to FirstSpirit: The Change Master Server button can be used to specify the path to the FirstSpirit Master server. After clicking the button a dialog opens in which the path can be entered manually:



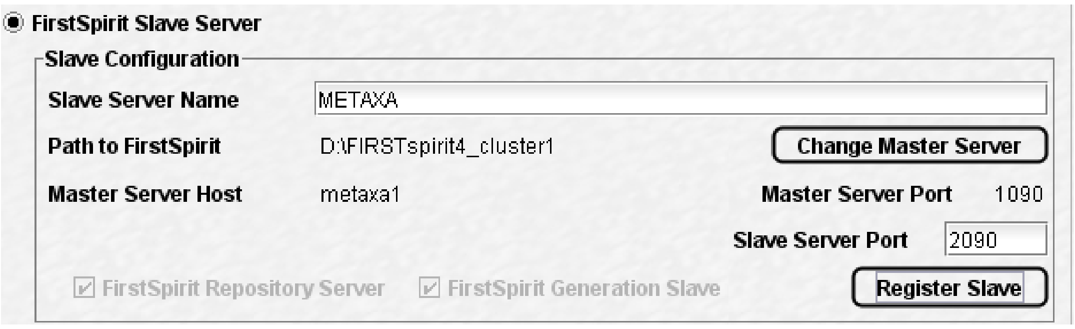
Change Master Server

Path to FirstSpirit Master Server: \\myserver\fs4\

Figure 7-70: Path to the FirstSpirit-Master-Server

After clicking "OK" the system checks whether a further FirstSpirit server has been installed. This FirstSpirit server (Master server) may not be the same as the server currently being configured ("Slave server"). If a further FirstSpirit server has been identified the "Master Server Host" and "Master Server Port" parameters are automatically filled. The parameters of the `fs-server.conf` configuration file (of the FirstSpirit Master server) are used for this.

The dialog for a correctly configured slave server looks like this:



● FirstSpirit Slave Server

Slave Configuration

Slave Server Name: METAXA

Path to FirstSpirit: D:\FIRSTspirit4_cluster1

Master Server Host: metaxa1

Master Server Port: 1090

Slave Server Port: 2090

☑ FirstSpirit Repository Server ☑ FirstSpirit Generation Slave

Figure 7-71: Configuration as FirstSpirit Slave server

The "Change Master Server" button can be used to change the master server set here again at a later date.

Master Server Host: Host name of the FirstSpirit master server. The field is automatically filled on specifying the path to the FirstSpirit server.

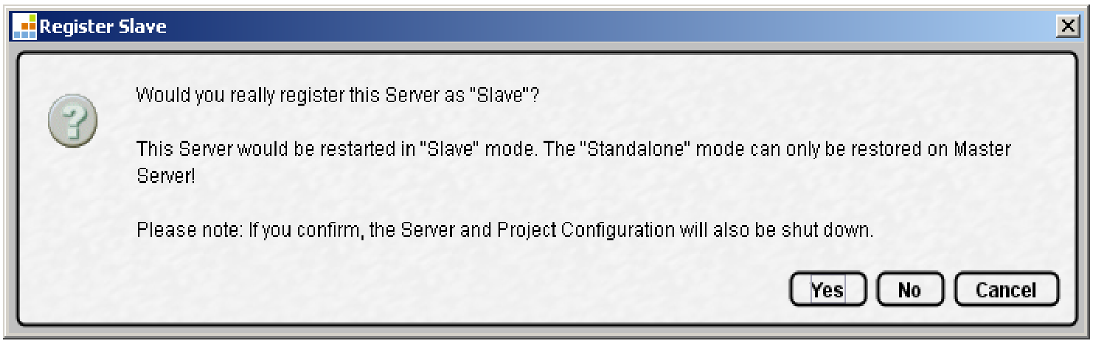
Master Server Port: Port number of the FirstSpirit master server. The field is automatically filled on specifying the path to the FirstSpirit server.

FirstSpirit Read-Only Repository Server / FirstSpirit Generation Slave: The checkboxes are automatically activated. The FirstSpirit server is added to the cluster as a generation server. A generation server contains an RORS[27]. The "FirstSpirit Read-Only Repository Server" checkbox is therefore also automatically activated.

Slave Server Port: In this field must be entered a free port for the slave server.

When this button is clicked a confirmation prompt opens:

[27] ReadOnly-Repository-Server (see chapter 7.6.2 page 352)



Register Slave

Would you really register this Server as "Slave"?

This Server would be restarted in "Slave" mode. The "Standalone" mode can only be restored on Master Server!

Please note: If you confirm, the Server and Project Configuration will also be shut down.

Figure 7-72: Register Slave

If the dialog is confirmed with "Yes" the configuration settings are transferred into the corresponding configuration files (fs-server.conf, fs-wrapper.conf). The server is then shut down and started in "Slave" mode. The application for the Server and Project configuration of the new slave server will be closed automatically too.

An already registered slave can be deregistered via the server properties of the FirstSpirit Master server only.

7.4 Project properties

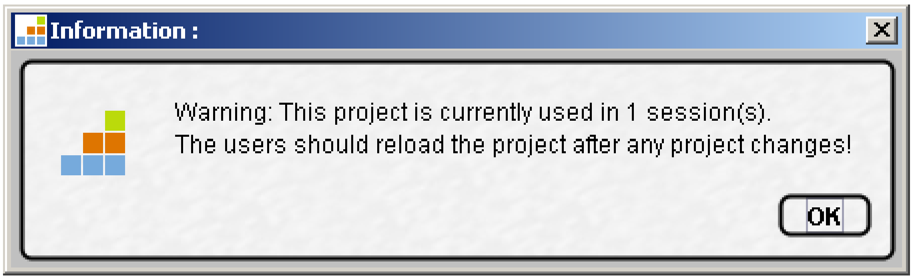
Double-click on the desired project in the project list to open the project properties. They can also be opened via menu item “Project”/”Properties” in the menu bar (see section 7.2.3.7 page 198) or via the “Change properties” button.

Click on the button to open the project properties of the project highlighted in the project selection list.

The project selection list only displays the projects for which the logged in user has the required permissions. A server administrator has access to all server projects while a project administrator only has access to projects to which he/she has been assigned as administrator (for further information see section 7.1 page 184 and section 7.4.2 page 257).

The “Edit project” dialogue window appears after calling (see Figure 7-74). The properties of the selected project can be edited via the menu items on the left. The respective settings for each menu item can be carried out on the right.

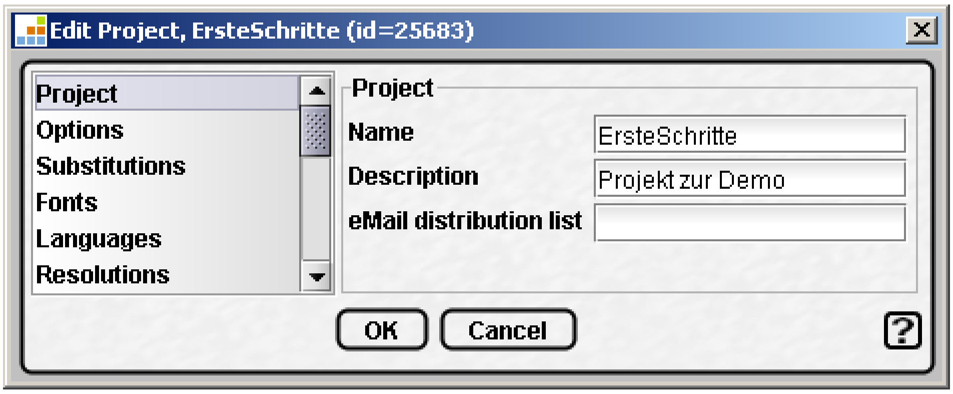
The project settings of a project can only be carried out by one user at a time. An error message appears if a second user tries to open the project settings for a specific project:



Warning: This project is currently used in 1 session(s).
The users should reload the project after any project changes!

Figure 7-73: Error message – Simultaneous editing

7.4.1 Project



Project

Name: ErsteSchritte

Description: Projekt zur Demo

eMail distribution list:

Figure 7-74: Project properties – Project

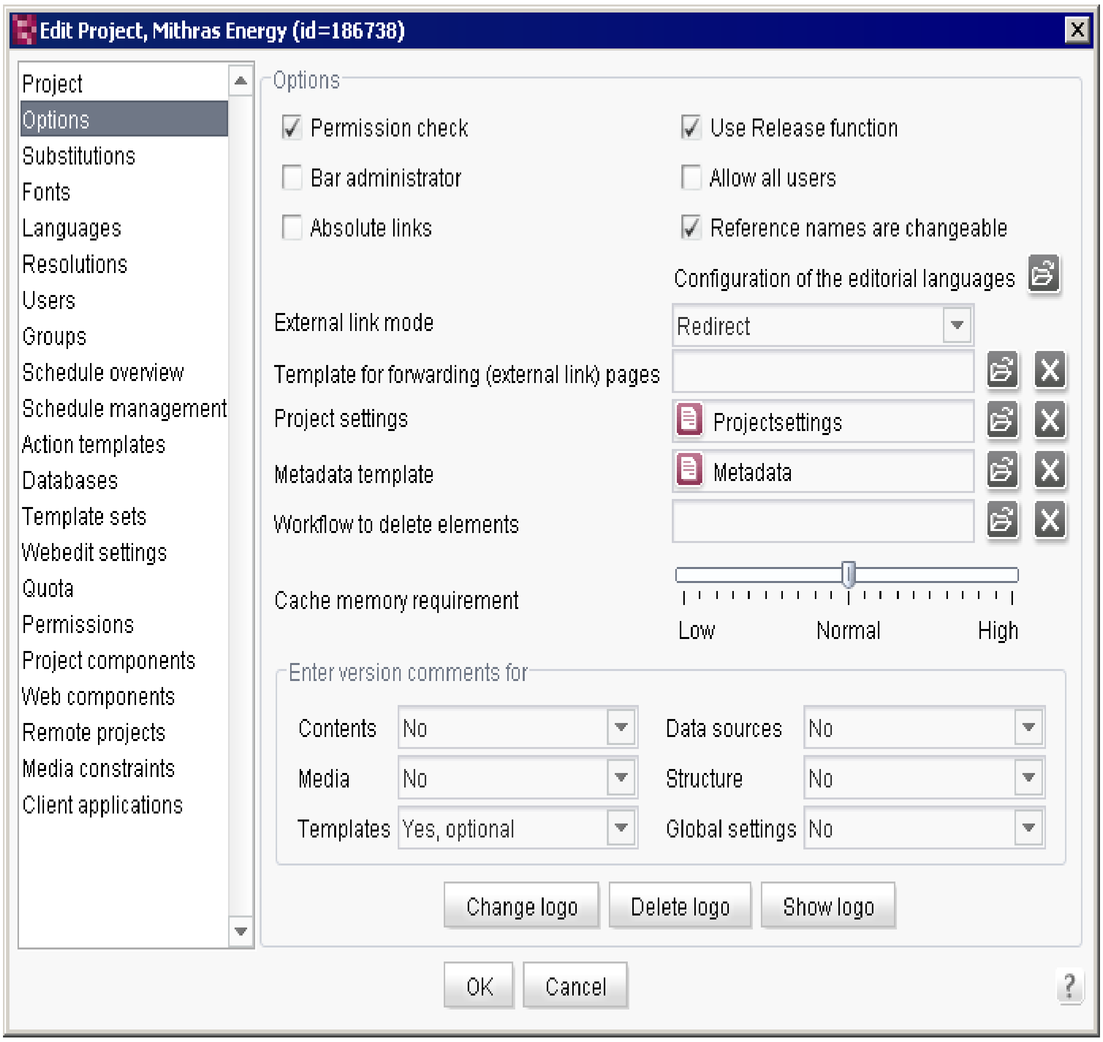
Name: The unique project name, which can also be changed, is displayed in this field. If the project name is changed, the "OK" button remains inactive until a unique name has been assigned.

Description: The project description, which can also be changed, is displayed in this field. The description does not have to be unique. Nevertheless, this field is a required field.

eMail distribution list: After each generation, an email with state information is sent to the specified addresses. Multiple addresses must be separated by a semicolon.

7.4.2 Options

Some parts of this functionality are only released with FirstSpirit Version 4.1. For this reason, screenshots are shown in the new Look & Feel "LightGray". The layout in the Look & Feel "Classic" can differ slightly.



Edit Project, Mithras Energy (id=186738)

- Project
- Options
- Substitutions
- Fonts
- Languages
- Resolutions
- Users
- Groups
- Schedule overview
- Schedule management
- Action templates
- Databases
- Template sets
- Webedit settings
- Quota
- Permissions
- Project components
- Web components
- Remote projects
- Media constraints
- Client applications

Options

☑ Permission check ☑ Use Release function
☐ Bar administrator ☐ Allow all users
☐ Absolute links ☑ Reference names are changeable

Configuration of the editorial languages

External link mode: Redirect
Template for forwarding (external link) pages:
Project settings: Projectsettings
Metadata template: Metadata
Workflow to delete elements:
Cache memory requirement: Normal

Enter version comments for

Contents	No	Data sources	No
Media	No	Structure	No
Templates	Yes, optional	Global settings	No

Figure 7-75: Project properties – Options (Version 4.2R4)

Permission check: Use this option to define whether a permission check is to be used for the project. If the box is *unchecked*, users are automatically granted full access permissions to the project. They can then access menu functions or objects to which usually only project administrators have access. If the box is *checked*, the users' permissions are analysed. A user who is not a project administrator cannot, e.g., generate a project. **From FirstSpirit Version 4.2**, when creating a new project this option is active by default, before FirstSpirit Version 4.2 it was inactive.

Use Release function: If the box is *checked*, FirstSpirit distinguishes between the released and the current project status. As soon as the project is changed, the change has to be released by a user with the respective permissions (e.g. the “chief editor”). An unreleased project status, e.g. a page from the Content-Store, is not transferred during the next generation (the last released state of the page is considered instead.) If the box is *unchecked*, each project change is automatically transferred to the release status (“auto release”) and is thus immediately effective.

If this option is *activated anew*, the following query will appear: “Do you wish to create a release version of all stores?“ If you choose “Yes“ a release version of all stores, from FirstSpirit Version 4.1 on except for the Template Store, is created. This process can take some time. If you choose “No” the release function will be activated without creating a release version of the stores.

From FirstSpirit Version 4.2, when creating a new project this option is active by default, before FirstSpirit Version 4.2 it was inactive.

! *Unchecking this option is only recommended for smaller projects and has negative effects on the overall system performance, since each change results in an automatic release.*

MultiViews path: This option supports the “Apache multiview” function for generating multilingual pages. The project files are provided with a language extension, e.g. `index.html.de` and `index.html.en`. This URL is automatically analysed by the web server (if it supports “multiviews”, e.g., the Apache web server). Depending on the data transmitted by the request, the web server decides which contents or in which language the contents are displayed. Longer URLs, such as `.../de/index.html` or `.../en/index.html`, can thus be avoided (see section 7.5.9.2 page 332).

! *This option will be omitted from FirstSpirit Version 4.2 on. For being able to create MultiViews paths furthermore, select the option “Multiview URLs“ in the action “Execute generation“ (see Chapter 7.5.9.2 page 332).*

Bar administrator: If the box is *checked*, the server administrator cannot access this project (see section 7.1 page 184). The project is no longer displayed for the server administrator in the project selection list of the JavaClient or WebClient and cannot, therefore, be edited. However, server administrator tasks can still be carried out.

- Create / export / delete new projects
- Create users
- Change the properties of all the projects
- Define project administrators
- Install und uninstall editor and function components
- Execute special server operations

If this option is unchecked, the server administrator can also carry out project administrator tasks and select the project via the project selection list in the JavaClient and WebEdit.

Allow all users: If the box is *checked*, all users known on the server are automatically granted access to the project. This means that users who have not been explicitly added to the project and are thus not listed under the "User" project properties (see section 7.4.7 page 271) are also granted access to the project, i.e. all users authenticated on the server (e.g. via LDAP, SSO). These users do not belong to an internal project group and only have limited permissions in the project, unless they are members of an external project group. These permissions can be configured via access permissions in the project group "Everyone". Access to the project via "external users" can be limited by granting external group members access to the project instead of "Allow all users". They do not have to be explicitly added to the project as users (for further information on external groups see section 7.4.7.2 page 272). If the box is *unchecked*, users who are to be granted access to the project have to be explicitly added to the project (see section 7.4.7 page 271).

Absolute links: Absolute paths can be generated instead of relative paths for page references. This project property affects all page references. The prefix required for completing the absolute links is read from the "Prefix for absolute paths" field in the generation task (see section 7.5.9.2 page 332). This option does not apply to media references. If media references are to be displayed as absolute paths, the reference has to be provided with the attribute "abs". If the setting "abs:2" is used, the prefix for references to media and pages is not used (see the FirstSpirit Online Documentation).

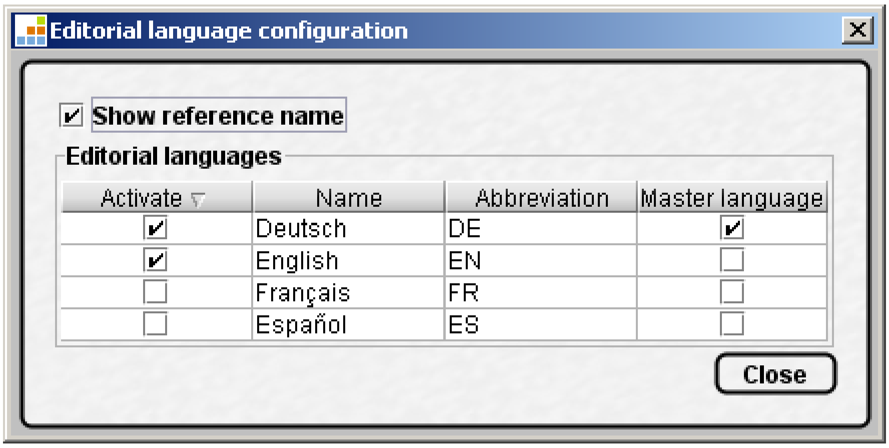
Activate 3.1 compatibility mode: If this option is activated the option „Generate Media in the generation directory" within the Media Store can be used furthermore. This function will be replaced in FirstSpirit Version 4.1 by the possibility to swap the generation of media to a separate schedule (see chapter 7.5.9.2 page 332).

! *The compatibility mode for FirstSpirit Version 3.1 will be omitted in FirstSpirit Version 4.2.*

Reference names are changeable: If this checkbox is activated, reference names can be changed in the project concerned. If the checkbox is deactivated, reference names cannot be changed, the "Change reference names" menu item is greyed out. As a default, the option is enabled when creating new projects, so that reference names can be changed as to date. Regardless of the setting in the Project properties, server and project administrators can change reference names at any time.

! *The "Rename" function (F9) in the JavaClient is not affected by the configuration of the "Reference names changeable" option in the Project properties: Reference names of elements without UID (e.g. sections in the Page Store) can continue to be changed using "Rename". Changes to the reference names of sections can, however, be prevented by disabling the "Display reference names" option in the Project properties, "Options" area, "Configuration of the editorial languages". A change is then only possible via BeanShell (API).*

Editorial language configuration: Click the icon behind the "Editorial language configuration" entry to open the following configuration dialog:



☑ Show reference name

Editorial languages

Activate	Name	Abbreviation	Master language
☑	Deutsch	DE	☑
☑	English	EN	□
□	Français	FR	□
□	Español	ES	□

Figure 7-76: Configure editorial languages

Show reference name: If this checkbox is activated (default setting), in addition to the language-dependent display names (cf. "Activate"), the language-independent reference names of the objects can also be displayed.

Activate: The editorial languages for a project can be specified by activating this checkbox. The "activated" languages can then be set in the FirstSpirit editing environment as the "Preferred display language" (via the "Extras" menu). The editorial languages affect language-dependent contents which have been defined by the template developer, e.g. within the page or section templates. The relevant language-dependent labelling are displayed to the editor, for example in the forms area (labelling of the input fields, tool tips, elements of a ComboBox, etc.) and this affects also the display of the objects in the tree. The editorial languages should not be confused with the project languages (cf. chapter 7.4.5 page 265).

Name / Abbreviation: Name and abbreviation of the required editorial language.

Master language: The activated checkbox in this column identifies the project master language. This is activated as the editorial language as a default.

External link mode: The selection is relevant if the “external address” (URL) link is to be used in the Site-Store of a project. Page references can directly refer to an external page. In this case, either a direct “redirect” to this page is executed (without storing the page on the FirstSpirit Server) or a redirect is executed while additionally generating and storing a page on the server.

Project settings: A page template from the project which is to be used for managing the global project settings can be selected here. The GUI elements defined in the “Form” tab of the page template are displayed in the “project settings” of the “Global settings” administration area and can be filled with contents. In the generation and preview context, the contents are treated like structure variables which have been defined on the root node of the Site-Store. Data preparation takes place via the page template where functions required for manipulating the Project Configuration data are called locally. Values calculated in the template can, if necessary, be inserted into the context via the $CMS_SET(..)$ or CMS_HEADER function and thus requested by any template.

Metadata template: A template which is to be used for the metadata of the project can be selected in this field.

! *In contrast to the page template selection in the editorial environment JavaClient, template selection (for the settings page as well as the metadata template) is not restricted in this area. All the available page templates are always displayed for selection, including templates which have been marked as “not visible”.*

Workflow for deleting elements (from V4.1): A project-specific workflow can be created and tied directly to the delete controls available to date (menu bar buttons, context menu entry) to delete elements in FirstSpirit JavaClient and in FirstSpirit WebClient. Instead of simply deleting an object, for example a page, a more complex delete function can be provided via the workflow, for example the additional deletion of dependent objects of a page.

For further information on deleting elements by means of a workflow, see "FirstSpirit Manual for Developers (basics)".

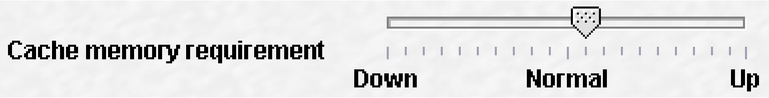


Figure 7-77: Project properties – Cache memory requirement setting

Cache memory requirement: Use the slide control to increase (up) or reduce (down) the distributed memory size for the project cache. The memory allocated by the CacheSizeManager additionally depends on the actually required memory ("preferred size") and the available total memory (see section 9.5 page 423).

Enter version comments for (from FirstSpirit Version 4.1): These combo boxes can be used to individually specify for each store, whether a comment is to be queried for changes to objects when the editor quits edit mode.

> ! *The "Enter version comments for" function is released for FirstSpirit Version 4.1 and higher only. Screenshots are therefore displayed in the new "LightGray" look & feel. The display can differ slightly in the "Classic" look & feel.*

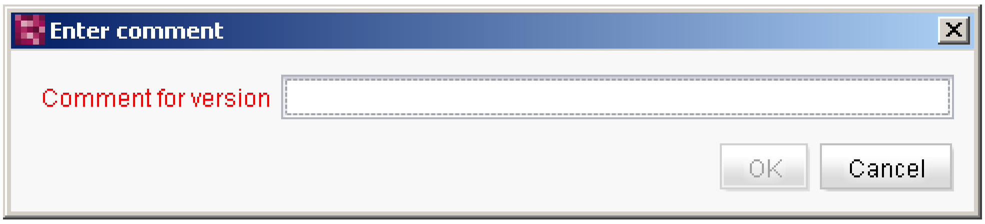


Figure 7-78: Enter version comment

The following options are possible:

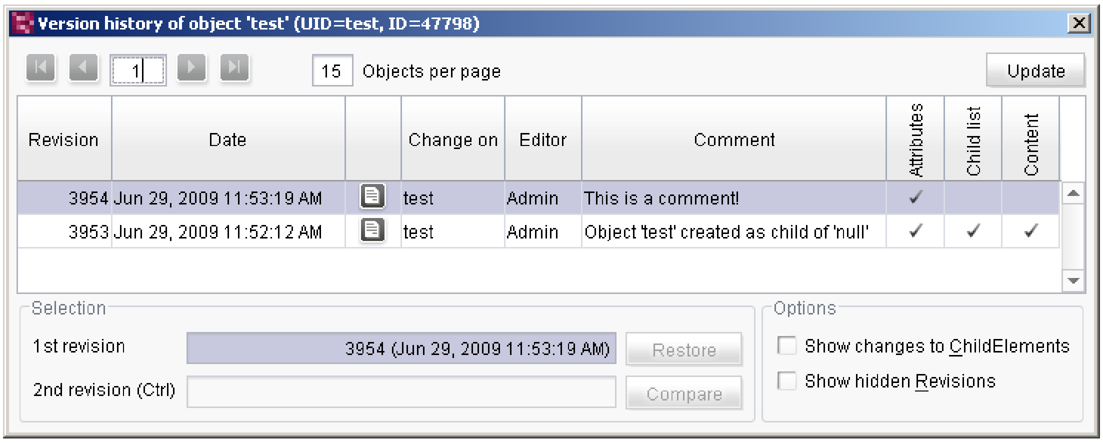
Yes, force: If this option is selected, the comment line shown in Figure 7-78 is displayed to the editor. It can then not be closed using **OK** until the editor has entered a text.

Yes, optional: If this option is selected, the comment line shown in Figure

7-78 is displayed to the editor. It can also be closed without making an entry by clicking **OK**.

No: If this option is selected, the edit can quit edit mode without having the comment line displayed.

If a comment is entered while changing an object, it is subsequently displayed in the object's version history too (in Figure 7-79 "This is a comment!"):



Version history of object 'test' (UID=test, ID=47798)

1 | 15 Objects per page | Update

Revision	Date		Change on	Editor	Comment	Attributes	Child list	Content
3954	Jun 29, 2009 11:53:19 AM		test	Admin	This is a comment!	✓		
3953	Jun 29, 2009 11:52:12 AM		test	Admin	Object 'test' created as child of 'null'	✓	✓	✓

Selection

1st revision: 3954 (Jun 29, 2009 11:53:19 AM) | Restore

2nd revision (Ctrl): | Compare

Options

□ Show changes to ChildElements

□ Show hidden Revisions

Figure 7-79: Change comment in the version history

Figure 7-80: Project properties – Edit logo

Change logo: A chart can be selected here which is to be displayed on the right (editing area) after client start.

Delete logo: A previously defined chart is deleted. After client start, the FirstSpirit logo is displayed on the right (editing area.

Show logo: A preview of the selected logo is displayed. If a logo is not available here, the FirstSpirit logo is displayed by default.

7.4.3 Substitutions

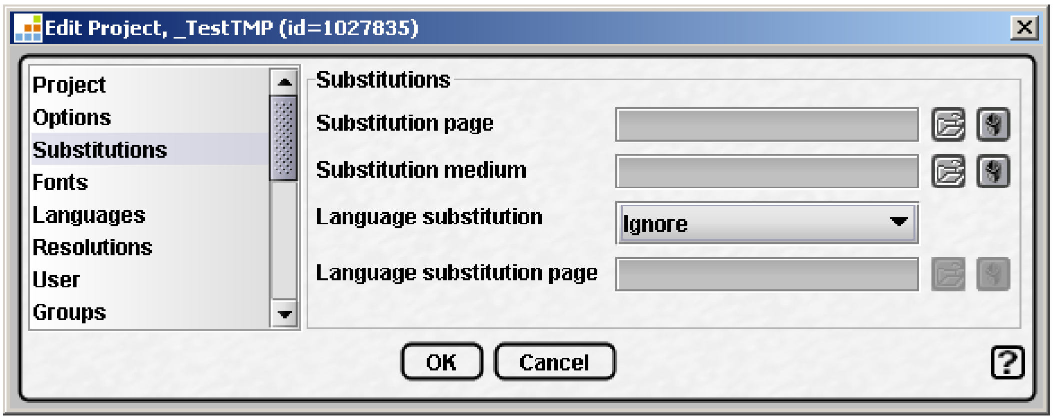


Figure 7-81: Project properties – Substitutions

Substitution page: If a page requested by the user is not available, an alternative page can be selected here. Use the file icon to select the desired page from the Site-Store.

Substitution medium: If a medium is not available, an alternative medium can be specified here. Use the file icon to select the desired medium from the Media-Store.

Language substitution: In multilingual projects some pages might not yet be translated. In FirstSpirit, this is indicated by a respective check mark on the page in the Page-Store.

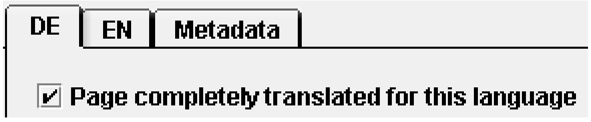


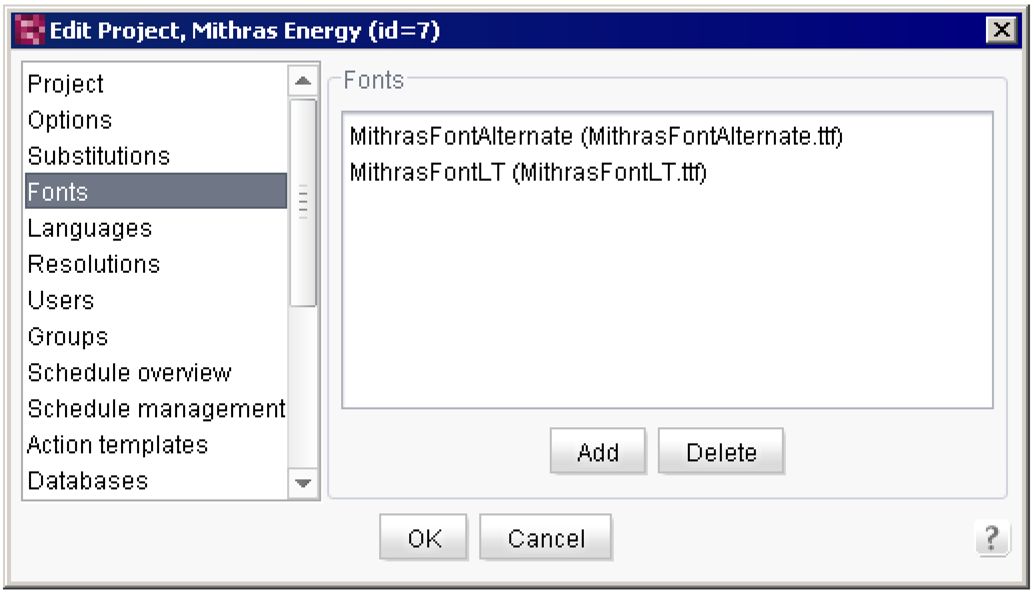
Figure 7-82: Page completely translated for this language

In this case, it is possible to define a rule for handling the respective pages. The following options are available:

- **Ignore:** The page is simply used in its current state for the language. This means that the system assumes the editor has just forgotten to set the check mark.
- **Use master language:** The page is generated in the master language of the project instead of the specified language. If this option is used, the language of a site might change suddenly when changing pages. Therefore, consider this option thoroughly prior to use.
- **Use substitution page:** The substitution page defined above is used.
- **Use special page:** A separate language substitution page (see below) is displayed.

Language substitution page: If the actually requested page for a project language has not yet been translated and the language substitution rule “Use special language” (see above) is active, use the file icon to select an alternative page from the Site-Store.

7.4.4 Fonts

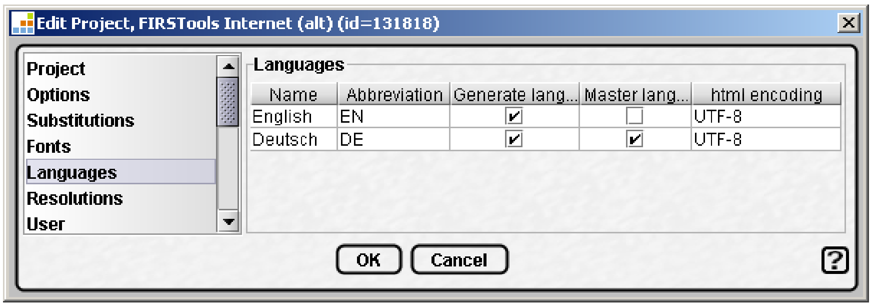


MithrasFontAlternate (MithrasFontAlternate.ttf)
MithrasFontLT (MithrasFontLT.ttf)

Figure 7-83: Project properties – Fonts

Here, you can select which fonts from the fonts which are installed on the server (see Chapter 7.3.5 page 218) are to be available in the current project. Use the buttons "Add" and "Delete".

7.4.5 Languages



Name	Abbreviation	Generate lang...	Master lang...	html encoding
English	EN	☑	□	UTF-8
Deutsch	DE	☑	☑	UTF-8

Figure 7-84: Project properties – Languages

All the project languages are listed here. The table consists of the following columns:

Name: This column displays the name under which the language has been integrated in FirstSpirit. New languages have to be provided to the server via the menu bar before they can be inserted for a specific project in this tab (see section 7.3.7 page 224).

Abbreviation: This column displays the language abbreviation under which the language is used in the clients.

Generate language: This option must be checked, otherwise the language is not considered for generation. This is recommended, e.g., if contents are not yet completely available in this language.

Master language: One language has to be marked as the master language. The master language can be changed by checking the box of the respective language.

html encoding: This column states the encoding of the HTML page which is to be subsequently generated. The presetting for each newly added language is ISO-8859-1 (Western Europe).

With 4.2R4, the "html encoding" text field has been changed into a combobox. All encodings supported by the respective Java version under which the FirstSpirit Server runs are now displayed in this combobox. Click the field to open the combobox, and the required encoding can be selected.

If, on changing from one JDK version to another, the encoding is not supported (any more), this is visualised with red lettering. If a project is exported from one FirstSpirit Server to another, which does not support an encoding used in the project, a warning is issued during the import in the form

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

Right mouse click in the overview window to open the context menu:

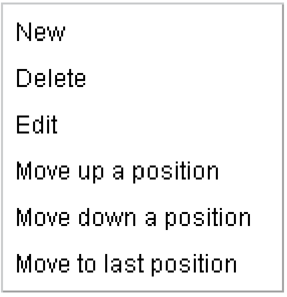


Figure 7-85: Context menu (language area)

New: Use this function to add a new language to the project. A selection list of the languages available on this server appears.

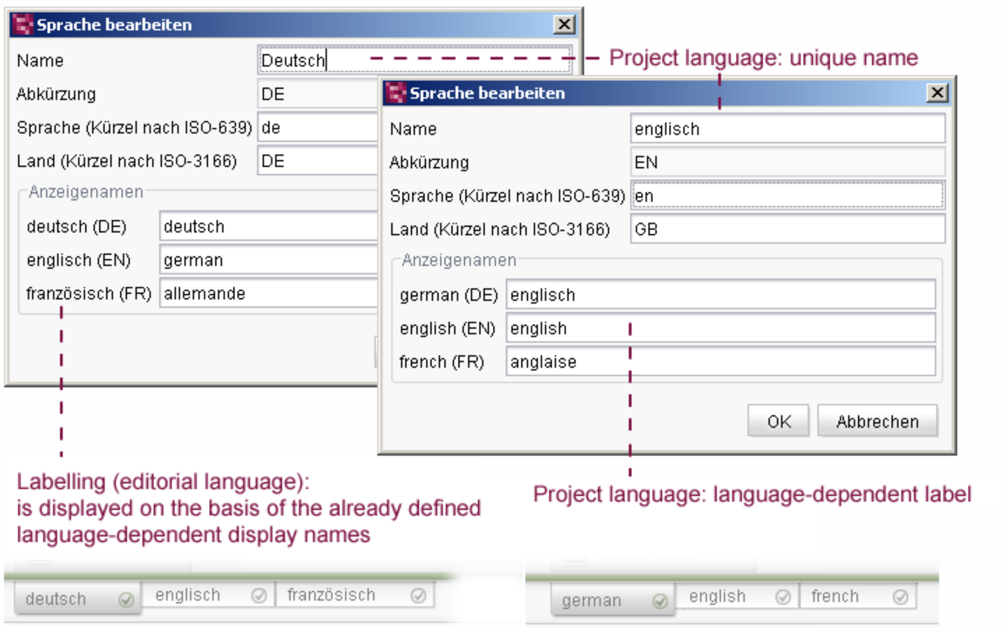
If a new language is inserted, all the Stores in the client (Page-Store, Site-Store, Media-Store, Content-Store) have to be re-released.

Please ensure there is sufficient memory space for this action when handling large projects with a large media volume (>>1000 or mediastore.xml file ~1MB)!

Delete: Use this function to delete the highlighted language from the project.

Project contents may be lost when deleting a language. However, a deleted language can be recreated on the basis of the respective language template. In this case, the previously managed project contents are displayed in the respective language tab again. Contents which have not been saved prior to (language) deletion cannot be restored.

Edit: Use this function to open the selected language for editing. All the language's properties (Name, language according to ISO-639, country according to ISO-3166 and, from FirstSpirit Version 4.2, the language-dependent display names too) can be changed, with the exception of the abbreviation which is required for clear, unique identification of the language in the project and may not be changed.



Sprache bearbeiten

Name	Deutsch
Abkürzung	DE
Sprache (Kürzel nach ISO-639)	de
Land (Kürzel nach ISO-3166)	DE

Anzeigenamen

deutsch (DE)	deutsch
englisch (EN)	german
französisch (FR)	allemande

Sprache bearbeiten

Name	englisch
Abkürzung	EN
Sprache (Kürzel nach ISO-639)	en
Land (Kürzel nach ISO-3166)	GB

Anzeigenamen

german (DE)	englisch
english (EN)	english
french (FR)	anglaise

Project language: unique name

Labelling (editorial language): is displayed on the basis of the already defined language-dependent display names

Project language: language-dependent label

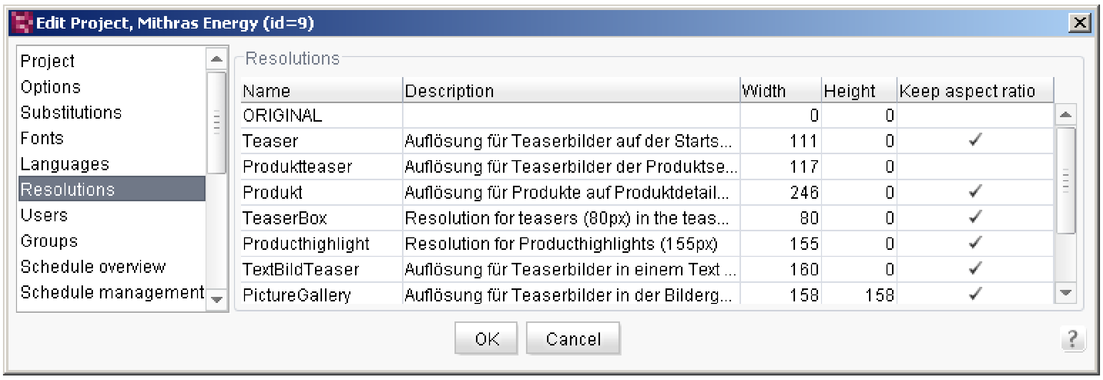
Figure 7-86: Edit languages

From FirstSpirit Version 4.2, language-dependent display names can be defined for a project language. The relevant input fields are displayed in the "Display names" area of the form. Language-dependent display names can be defined for all a project's editing languages.

One position up/down: The defined project languages are displayed in JavaClient in the defined order. The context menu can be used to adjust the positioning of the language tab in the project. To this end, the relevant entry simply has to be selected and gradually moved up or down using the context menu or the relevant buttons. The following applies: The master language always remains in the first place in the order.

Place in the last position: Use this entry to place the selected project language in the last position.

7.4.6 Resolutions



Name	Description	Width	Height	Keep aspect ratio
ORIGINAL		0	0	
Teaser	Auflösung für Teaserbilder auf der Starts...	111	0	✓
Produktteaser	Auflösung für Teaserbilder der Produktse...	117	0	
Produkt	Auflösung für Produkte auf Produktdetail...	246	0	✓
TeaserBox	Resolution for teasers (80px) in the teas...	80	0	✓
Producthighlight	Resolution for Producthighlights (155px)	155	0	✓
TextBildTeaser	Auflösung für Teaserbilder in einem Text ...	160	0	✓
PictureGallery	Auflösung für Teaserbilder in der Bilderg...	158	158	✓

Figure 7-87: Project properties – Resolutions

All the resolutions defined for the project are listed here. The table consists of the following columns:

Name: This column displays the unique technical name assigned to a resolution. This name is required fort he identification in the project and can not be changed afterwards. At first, a project only has the resolution “ORIGINAL”. This resolution represents the unchanged media in the resolution in which it has been inserted into the Media-Store (symbolised by “0” for width and height). Upper and lower case letters can be used for the name.

Description: Input field for an optional technical resolution description. This information is only provided in the project properties.

Width: This column displays the resolution width in pixels. “0” symbolises that the width results from the width-to-height ratio compared to the original resolution.

Height: This column displays the resolution height in pixels. “0” symbolises that the height results from the width-to-height ratio compared to the original resolution.

Keep aspect ratio: This option can be used to specify whether the original picture's aspect ratio is to be taken into account or not for the respective resolution of the subsequent output on the website. If the option is selected, the aspect ratio of the original picture is retained.

Resolutions for which this option is disabled, can be output "squashed" on the website, if the aspect ratio of the original picture is not the same as the aspect ratio of the corresponding resolution. The option should be selected to prevent this

Right-click in the overview window to open the context menu:

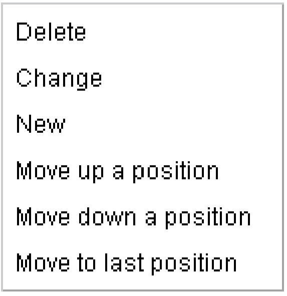


Figure 7-88: Context menu (resolutions area)

Delete: Use this function to delete the selected resolution from the project.

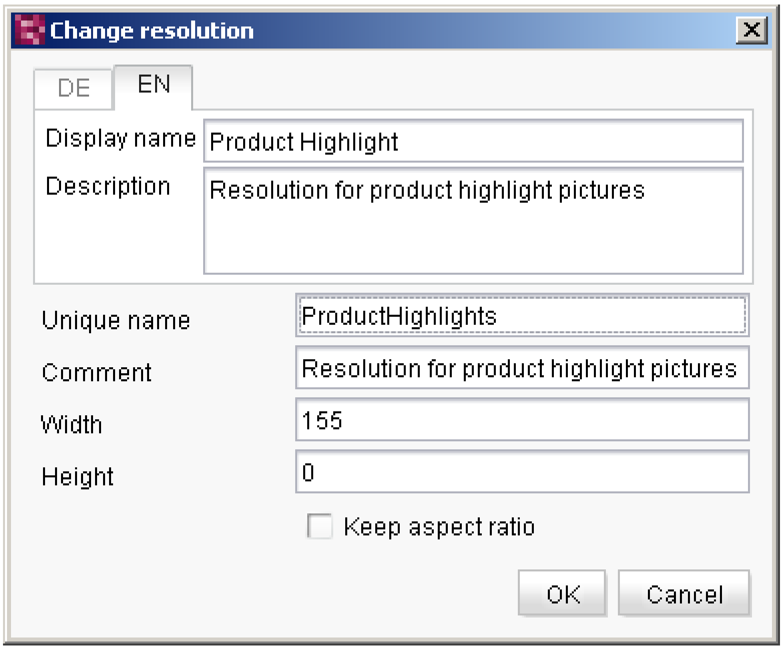
Change: This function can be used to open the selected resolution for editing. In addition, from FirstSpirit Version 4.2, the language-dependent display names and the language-dependent descriptions of a resolution can be changed. All other properties cannot be subsequently changed.

New: Use this function to add a new resolution. Invoking the function opens a dialog box in which the settings for the new resolution can be made.

Apart from the known input fields for the unique technical name, the description and the definition of the resolution width and height, **from FirstSpirit Version 4.2**, additional language-dependent display names and language-dependent descriptions of a resolution can be defined in all editing languages.

If a "preferred display language" is defined in FirstSpirit JavaClient, the corresponding language-dependent display names of the resolutions are displayed in JavaClient.

Once the new resolution for media has been defined, a new resolution line appears in the Media-Store.



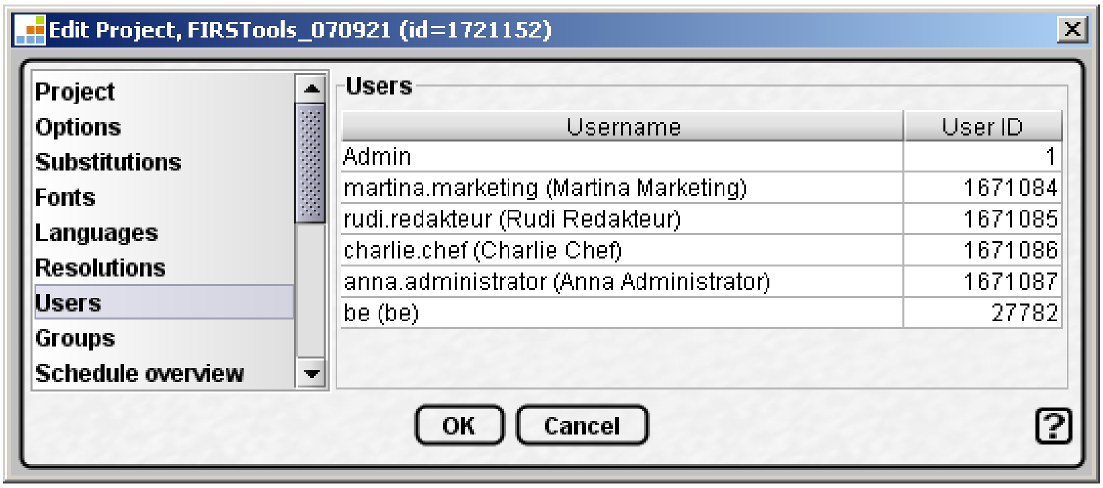
Display name: Product Highlight
Description: Resolution for product highlight pictures

Unique name: ProductHighlights
Comment: Resolution for product highlight pictures
Width: 155
Height: 0
□ Keep aspect ratio

Figure 7-89: Add new resolution

The names of the already deleted resolutions should not be reused, as this could lead to the transfer of media calculated according to the deleted resolution.

7.4.7 Users



Users

Username	User ID
Admin	1
martina.marketing (Martina Marketing)	1671084
rudi.redakteur (Rudi Redakteur)	1671085
charlie.chef (Charlie Chef)	1671086
anna.administrator (Anna Administrator)	1671087
be (be)	27782

Figure 7-90: Project properties – Users

All the users who have project access are listed here. The table consists of the following columns:

User name: This column displays the unique user name used for identification on the server. For manually added users, this is the name entered in the "Login" field during new user creation (see Figure 7-23). Users who are authenticated from a third-party system are also identified via the login and listed here under this name. If an optional entry has been defined for the user in the "Name" field, it is displayed in brackets after the login in the list.

User ID: The ID is automatically assigned by the system and cannot be changed.

Right mouse click in the overview window to open the context menu:

Figure 7-91: Context menu (user area)

Additional functions which are described below can be used via the context menu:

7.4.7.1 Delete user from the project

Use this function to delete the highlighted user from the project. Select the user to be deleted in the user overview and subsequently call the context menu item "Delete". The selected user is deleted from the project.

7.4.7.2 Add user to the project

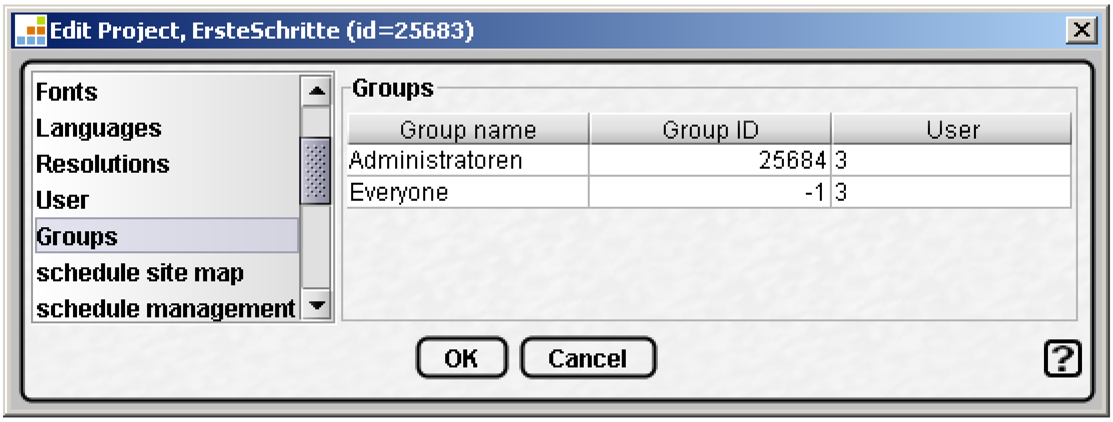
Use the context menu function “Add” to add new users to a project. A selection list of all the users on the server appears. Select the desired user here. Only after this allocation are users granted project access permissions. Project users can receive extended access permissions by being allocated to a project group.

FirstSpirit differentiates between users who have been manually created on the FirstSpirit Server and users who have been automatically imported from a third-party system:

1. Manually created users: It is possible to manually create a user via the menu item “User” in the Server and Project Configuration menu bar (see section 7.2.3.8 page 198). In the project properties, a user who has been manually created on the server can be allocated to a project (see section 7.4.7 page 271) The user is then automatically a member of the project group “Everyone”, but can also be added to any other internal group (see section 7.4.8.6 page 278).

2. Automatically created users: Besides manual user creation on the FirstSpirit Server, it is also possible to import a user automatically via a third-party system. Users who authenticate themselves on the FirstSpirit Server via a third-party system (e.g. LDAP) are automatically created as FirstSpirit Server users after initial login (and are, therefore, displayed in the user list) without having to be explicitly created via the Server and Project Configuration. Group allocation takes place according to point 1. Nonetheless, these users can also be a member of an external group (see section 7.4.8.2 page 275).

7.4.8 Groups



Group name	Group ID	User
Administratoren	25684	3
Everyone	-1	3

Figure 7-92: Project properties – Groups

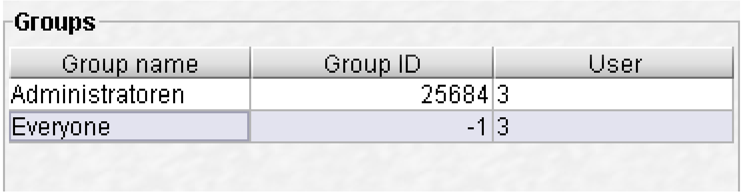
All groups with project access are listed here. Each project initially has the default groups “Administrators” and “Everyone”, which cannot be deleted. It is possible to define as many groups as desired for each project. Groups contain a number of users, but no further groups.

Group name: This column displays the unique group name. The evaluation differentiates between upper and lower case letters.

Group ID: This column displays the unique group ID which is automatically assigned during new group creation.

User: Number of group members. Number of users added to this group. It is possible to add or delete users for internal groups (see section 7.4.7 page 271). It is not possible to change the number of users via the Server and Project Configuration for external groups; the comment “External group” is displayed instead of the number of users (see Figure 7-93). (For further information on internal and external groups see section 7.4.8.2 page 275.)

E-mail distribution list: From FirstSpirit Version 4.2R4 on, there is an option in the project properties for specifying an e-mail distribution list for groups, to which the notification e-mails are sent when a workflow activity or transition is performed. In this way it is now possible to send e-mails, which are sent as part of a workflow, to be sent to all members of external groups.



Groups

Group name	Group ID	User
Administratoren	25684	3
Everyone	-1	3

Figure 7-93: Project properties – Groups: Group overview

Group and access permissions: The assignment and management of access permissions can be considerably simplified via group definitions. For example, if a specific area is to be hidden for a number of editors and the number of editors occasionally changes, it is recommended to define a new group "Editors A". All the editors for whom the area is to be hidden are added to this group. In the FirstSpirit JavaClient, the root of the affected part tree is hidden for the group "Editors A" by withdrawing the respective permissions of this group. If it becomes necessary to grant a specific editor access, the user is simply removed from the group "Editors A"; therefore, permission definition in the FirstSpirit JavaClient does not have to be modified.

Right mouse click in the overview window to open the context menu:

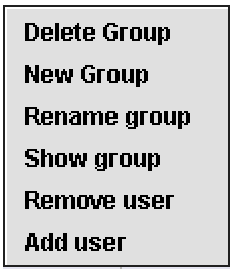


Figure 7-94: Context menu (groups area)

The additional functions described below can be used via the context menu:

- Delete group (see section 7.4.8.1 page 274)
- New group (see section 7.4.8.2 page 275)
- Edit E-mail distribution list (see section 7.4.8.3 page 277)
- Show group (see section 7.4.8.4 page 277)
- Remove user (see section 7.4.8.5 page 277)
- Add user (see section 7.4.8.6 page 278)

7.4.8.1 Delete group

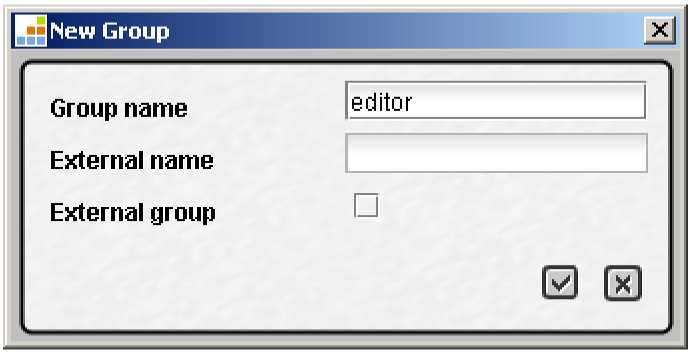
Use the context menu item "Delete group" to delete groups which have been added to a project. A confirmation prompt appears prior to deleting. After confirmation, the group is deleted from the project and no longer appears in the "Groups" area in the project properties. All the members of the deleted group loose their access permission to the project (exception: Users who are a member of another internal or external group still allocated to the project).

The default groups "Administrators" and "Everyone" manage the initial permission assignment in a project and cannot, therefore, be deleted.

7.4.8.2 New group

FirstSpirit differentiates between internal and external groups:

Internal groups are used for internal user and permission management and can be generated and edited directly via FirstSpirit. For example, users can be added to or deleted from an internal group via the project properties. The group properties can be edited via the Server and Project Configuration. Only the "Group name" is entered when creating an internal group (see Figure 7-95). After saving the "New group" dialogue box, the new group appears in the group overview. Users can be subsequently added to the new internal group (see section 7.4.8.6 page 278).



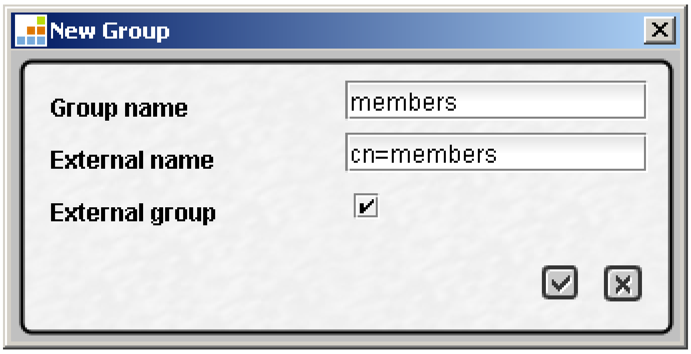
Group name: editor
External name:
External group: □

Figure 7-95: Create new internal group

External groups are also allocated to a project via the Server and Project Configuration. In contrast to internal groups, they cannot be generated via FirstSpirit but derive from a third-party system (e.g. LDAP). External group membership is assigned via user attributes, i.e. users cannot be added to an external group via the context menu (see section 7.4.8.6 page 278). Users using, e.g., LDAP authentication receive membership to a group (which does not necessarily have to be allocated to a project) as attributes and can be added to the project via this group. External group members are only granted project access once the external group has been allocated to the project.

The "Group name" field has to be filled in first to create an external group in the project. This is the internal group name under which the group is known and used in the FirstSpirit project. The external group name, i.e. the name of the group in external systems, is entered in the “External name” field. For LDAP the “External name” is the LDAP-DN, e.g. `CN=employee,CN=users,DC=e-spirit,DC=de`. The field can only be edited if the box “External group” has been checked.

On checking group membership, the system checks internally whether the character string given for the "External name" is fully contained in an LDAP-DN of the logged in user's groups. If, for example, `cn=Mitarbeiter` only is given for "External name", the group membership would fit the LDAP groups
`cn=Mitarbeiter,ou=Entwicklung,ou=dc=domain,dc=com`
and
`cn=Mitarbeiter,ou=Vertrieb,dc=domain,com` passen. To ensure unique assignment, the group's full LDAP-DN must be entered for the "External name". Case sensitivity (upper and lower case) is irrelevant here, as it generally is in LDAP.



Group name: members
External name: cn=members
External group: ☑

Figure 7-96: Create new external group

FirstSpirit does not check whether an external group is available. If the external group name is unknown, the group is still added to the project as an external group. However, if this is the case, it does not have any members (allocated users).

7.4.8.3 Edit E-mail distribution list

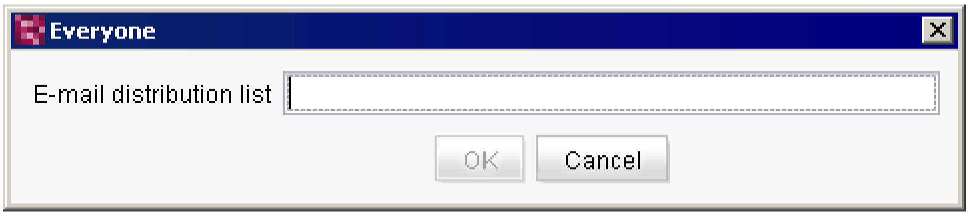
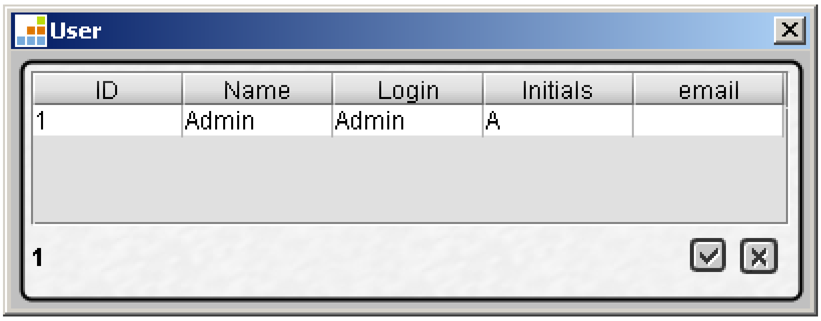


Figure 7-97: Specify e-mail distribution list

Several e-mail addresses can be entered, separated by semicolons. The field can also be left empty, or e-mail addresses already entered can be deleted.

Press "OK" to accept the changes for the group, the e-mail distribution list is now displayed in the groups overview:

7.4.8.4 Show group



ID	Name	Login	Initials	email
1	Admin	Admin	A	

Figure 7-98: Show group

Use the context menu item “Show group” to display the users of a group. All the users who are a member of the group and, therefore, have project access are displayed here.

The members cannot be displayed if it is an external group. It is also impossible to add users to or to remove users from an external group. Only the external group name can be changed here.

7.4.8.5 Remove user

A list of all the members of the highlighted group appears for internal groups. Select the user to be removed in the overview and confirm it via ☑. The selected user is deleted from the internal group.

Multiple selection is possible via:

- Select a user in the list while at the same time pressing the CTRL key
- CTRL+ SHIFT (select from start to end point)
- CTRL + A (select all users)

> *Membership to an external group is assigned via user attributes of the third-party system, i.e. users of an external group cannot be deleted via the Server and Project Configuration.*
>
> *It is not possible to add users to or delete users from the default group "Everyone". The following applies to this group: All users with project access (via membership to an internal or external group allocated to the project) are automatically members of the "Everyone" group and are at least granted access permissions defined for "Everyone".*

7.4.8.6 Add user

A user can be a member of several groups. Variously configured access permissions can be assigned to each group (see section 7.4.8 page 273). The members of the group receive all the project access permissions of the group.

A list of all the users added to the project and who are not currently members of the selected group appears for internal groups. Select the users to be added in the overview. Confirm via ☑ to add the selected users to the internal group.

Multiple selection is possible via:

- Select a user in the list while at the same time pressing the CTRL key
- CTRL+ SHIFT (select from start to end point)
- CTRL + A (select all users)

Membership to an external group is assigned via user attributes of the third-party system, i.e. users cannot be added to an external group via the Server and Project Configuration.

It is impossible to add users to or delete users from the default group "Everyone". The following applies to this group: All users with project access (via membership to an internal or external group allocated to the project) are automatically members of the "Everyone" group and are at least granted access permissions defined for "Everyone".

7.4.9 Schedule entry overview

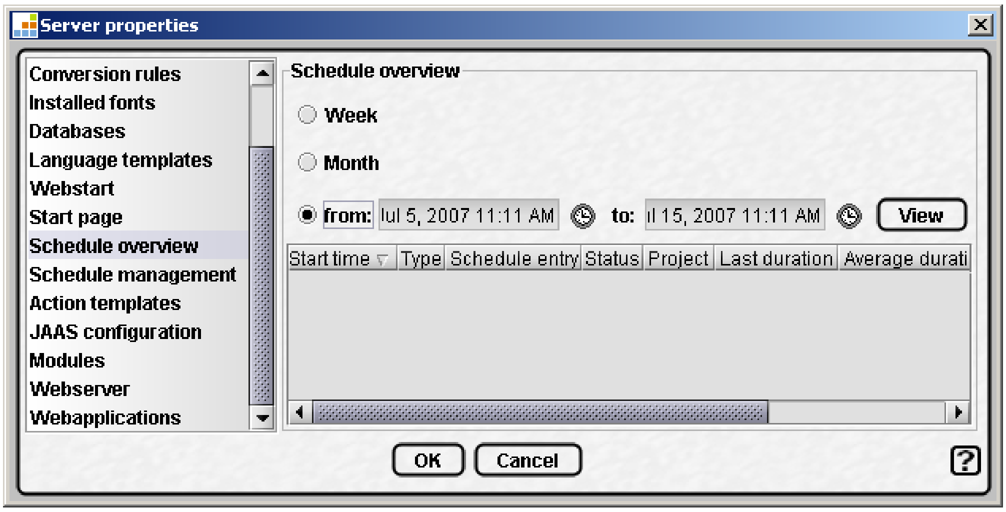
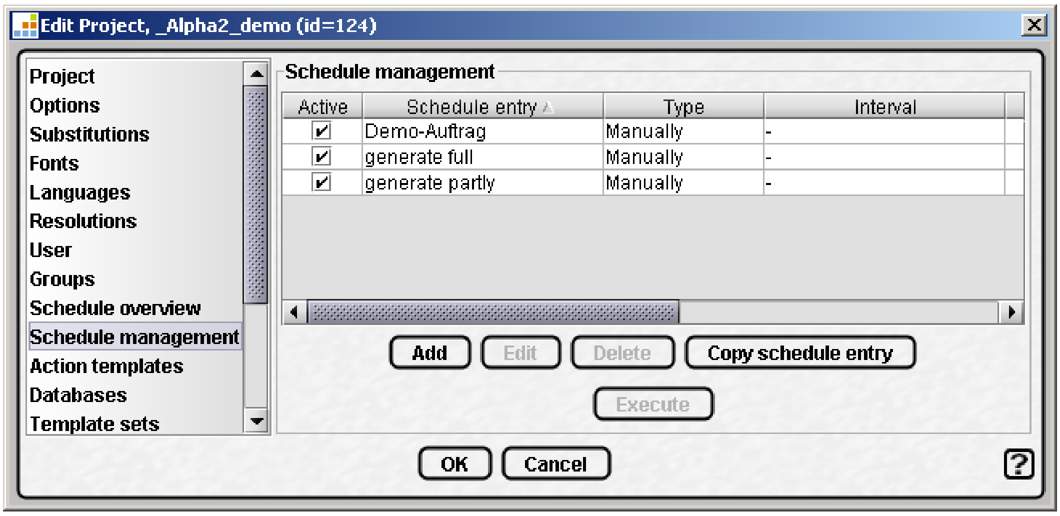


Figure 7-99: Project properties – Schedule entry overview

Schedule entry planning details are displayed in this area. Besides project-related settings, it is also possible to define server-related settings for schedule entry planning (see section 7.3.10 page 229). See section 7.5 (page 310 ff) for a complete overview of FirstSpirit schedule entry planning.

7.4.10 Schedule entry management

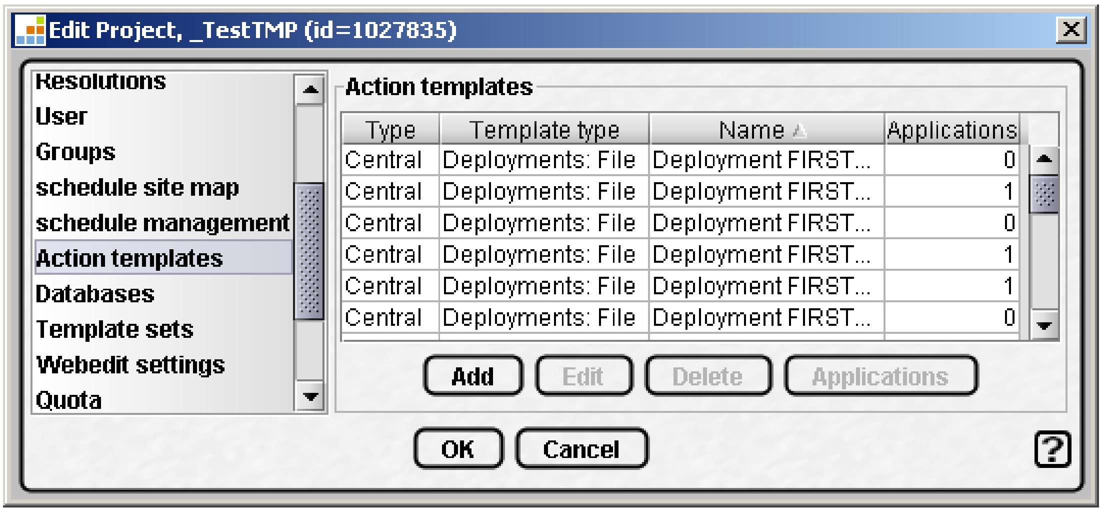


Active	Schedule entry	Type	Interval
☑	Demo-Auftrag	Manually	-
☑	generate full	Manually	-
☑	generate partly	Manually	-

Figure 7-100: Project properties – Schedule entry management

Schedule entries can be created via the schedule entry planning. Schedule entries summarise supplementary actions and, depending on the creation location, are either server or project-related (see section 7.5.2 (page 314)).

7.4.11 Action templates

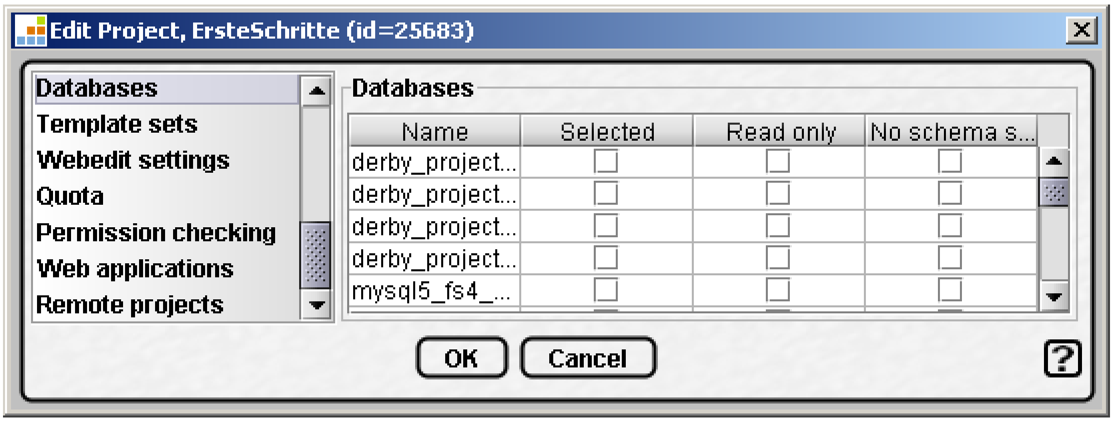


Type	Template type	Name	Applications
Central	Deployments: File	Deployment FIRST...	0
Central	Deployments: File	Deployment FIRST...	1
Central	Deployments: File	Deployment FIRST...	0
Central	Deployments: File	Deployment FIRST...	1
Central	Deployments: File	Deployment FIRST...	1
Central	Deployments: File	Deployment FIRST...	0

Figure 7-101: Project properties – Action templates

See section 7.5.3 (page 317)

7.4.12 Databases



Name	Selected	Read only	No schema s...
derby_project...	□	□	□
derby_project...	□	□	□
derby_project...	□	□	□
derby_project...	□	□	□
mysql5_fs4_...	□	□	□

Figure 7-102: Project properties – Databases

All the databases defined for the project are listed here. The table consists of the following columns:

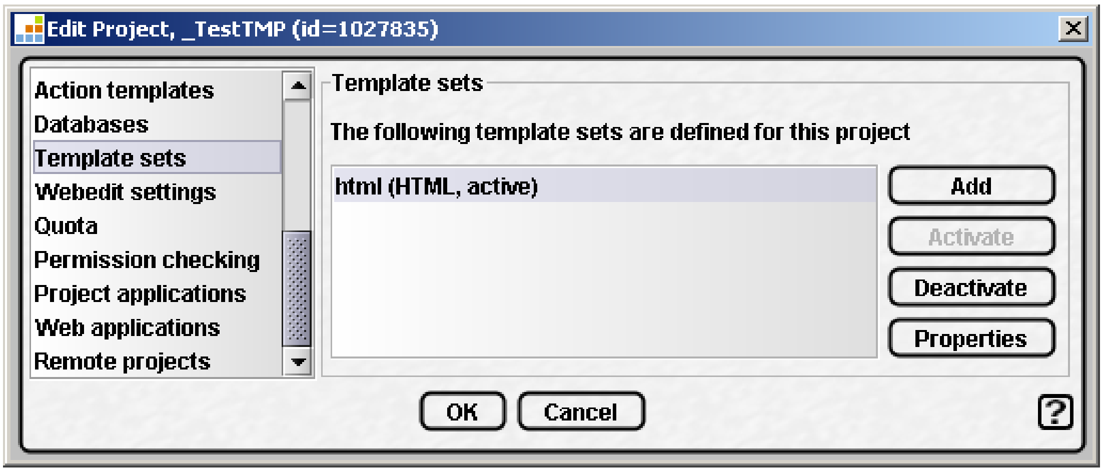
Name: This column displays the names of all the server databases.

Selected: Each checked database can be used for a database schema in the FirstSpirit-Client. This automatically sets write access to the database for this project.

Read-only: Check this option if the selected database should only be available as read-only in the project. For external databases („No schema Sync“ is active) the write protection must be activated.

No schema sync: If this option is checked, database schema changes carried out in the FirstSpirit JavaClient are not transferred to the physical database. This option must be activated for the connection of external databases (see chapter 4.8.8 page 152).

7.4.13 Template sets


Edit Project, _TestTMP (id=1027835)

Template sets

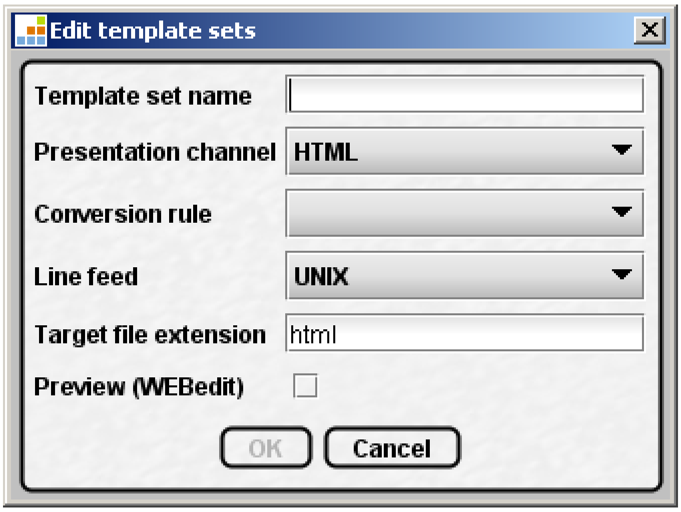
The following template sets are defined for this project

html (HTML, active)

Figure 7-103: Project properties – Template sets

Template sets which are to be available in the project are defined from the presentation channels of the server.

Add Click on this button to open a window for the settings of the new template set.


Edit template sets

Template set name	
Presentation channel	HTML
Conversion rule	
Line feed	UNIX
Target file extension	html
Preview (WEBedit)	□

Figure 7-104: Add new template set

Template set name: Enter the template set name which is displayed in the page, section and format templates in the JavaClient.

Presentation channel: A presentation channel defined in the server properties can be selected here. (See section 7.3.2 page 214)

Conversion rule: A conversion rule defined in the server properties can be selected here. (See section 7.3.3 page 215)

Line feed: All the line feeds to be generated from the template sets are replaced by operating system-specific line feeds. For the “Windows” default setting the line feed is automatically replaced by the control characters "carriage return" and "line feed" [\r\n]; “line feed” [\n] and "carriage return" [\r] are replaced for “Unix” and “Mac” respectively.

Target file extension: Specify the final file extension on the web server here. The link to this file generated by FirstSpirit consists of the file name used in the Site-Store and this extension.

Preview (WEBedit): The default preview URL is defined as `/fs4preview/ preview`. The previews of the individual projects occur, as far as configured, as `/fs4preview_{PID}/preview`. If the checkbox „Preview (WEBedit)“ is enabled, the rendered preview will be displayed in the WebClient.

Activate Click on this button to activate the selected template set. (The current state is displayed in brackets.)

Deactivate Click on this button to deactivate the selected template set. (The current state is displayed in brackets.)

Properties Click on this button to open a window for editing the settings for the selected template set (see Figure 7-104: Add new template set).

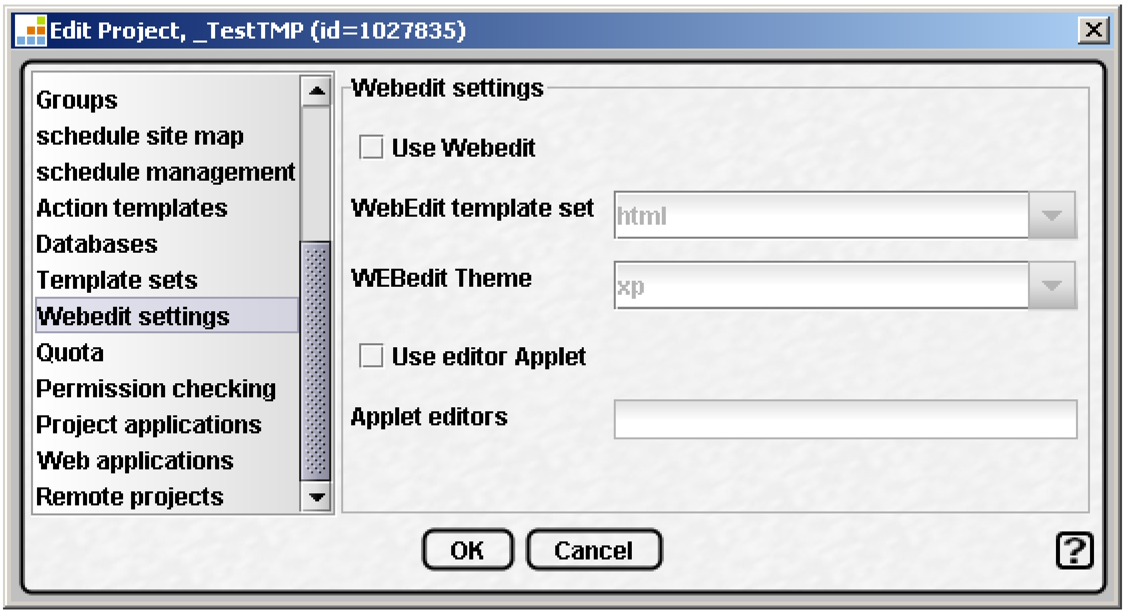
<u>From FirstSpirit Version 4.2:</u>

Display name / description: From FirstSpirit Version 4.2, an optional language-dependent display name and a language-dependent description can be issued in all editing languages for each template set.

If a "preferred display language" is defined in FirstSpirit JavaClient, the corresponding language-dependent display names of the template sets are displayed in JavaClient.

▲ ▼ From FirstSpirit Version 4.2, the template sets of a project can be re-sorted within the server and project configuration. To do this, the relevant entry simply has to be selected and gradually moved up or down using the relevant buttons. The changed order then affects the order of the tabs in JavaClient.

7.4.14 WebEdit settings



Webedit settings

□ Use Webedit

WebEdit template set: html

WEBedit Theme: xp

□ Use editor Applet

Applet editors:

Figure 7-105: Project properties – WebEdit settings

Use WebEdit: If this option is selected, this project can then be edited in WebClient (author environment). If this option is disabled, the project can no longer be edited in WebClient. The following message appears if the option is disabled: "Remove the WebEdit templates from the project?". If the "WebEdit Format Templates" option was selected on creating the project (see Figure 7-5), the folder with the WebEdit format templates is deleted with "Yes". Format templates subsequently created in this folder are also deleted at the same time. If the answer is "No", the WebEdit format templates are retained, but WebEdit usage is prevented.

For the use of WebEdit **from FirstSpirit Version 4.2** besides the WebEdit format templates also special media are necessary (for the Easy-Edit mode). These media are located in the folder “WebClient media (EasyEdit)“ in the Media Store, directly beneath the root node. If the option “Use Webedit“ is deactivated the following query appears in projects which use the release function: “Remove the WEBedit templates from the project? IMPORTANT: Removing the WEBedit templates results in the release of the higher level folders of "WebClient media (EasyEdit)." With “Yes“ the WebEdit format templates as well as the WebEdit media will be deleted and the superordinate folder of "WebClient media (EasyEdit)" will be released. This is necessary to be able to re-use the UIDs of the WebEdit media in case of re-activating the WebEdit mode later on. Without releasing, the Easy-Edit function would not work properly after re-activating the WebEdit mode. If the answer is "No", the WebEdit format templates and the WebEdit media are retained, but WebEdit usage is prevented.

When updating WebEdit format templates, it must be noted that assignment of the WebEdit channel is NOT taken into account in the import.

If the WebEdit channel in the specific project is not "html", assignment is attempted on the basis of the channel type. The format templates are copied into the first channel of the type "html" (regardless of the name of the channel). If this is not the required WebEdit channel, the format templates must be manually copied from the Html channel into the WebEdit channel.

Therefore, the following applies: On updating the format templates it is necessary to always ensure that the new templates are really imported into the WebEdit channel. To this end, the version number of the format template should be checked.

WebEdit template set: The template set of the project to be used for the WebEdit page presentation can be selected here.

WebEdit Theme: Up to and including Version 4.2R2, the theme for setting the WebEdit menu bar appearance can be selected here.

❗ *When switching over the themes it is necessary to ensure that the preview cache is completely deleted. To this end, the preview web application ("fs4preview") must be restarted (see chapter 7.3.14 page 232).*

❗ *The creation of **new** "WebEdit Themes" is no longer supported **from FirstSpirit Version 4.0**. The Theming function of WebEdit will be dropped with the release of WebEdit Version 5.0. Support for the creation of new themes for WebEdit is therefore no longer planned.*

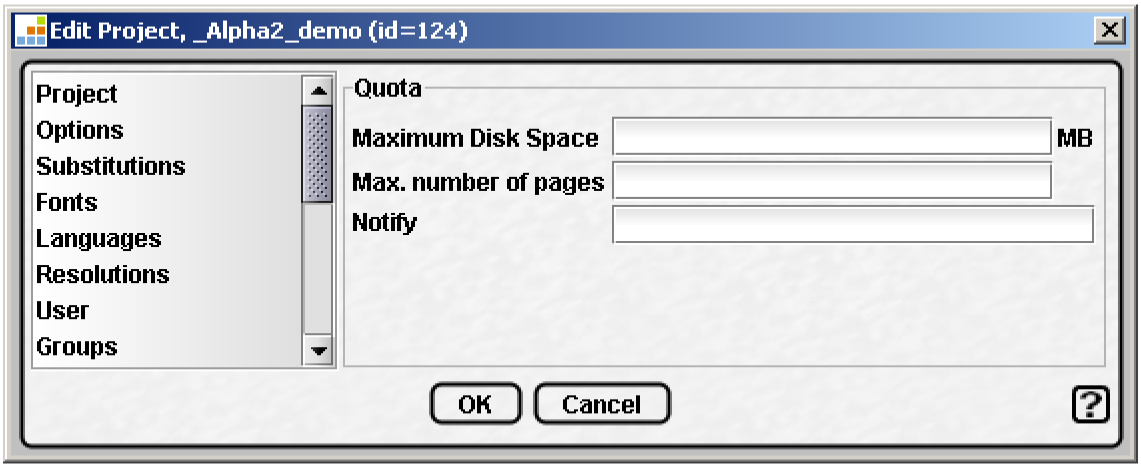
❗ *The FirstSpirit standard themes ("default", "xp") and the specific SAP theme ("sap") can however continue to be used **in FirstSpirit Version 4.1**. **From FirstSpirit Version 4.2**, the theming function for WebEdit can only be used to a limited extent. Among other things, the toolbar in XP-Theme has been completely revised and is therefore no longer included in XP-Theme. The following applies to all other themes: All css files from the theme project have been manually revised and are no longer contained in the theme project. The same applies to new (and changed) input components in WebEdit 4.2 (among other things, CMS_INPUT_ OBJECTCHOOSER, CMS_INPUT_REF). These components are no longer contained in the Theme project.*

Use editor Applet: The setting “Use editor Applet” is only possible if the option “Use WebEdit” has also been checked for a project. If “Use editor applet” has been checked, some input components can be used as Java Applet in WebEdit (instead of a JSP editor).

Development in this area is not yet fully completed. The function has therefore not yet been released.

Applet editors: This field lists the input components which are to be rendered as applet, e.g. input component CMS_INPUT_TEXT. A comma-separated input of several components is possible. (The change affects all this type of input components.)
Development in this area is not yet fully completed. The function has therefore not yet been released.

7.4.15 Quota



Edit Project, _Alpha2_demo (id=124)

Quota

Maximum Disk Space MB

Max. number of pages

Notify

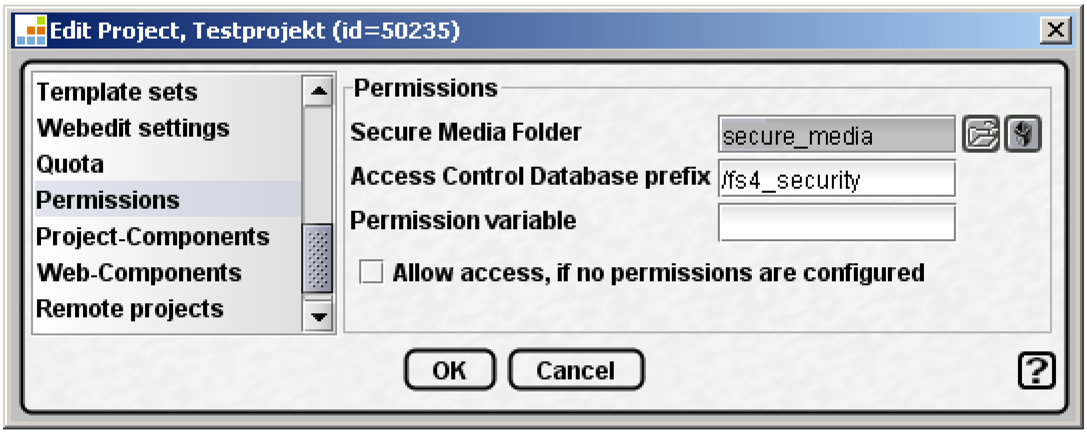
Figure 7-106: Project properties - Quota

Maximum disk space: Specifies the maximum memory space available to the project on the CMS server.

Max. number of pages: Specifies the maximum number of pages for a project.

Notify: Specify who is to be notified once the memory or page limit is reached.

7.4.16 Permissions



Edit Project, Testprojekt (id=50235)

Permissions

Secure Media Folder: secure_media

Access Control Database prefix: /fs4_security

Permission variable:

☐ Allow access, if no permissions are configured

Figure 7-107: Project properties – Permissions

Secure Media Folder: Click on the icon to open the dialogue „Secure media folder“, which contains the Media-Store’s tree structure. A Media-Store folder can be selected, which should contain the „secure media“ in the project. The „secure media folder“ is particularly marked in the JavaClient. Click the icon to delete an already selected folder[28] (see chapter 11.3 page 455).

> ! *The settings defined here for "Secure Media" affect the FirstSpirit preview generation only. The generated or published content can be protected via the FirstSpirit SECURITY module only (cf. chapter 11.3 page 455).*

Access Control Database prefix: Specify a prefix to complete the ACL database information of a file. The complete path to a file in FirstSpirit always consists of three parts:

- URL of the web application (e.g. http://myServer.de)
- Prefix for the Access Control Database (/fs4_security)
- Path to a file (/de/index.html)

The complete path within the ACL database is made up of the prefix and the path to the file, e.g. `/fs4_security/de/index.html`

In Figure 7-107 the prefix is "/fs4_security" and corresponds to the direct sub-subdirectory "fs4_security" of the web application, for example "live": "~Webserver/webapps/live/fs4_security".

Specification of the prefix is absolute for the web application. In addition, in this case the prefix equals the last part of the "Path on live server" field value in the dialog of the deployment servlet (see chapter 7.5.9.3.3 page 343).

For further information, see documentation for the FirstSpirit SECURITY module.

Permission variable: The access permissions for objects can be defined via the metadata in the Media-Store. To achieve this, an input component is defined in a page template (CMS_INPUT_PERMISSION). The name assigned to this input component (e.g. meta-permission) has to be specified here. FirstSpirit decides whether a user has access to an object or not based on the input component value in the metadata tab.

For further information see the FirstSpirit SECURITY documentation.

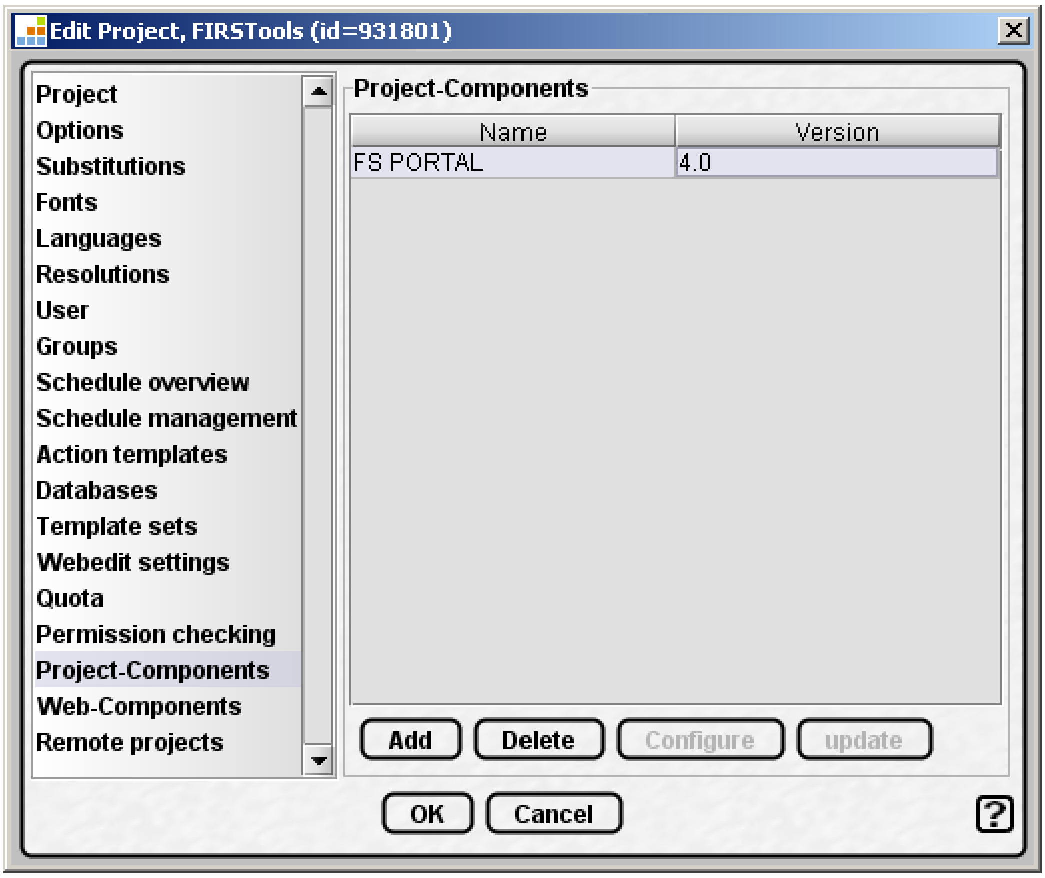
Allow access, if no permissions are configured: If the checkbox is enabled, all group members (groups.xml file) have access to the secure media, if no different

[28] See documentation: SECU40DE_FirstSpirit_Modules_Security.pdf

definition has been defined by the Permission variable. If the checkbox is disabled, only the permissions defined by the Permission variable are evaluated.

Further Information to user permissions see section 11 page 445. Further Information to the “secure media” concept see section 11.3.3 page 458.

7.4.17 Project components



Name	Version
FS PORTAL	4.0

Figure 7-108: Project properties – Project components

Components which are to be available for the entire project (and not only for individual project parts) are stored here (see section 7.4.17). Specific steps are required for installing a project application. For example, it is necessary to install the respective module on the server before installing the portal component in a project (see section 7.3.14 page 232). An example of a project component is the FirstSpirit portal component, which is part of the module “FS Portal” (see the FirstSpirit Portal documentation).

Add Subsequently click on the button to add the components (“project application” type) to a project. All the functions of the installed component are then available in the project. (In case of the portal component, portal folders are, e.g., displayed in the Template-Store of the project.)

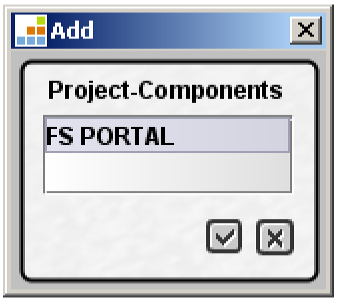


Figure 7-109: Add a project component

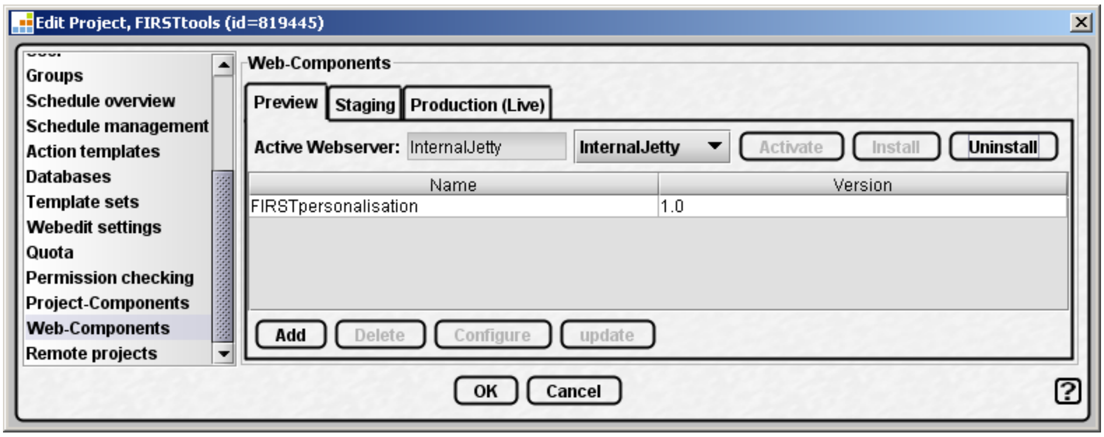
Delete Click on the button to delete components ("project application" type) which have already been added to a project.

All the related contents as well as the Project Configuration of the component are lost when deleting a project component.

Configure Click on the button to edit the configuration of a previously added project component (see section 7.3.14 page 232). Depending on the component, the configuration is carried out either via a GUI generated by the component or via a generic GUI.

Update: Click on the button to update the component. The update takes place for the current component version on the FirstSpirit Server. If a later version is available, an update for the respective project can be initiated. Projects might have to be adapted if project components are updated (e.g. configuration adaptation).

7.4.18 Web components



Active Webserver: InternalJetty InternalJetty

Name	Version
FIRSTpersonalisation	1.0

Figure 7-110: Project components – Web components

Web components can be activated for a project here.

Web areas:

Figure 7-111: Web areas in a project

There are three different web areas for each project. The web components can be individually activated and configured for each area via the respective tab.

- Preview: Location of the project contents for which a preview has been requested.
- QA (Staging): Location for the generated project contents
- Production (Live): Location for the deployed project contents
- WebEdit: Configuration for a project local WebEdit instance (see chapter 5.2.2 page 159) (from FirstSpirit Version 4.1).

Web server area:
In each web area it is possible to configure the web components for each project. The currently **active web server** for each area is displayed in the dialogue. An entry for the “internalJetty” is available by default. However, it is also possible to add further web servers. All the web servers which have been configured in the “Web server” area can be selected (see section 7.3.15 page 238). Another web server can be selected via the combo box. Depending on the web server type, various steps are required for changing it:

▪ Internal web server	(see section 7.4.18.1 page 292)
▪ Generic web server	(see section 7.4.18.2 page 293)
▪ External web server	(see section 7.4.18.3 page 293)

Install Click on the button to summarise all web components in the respective web area of the project to a WAR file and to install them depending on the configured web server. If the web component is ready for installation and has not yet been installed the button is active. If the button is deactive, the web components have already been installed. If the selected web server is an external web server or a generic web server (without the required script function), the “Download” button is displayed instead (see section 7.4.18.3).
The Button “Update” will be displayed instead of the Button “Install”, if the web application has already been installed but subsequently the configuration has been changed (see section 7.4.18.6) or a component has been added (see section 7.4.18.4) or deleted (see section 7.4.18.5).

Uninstall Click on the button to delete the web components in the respective web area of the project again. This action is executed for all the web components of the respective area. Depending on the used web server, uninstallation is carried out analogue to the web component installation. If the button is deactive, the web components have not yet been installed.

Download Click on the button to download a WAR file of the application. This file has to be installed manually on the web server (see section 7.3.16.5). The button is only displayed for the configuration of external web servers or generic web servers (without the required script function.

Activate Click on the button to change the configuration of the project-specific web area to the selected web server. This web server is subsequently displayed as an active web server for this area.

7.4.18.1 Configure an internal web server for a web application

The control for the internal web server Jetty is available by default and cannot be changed.

If a different web server (e.g. Tomcat) has been activated for a project-specific web area, the configuration can be reset to the internal web server as follows:

1. Select “InternalJetty” in the combo box.
2. The Install button is activated. Click on the button to unpack the WAR file into the target directory of the web server (Jetty). The web components are then directly registered in the Jetty.
3. The Activate button is active after installation. Click on the button to change the

web area configuration to the selected web server, which is subsequently displayed as an active web server for the area.

4. Click on OK to confirm and save all configuration changes. .

7.4.18.2 Configure a generic web server for a web application

A generic web server can only be selected if a respective web server entity has been added to the server (see section 7.3.15.1 page 239). A generic web server control is not available by default, but can be realised via scripts (see section 7.3.15.2 page 240). If these scripts are unavailable, please proceed as described for the external web server (see section 7.4.18.3).

Carry out the following steps if a generic web server (e.g. Tomcat) is to be activated for a project-specific web area:

1. Select the entry for the desired generic web server in the combo box.
2. The Install button is activated. If the respective function has been provided by a script, click on the button to copy the WAR file, to automatically unpack it and to register the individual components on the web server (the respective script-based web server control has to be configured in the “Server properties” first (see section 7.3.15.2, page 154)).
3. The Activate button is active after installation. Click on the button to change the web area configuration to the generic web server, which is subsequently displayed as an active web server for this area.
4. Click on OK to confirm and save all configuration changes.

7.4.18.3 Configure an external web server for a web application

An external web server can only be selected if a respective web server entity has been added to the server (see section 7.3.15.3 page 242). FirstSpirit does not support an external web server control. This has to be carried out manually (see section 7.3.15.4 page 242). Only a WAR file download is offered for external web servers. Installation on the web server and web component registration have to be carried out manually.

Carry out the following steps if an external web server is to be activated for a project-specific web area:

1. Select the entry for the desired external web server in the combo box.
2. The “Download” button is activated. Click on the button to download an

application WAR file.

3. The WAR file has to be installed manually on the external web server. Installation either occurs manually via the administration user interface of the external web server or automatically from the file system of the web server.
4. After installation, the web area configuration can be changed to the external web server by clicking on Activate. The external web server is subsequently displayed as the active web server for the application.
5. Click on OK to confirm and save all configuration changes.

7.4.18.4 Add a web component

Add: Click on the button to open the “Add” dialogue. All the web components installed on the server are displayed in the list (see section 7.3.14 page 232).

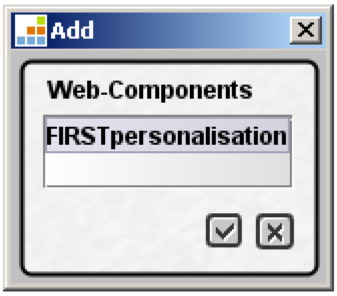


Figure 7-112: Adding a Web-Component

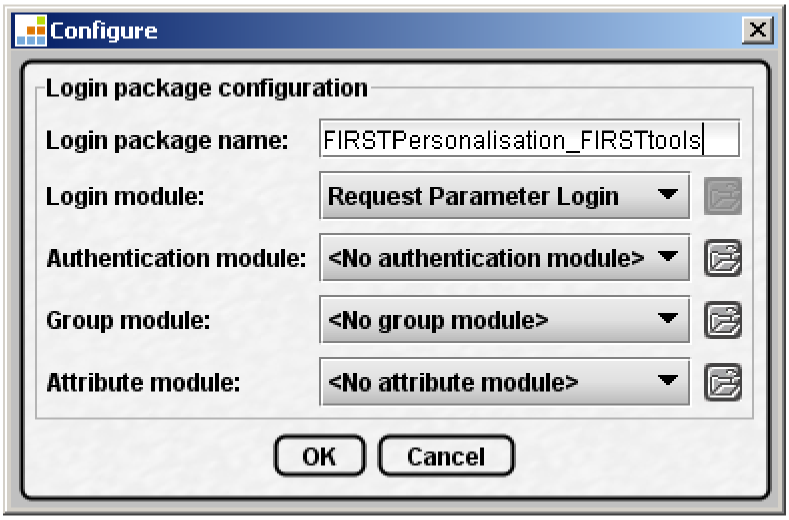
These web components can be added to the individual web areas (preview, staging, live) in the desired project. It is subsequently possible to configure these components either via a GUI generated by the components or via a generic GUI (see the “Configure” button). The components have to be activated after configuration. However, it is only possible to activate or deactivate a component for specific areas in a project (see “Installation").

7.4.18.5 Delete a web component

Delete Click on the button to delete a previously added component. These components are no longer displayed in the table and will no longer be part of the WAR file during the next deployment. Changes will only become effective if deployment occurs on the web server (see “Installation").

7.4.18.6 Configure a web component

Configure Click on the button to edit the configuration for a previously added component (see section 7.3.14 page 232). Depending on the component, configuration is carried out either via a GUI generated by the component or via a generic GUI. The FirstSpirit Personalisation configuration dialogue looks like this:



Login package configuration

Login package name: FIRSTPersonalisation_FIRSTtools

Login module: Request Parameter Login

Authentication module: <No authentication module>

Group module: <No group module>

Attribute module: <No attribute module>

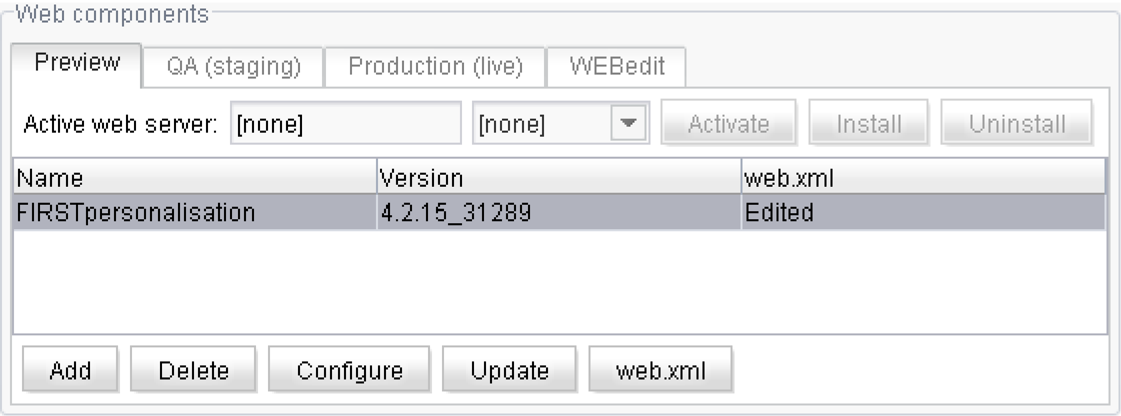
Figure 7-113: Configuration of a web local component (example)

7.4.18.7 Update a web component

Update: Click on the button to update the component. The update takes place for the current component version on the FirstSpirit Server. If a later version is available, an update for the respective project is initiated. Projects might have to be adapted if web components for a project are updated. For example, the Project Configuration might have to be adapted.

7.4.18.8 Edit web.xml

If one or more web components are configured for a web area, a file web.xml is created automatically. This file consists of the single web.xml files of the respective components. The file web.xml can be edited manually. By clicking the button “web.xml“ the dialogue for manual configuration of the file will open. After saving the changes the value "Standard" will no more be displayed in the column "web.xml" but the value "Edited".



Active web server: [none] [none]

Name	Version	web.xml
FIRSTpersonalisation	4.2.15_31289	Edited

Figure 7-114: After manual editing of the file web.xml

7.4.19 Remote projects

All the remote projects configured for the project are listed here. Remote projects are projects from which media or data can be referenced or loaded.

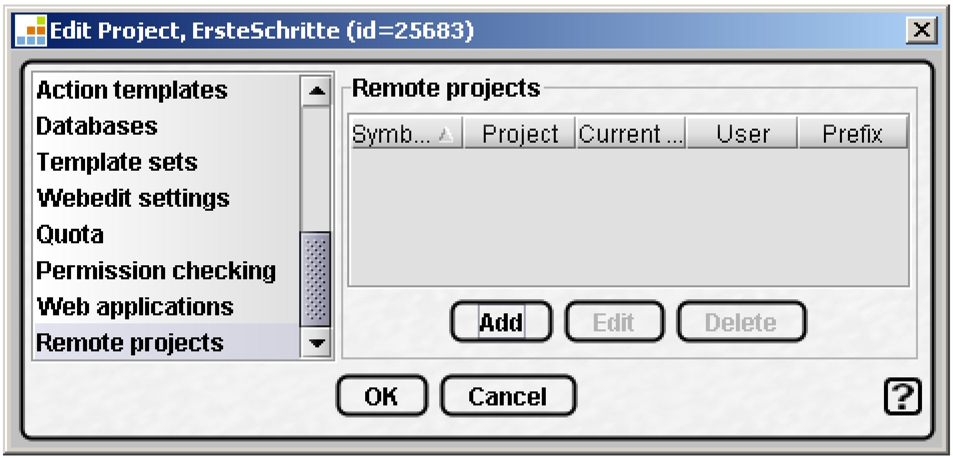
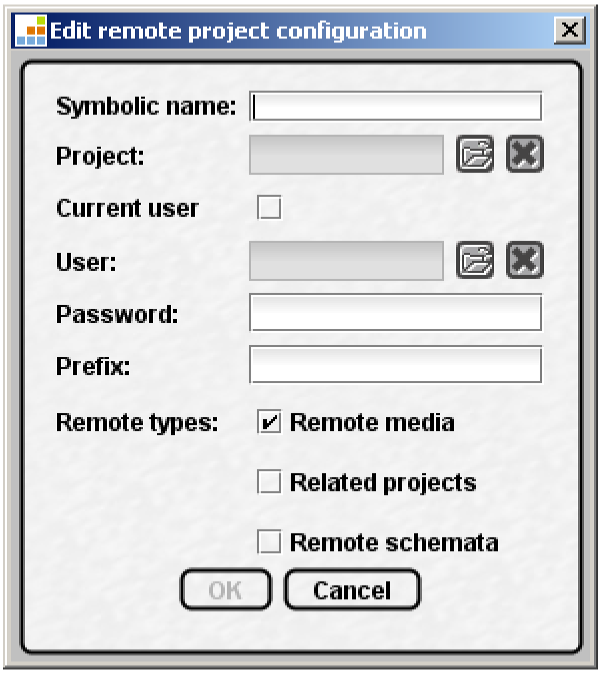


Figure 7-115: Project properties – Remote projects

Click on **Add** to open a window in which the remote project can be configured. If the remote project is already available, click on **Edit** to open the same window.



Edit remote project configuration

Symbolic name:

Project:

Current user □

User:

Password:

Prefix:

Remote types: ☑ Remote media

□ Related projects

□ Remote schemata

Figure 7-116: Configuration of a remote project

Symbolic name: Specify a unique name for the remote project here. This name can be used to reference the desired remote project in the target project.

> **!** *Only projects located on the server with the target project can be configured as remote projects.*

Project: Use the file icon to select the desired project from the project selection list of the server from which media/data is to be referenced.

Current user: During the remote project access process it is checked whether the user has the required permissions for this project if this option has been activated.

User: Click on the file icon to select a user from the user list of the server. The user's permissions are then used to manage remote project access.

Password: Enter the selected user’s password here.

> **!** *This user has to be known as a user in the target project and have at least the permission “visible” for the entire Media-Store or for individual folders in the Media-Store of the remote project.*

Prefix: Enter the name of the remote project again in this field. The prefix is required for URL generation when generating the target project.

Remote category: A category name for a remote project configuration is assigned in this field. The name must be given behind the colon following the key term "category", e.g. "category:mycategory". This enables several remote projects to be grouped together to form a single group. The category can be used in a link configuration in order to enable selection from the defined group of remote projects. As a default, the field is pre-assigned the value "category:default" (default category).

Remote types: Three different types of remote access are possible:

- Remote Media (see chapter 7.4.19.1 page 298).
- Related Projects (see chapter 7.4.19.2 page 298).
- Remote Schemata (see chapter 7.4.19.3 page 299).

! *All types of remote access are licence-dependent additional functions.*

7.4.19.1 Remote Media concept

The aim of the "Remote Media" concept is to create all media in a separate media project and to centrally manage them there. All FirstSpirit projects involved can then access this project's media data (pictures and files) via the remote media access.

Unlike distribution of the media via package management the media do not have to be imported into the projects involved, but instead can be directly referenced via the remote media access. The objects remain physically in the media project but can be used in all the required projects (for further information on package management, see documentation about functionality “PackagePool”).

If a valid licence is available for the function the remote media access can be activated via the FirstSpirit server and project configuration.

For further information on the configuration and use of remote media, refer to the documentation about FirstSpirit Remote Media, the FirstSpirit Manual for Editors, FirstSpirit Manual for Developers (Basic Principles) and the FirstSpirit Online documentation.

7.4.19.2 Related Projects concept

The "Related Projects" concept handles links from one project in another FirstSpirit project. These links can be realised indirectly using the definition of link targets in a FirstSpirit project (via specially configured input components), as well as by means of direct definition within the Site Store.

If a valid licence is available for the function the "Related Projects" access can be activated via the FirstSpirit Server and Project Configuration.

For further information on the configuration and use of related projects, refer to the FirstSpirit Manual for Editors (chap. 11.9), FirstSpirit Manual for Developers (Basic Principles) and the FirstSpirit Online documentation.

7.4.19.3 Remote Schemata concept

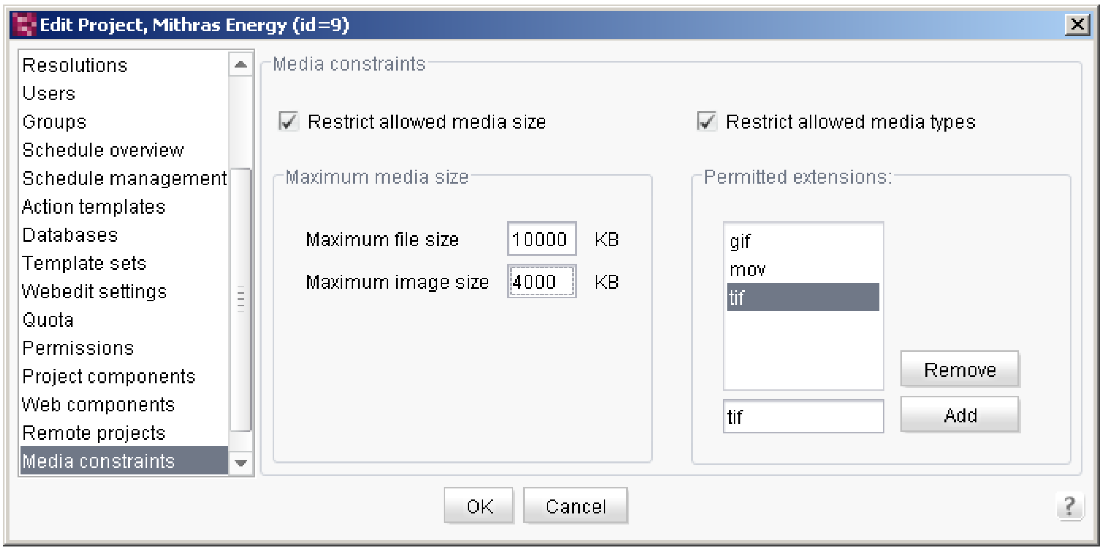
The "Remote Schemata" concept is required for the licence-dependent FirstSpirit module "FirstSpirit Integration", which is used for linking different database technologies in FirstSpirit. The FirstSpirit integration platform can be used to display and edit contents from a database in a web application. FirstSpirit Integration uses database schemata from the FirstSpirit Template Store. These Schemata can be redefined within a project. A graphic editor is available for editing a database schema; this editor can be used to create the required database schema. Each schema can fall back on existing database structures or can create new table structures in an existing database (for further information on using database schemata, see "FirstSpirit Manual for Developers (Basics)" or the "FirstSpirit Online Documentation").

Remote access to database schemata from other FirstSpirit projects is also possible (remote projects) if the remote type "Remote Schemata" has been activated within the remote project configuration.

For further information on FirstSpirit Integration, refer to the relevant module documentation.

7.4.20 Media constraints (from V4.1)

> ! *This function is released for FirstSpirit Version 4.1 and higher only. Screenshots are therefore displayed in the new "LightGray" look & feel. The display can differ slightly in the "Classic" look & feel.*



☑ Restrict allowed media size

☑ Restrict allowed media types

Maximum media size

Maximum file size 10000 KB

Maximum image size 4000 KB

Permitted extensions:

gif
mov
tif

tif

Figure 7-117: Project properties – Media constraints

The uploading of media the Media Store can be limited to specific file sizes and/or formats using the "Restrict allowed media size" and "Restrict allowed media types" options. Both options can each be activated individually or can be combined with each other.

> *The constraints are only applied when new media is uploaded, subsequent restrictions do not affect already created media.*

7.4.20.1 Restrict allowed media size (from V4.1)

Restrict allowed media size: If the checkbox is selected, the input fields of a maximum file size or a maximum picture size become active.

Maximum file size: The maximum file size (in KB) for the uploading of "file" type media in the Media Store can be defined in this field. Files which are larger than the given maximum value can no longer be created.

Maximum image size: The maximum image size (in kb) for the uploading of "picture" type media in the Media Store can be defined in this field. Pictures which are larger than the given maximum value can no longer be created.

Integers only are allowed as values in both fields. Entry of other values, e.g. letters or symbols, is directly prevented and the respective field flashes red.

7.4.20.2 Restrict allowed media types (from V4.1)

In the right-hand half of the media constraints, under the **Restrict allowed media types** selection box of the configuration interface, there is an option for restricting the file format for uploadable media. This is done by entering file name extensions.

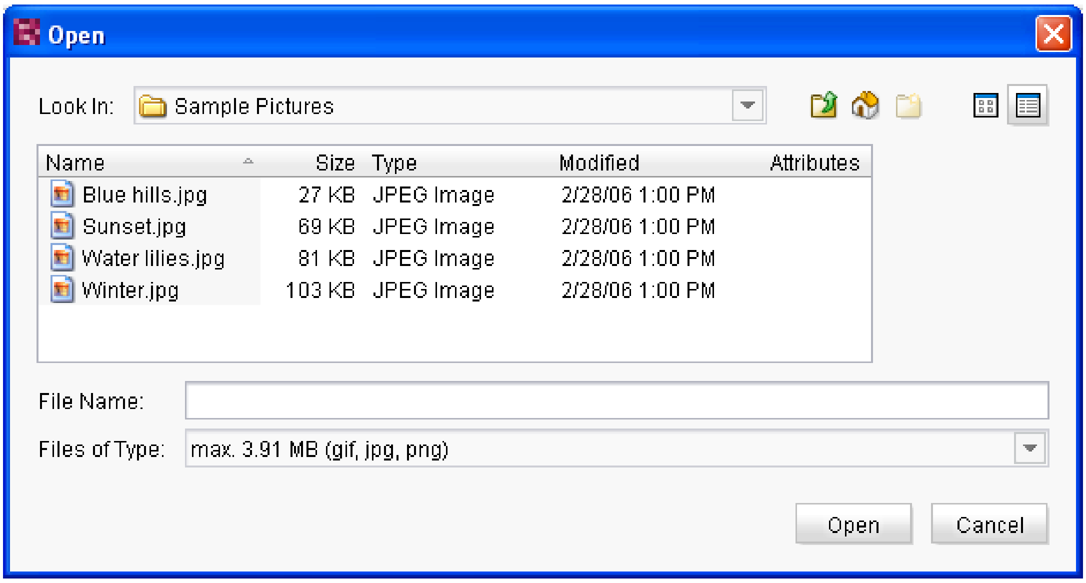
Permitted extensions: The allowed file name extensions can be added and removed here. The file format is only checked with regard to the file name extension and not by analysing the file contents. All extensions must be given here without ".". The extensions are case sensitive, i.e. if the file type is restricted to media type DOC, only files with extension .DOC can be subsequently uploaded, not files with extension .doc. In case of doubt, different notations should be given to enable all files of the required format to be covered.

Press the "**Add**" button to add and save an extension previously defined in the input field to the left of the "Add" button to the list of allowed extensions. The extensions are listed in alphabetical order.

The "**Remove**" button deletes a previously selected file name extension from the list.

7.4.20.3 Display of the constraints in JavaClient (from V4.1)

The media constraints, provided they have been activated in the project configuration, are displayed to the editor in JavaClient's file selection dialog in the form of a filter restriction. This also applies to the Media Wizard. Media which is larger than the defined maximum size and/or has a file name extension which is not allowed is not made available to choose from in the file selection dialog.



Look In: Sample Pictures

Name	Size	Type	Modified	Attributes
Blue hills.jpg	27 KB	JPEG Image	2/28/06 1:00 PM	
Sunset.jpg	69 KB	JPEG Image	2/28/06 1:00 PM	
Water lilies.jpg	81 KB	JPEG Image	2/28/06 1:00 PM	
Winter.jpg	103 KB	JPEG Image	2/28/06 1:00 PM	

File Name:

Files of Type: max. 3.91 MB (gif, jpg, png)

Figure 7-118: Activated file size and file name extension constraints

The constraints are displayed again on the right next to "Files of Type:". In the example in Figure 7-118, files with the file name extensions "jpg", "png" and "gif" up to a size of 3.91 MB may be selected and uploaded.

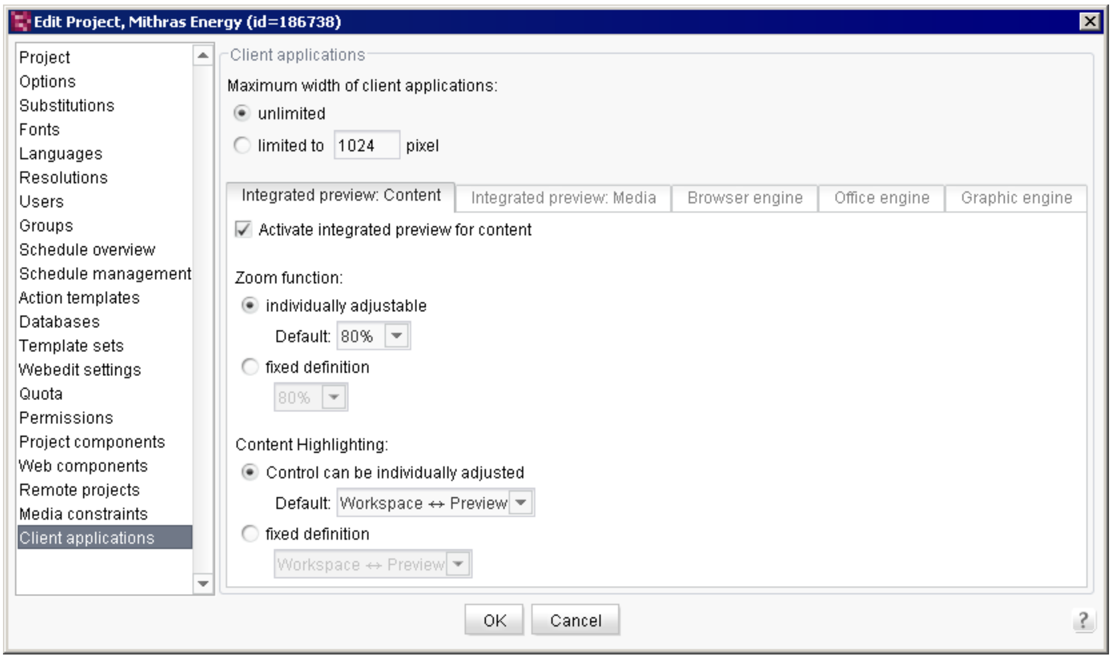
> *A corresponding warning appears if files from the workstation computer's directory structure which are not allowed due on the basis of the media restrictions are added to the Media Store by means of "Drag & Drop". The files are not created in the Media Store.*

7.4.20.4 Activated constraints in WebClient (from V4.1)

Technical implementation of the filter rules as in JavaClient, is not possible in WebClient. As the file selection dialog available in the browser for uploading media is not an independent implementation of FirstSpirit, but instead is permanently integrated in each browser (e.g. Firefox, Mozilla, Internet Explorer, Opera), filtering as in JavaClient is technically not possible. The files are therefore not filtered until after the upload and, if applicable, an error message is issued to the user if the media constraints defined in the project configuration are exceeded.

7.4.21 Client applications

The settings made here affect the "View" menu in JavaClient. To accept and implement the changes, the project concerned must be restarted.



Edit Project, Mithras Energy (id=186738)

Project
Options
Substitutions
Fonts
Languages
Resolutions
Users
Groups
Schedule overview
Schedule management
Action templates
Databases
Template sets
Webedit settings
Quota
Permissions
Project components
Web components
Remote projects
Media constraints
Client applications

Client applications

Maximum width of client applications:

● unlimited

○ limited to 1024 pixel

Integrated preview: Content | Integrated preview: Media | Browser engine | Office engine | Graphic engine

☑ Activate integrated preview for content

Zoom function:

● individually adjustable

Default: 80%

○ fixed definition

80%

Content Highlighting:

● Control can be individually adjusted

Default: Workspace ↔ Preview

○ fixed definition

Workspace ↔ Preview

Figure 7-119: Project properties – Client applications

Maximum width of client applications: Here it is possible to define whether the width is to be limited to a certain number of pixels, or whether the user is to be able to select any value.

7.4.21.1 Integrated preview: Content

Activate integrated preview for content: This options is linked to the "Activate browser engine" option on the "Browser engine" tab: It is only possible for both options to be activated or deactivated at the same time. If this option is deactivated, no changes can be made on this tab.

If this option is activated, the Integrated preview can be used for content in the JavaClient for the selected project, this means the editors can use the "View" / "Integrated Preview – use for content" menu function (see *FirstSpirit Manual for Editors (JavaClient)*, to individually choose whether they want to work with the Integrated preview or not. If the option is deactivated, the "View“ / "Integrated Preview – use for content" menu function is not available for selection in JavaClient in the entire project. This function is activated as a default.

Zoom function

Individually adjustable: If this option is activated, users can independently select the scaling factor for the project concerned. The "Default:" combobox can be used to specify which scaling factor is to be preselected as a default. As a default it is 80%.

Fixed definition: If this option is activated the selected scaling factor is used for the entire project. The editors cannot change the scaling.

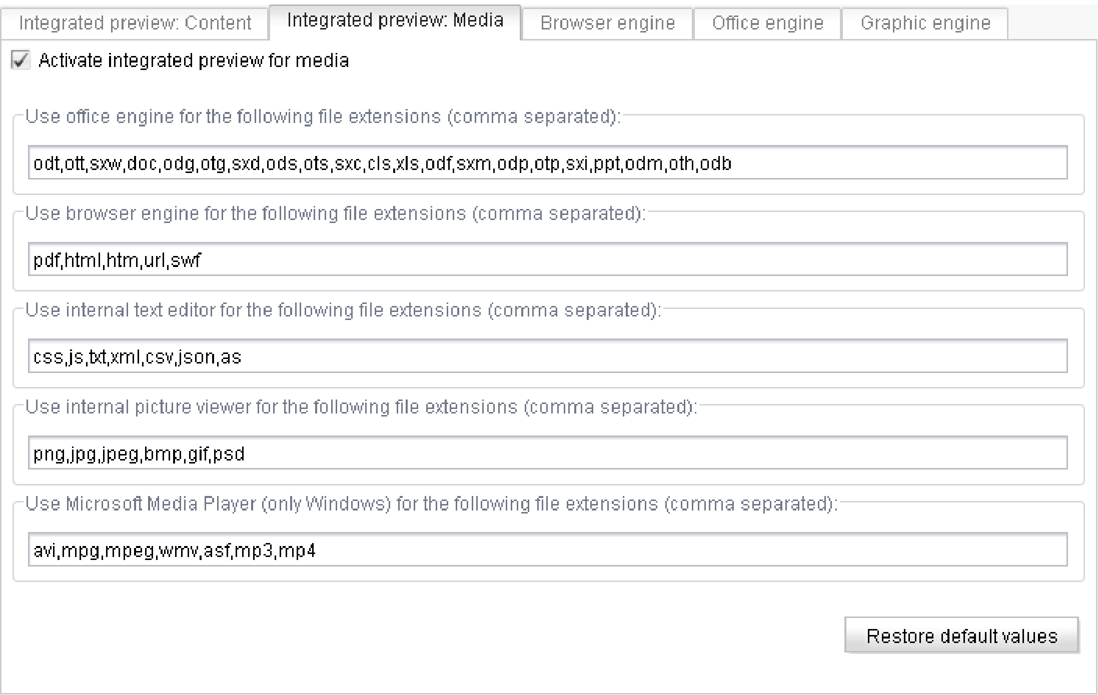
Content Highlighting

Control can be individually adjusted: If this option is activated the editors can individually set the behaviour of the Content Highlighting using the "View" / "Content highlighting control" menu item (see *FirstSpirit Manual for Editors (JavaClient)*). The "Default:" combobox can be used to specify which behaviour is to be preselected as a default. As a default this is "Workspace ↔ Preview".

Fixed definition: If this option is activated the selected behaviour is used for the entire project. The editors cannot change the behaviour.

7.4.21.2 Integrated preview: Media

Here it is possible to define whether the Integrated preview for media is to be available for use in the respective project and which file formats are to be displayed with which application:



☑ Activate integrated preview for media

Use office engine for the following file extensions (comma separated):
odt,ott,sxw,doc,odg,otg,sxd,ods,ots,sxc,cls,xls,odf,sxm,odp,otp,sxi,ppt,odm,oth,odb

Use browser engine for the following file extensions (comma separated):
pdf,html,htm,url,swf

Use internal text editor for the following file extensions (comma separated):
css,js,txt,xml,csv,json,as

Use internal picture viewer for the following file extensions (comma separated):
png,jpg,jpeg,bmp,gif,psd

Use Microsoft Media Player (only Windows) for the following file extensions (comma separated):
avi,mpg,mpeg,wmv,asf,mp3,mp4

Figure 7-120: Project properties – Integrated preview/Media (default settings)

Activate integrated preview for media: This option is linked to the "Activate office engine" option on the "Office engine" tab and the "Activate graphic engine" option on the "Graphic engine" tab. If one of the other options is activated, this option is also active. If this option is deactivated, no changes can be made on this tab.

If this option is activated, the Integrated preview can be used for media in the JavaClient for the selected project, this means the editors can use the "View" / "Integrated Preview – use for media" menu function (see *FirstSpirit Manual for Editors (JavaClient)*), to individually choose whether they want to work with the Integrated preview or not. If the option is deactivated, the "View“ / "Display integrated Preview – for media" menu function is not available for selection in JavaClient in the entire project.

In new projects created with FirstSpirit server version 4.2.400 or higher, the option is activated as a default, so that the following fields can be used to assign which file formats are to be displayed by which applications in the Integrated preview. In projects created with a FirstSpirit server version lower than 4.2.400 the option is deactivated as a default. With version 4.2.438.44956 and higher, the Microsoft Office 2010 formats "docx", "pptx" and "xlsx" belong to the default values shown in Figure 7-120, too.

Use office engine for the following file extensions (comma separated): In this field file extensions can be entered and separated by commas; these are the files to be displayed by the office engine (Microsoft Office, OpenOffice or Google Docs). As a default, file formats of text processing, spreadsheet and presentation programs are preset.

Use browser engine for the following file extensions (comma separated): In this field file extensions can be entered and separated by commas; these are the files to be displayed by the browser engine (Microsoft Internet Explorer and Mozilla Firefox). As a default, file formats that can be displayed by the web browsers or plugins of these browsers are preset here.

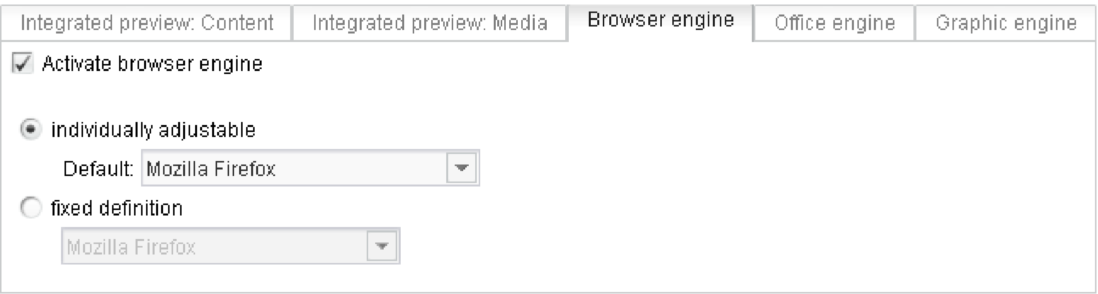
Use internal text editor for the following file extensions (comma separated): In this field file extensions can be entered and separated by commas; these are the files to be displayed by the text editor integrated in FirstSpirit. As a default the file formats preset here are those that can be created and edited using text editors.

Use internal picture viewer for the following file extensions (comma separated): In this field file extensions can be entered and separated by commas; these are the files to be displayed by the graphic engine (simple image processing, Java Image Editor, Picnik, Pixlr). As a default image file formats are preset here.

Use Microsoft Media Player (only Windows) for the following file extensions (comma separated): In this field file extensions can be entered and separated by commas; these are the files to be played back by the Windows Media Player. As a default audio and video file formats are preset here. The Windows Media Player can only be used in conjunction with Microsoft Windows, this field can be greyed out for other operating systems.

Restore default values: Click this button to restore the default settings.

7.4.21.3 Browser engine



☑ Activate browser engine

● individually adjustable

Default: Mozilla Firefox

○ fixed definition

Mozilla Firefox

Figure 7-121: Project properties – Integrated preview/Browser (default settings)

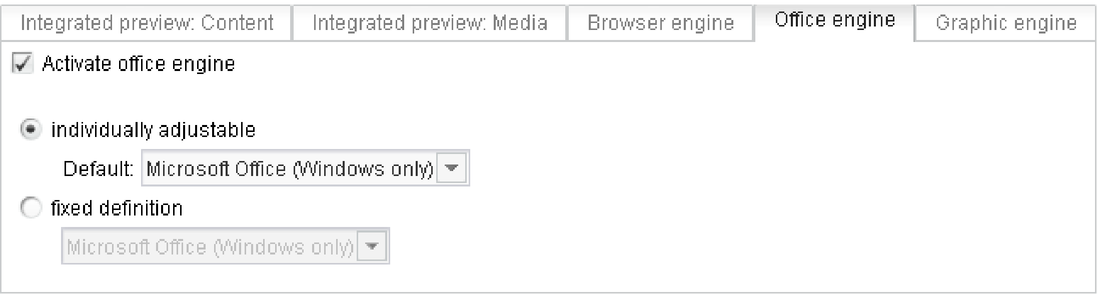
Activate browser engine: This options is linked to the "Activate integrated preview for content" option on the "Integrated preview: Content" tab: It is only possible for both options to be activated or deactivated at the same time. If this option is deactivated, no changes can be made on this tab.
The file formats defined on the "Integrated Preview: Media" tab in the "Browser engine" field (see chapter 7.4.21.2 page 304), are then not displayed by the browser engine integrated in FirstSpirit, but instead in an external application. The option is activated as a default, so that the following radio buttons and comboboxes can be used to define which browser engine is to be used for the file formats defined in the "Browser engine" field:

Individually adjustable: If this option is activated, users can independently select the browser engine in the project concerned. This function is activated as a default. The "Default:" combobox can be used to specify which browser engine is to be preselected as a default in the project. As a default this is Mozilla Firefox.

Fixed definition: If this option is activated, a browser engine can be preset for the project. It is then not possible for the user to make a selection in this project under the "Browser engine" submenu of the "View" menu in JavaClient.

For information on the system requirements and limitations regarding the browser engine, see *FirstSpirit Release Notes Version 4.2R4*, chapter 3.1.3 "Web Browser Integration".

7.4.21.4 Office engine



☑ Activate office engine

● individually adjustable

Default: Microsoft Office (Windows only)

○ fixed definition

Microsoft Office (Windows only)

Figure 7-122: Project properties – Integrated preview/Office (default settings)

A valid licence is required for this function: The `license.OFFICE_INTEGRATION` parameter in the `fs-license.conf` licence file must be set to value 1. Otherwise the "Office engine" tab is greyed out; no settings can be made and office documents cannot be displayed in the Integrated preview of JavaClient.

Activate office engine: This option is linked to the "Activate integrated preview for media" option on the "Integrated preview: Media" tab (see chapter 7.4.21.2 page 304): if the "Activate office engine" option is activated, the "Activate integrated preview for media" option is automatically also activated.
If this "Activate office engine" option is deactivated, no changes can be made on this tab. The file formats defined on the "Integrated Preview: Media" tab in the "Office engine" field (see chapter 7.4.21.2 page 304), are then not displayed by the Integrated preview for media, but instead in an external application.
This option is activated as a default with a valid licence, so that the following radio buttons and comboboxes can be used to define which application is to be used for the file formats defined in the "Office engine" field:

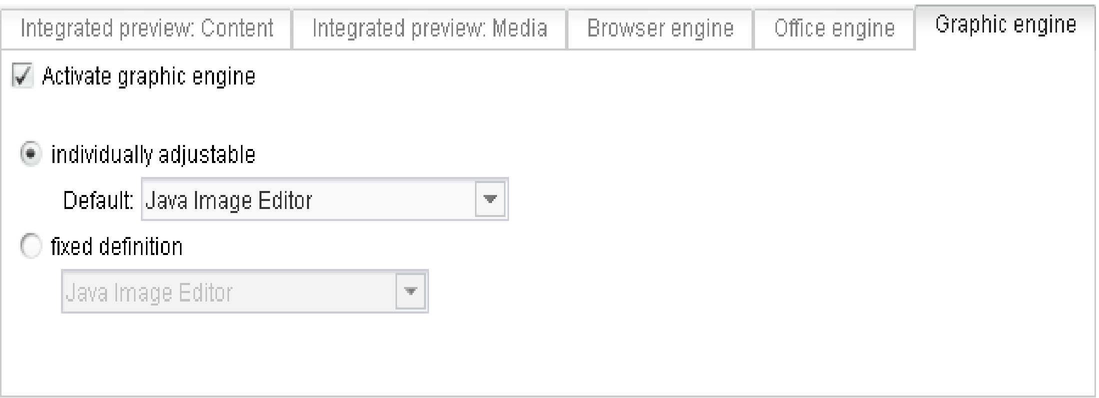
Individually adjustable: If this option is activated, users can independently select the office engine for the project concerned. The "Default:" combobox can be used to specify which office engine is to be preselected as a default in the project. As a default this is Microsoft Office. Microsoft Office cannot be used for non-Windows operating systems, therefore, "OpenOffice" (currently only BETA test phase!) should then be selected here.

Fixed definition: If this option is activated, an office engine can be preset for the project. It is then not possible for the user to make a selection in this project under the "Office engine" submenu of the "View" menu in JavaClient. This option is activated as a default in new installations.

If using applications in the Integrated preview, please note that FirstSpirit provides the interfaces required for the application integration, but in general does not have any influence on the integrated applications themselves. Integrated external applications are not part of the FirstSpirit product. Among other things, this means that responsibility for the function of the integrated applications lies with the manufacturer of the application or with the customer or partner who implements the application. (See also FirstSpirit Release Notes Version 4.2R4, chapter 3 "The FirstSpirit AppCenter").

Further information on the use of Microsoft Office are given in *FirstSpirit Release Notes Version 4.2R4*, chapter 3.1.4 "Microsoft Office Integration", for information on the use of OpenOffice and Google Docs see *FirstSpirit Release Notes Version 4.2R4*, chapter 3.3 "Application integration in the BETA stage".

7.4.21.5 Graphic engine


Integrated preview: Content | Integrated preview: Media | Browser engine | Office engine | Graphic engine

☑ Activate graphic engine

● individually adjustable

Default: Java Image Editor

○ fixed definition

Java Image Editor

Figure 7-123: Project properties – Integrated preview/Graphic

Activate graphic engine: This option is linked to the "Activate integrated preview for media" option on the "Integrated preview: Media" tab (see chapter 7.4.21.2 page 303): if the "Activate graphic engine" option is activated, the "Activate integrated preview for media" option is automatically also activated.

If this "Activate graphic engine" option is deactivated, no changes can be made on this tab. The "Graphic engine" entry in the "View" menu in JavaClient is greyed out and deactivated. Pictures can continue to be edited with the familiar functions available to date (icon).

If this option is activated, the following radio buttons and comboboxes can be used to define which applications are to be used:

Individually adjustable: If this option is activated, users can independently select the graphic engine for the project concerned. The "Default:" combobox can be used to specify which graphic engine is to be preselected as a default in the project.

Fixed definition: If this option is activated, a graphic engine can be preset for the project. It is then not possible for the user to make a selection in this project under the "Graphic engine" submenu of the "View" menu in JavaClient.

! *If using applications in the Integrated preview, please note that FirstSpirit provides the interfaces required for the application integration, but in general does not have any influence on the integrated applications themselves. Integrated external applications are not part of the FirstSpirit product. Among other things, this means that responsibility for the function of the integrated applications lies with the manufacturer of the application or with the customer or partner who implements the application. (See also FirstSpirit Release Notes Version 4.2R4, chapter 3 "The FirstSpirit AppCenter").*

! *The release for use of the* ***Java Image Editor*** *is explicitly given "without function guarantee" for the application itself. I.e. e-Spirit does not provide a guarantee for the image processing functions, neither explicitly nor implicitly, and instead they are released for use "as is". If the use of image processing functions is a critical function for production, external image processing software should be used with the corresponding manufacturer support. Java Image Editor is currently available under MacOS only to a very limited extent.*

! *Use of the* ***Picnik*** *or* ***Pixlr*** *options requires an active connection with the internet. The pictures to be edited are in fact uploaded onto the server of the respective provider and are edited there. This should be taken into account with regard to data protection issues if these applications are used*

For information on the system requirements and limitations regarding the graphic engine, see *FirstSpirit Release Notes Version 4.2R4*, chapter 3.2 "Java Application Integration in 4.2R4 (Java Image Editor)".

7.5 Schedule entry planning

Related actions can be summarised to a schedule entry and started at defined times via FirstSpirit schedule entry planning.

Existing schedule entries and related information can be displayed via FirstSpirit Server and Project Configuration. To achieve this, the tabular “Schedule entry overview” can be called in the Server and Project Configuration (see section 7.5.1).

Use menu item “Schedule entry management” to add new schedule entries and manage existing schedule entries (see section 7.5.2). Das Creation of new schedule entries is subdivided into configuring the schedule entry properties (see section 7.5.4), e.g. specifying the execution time, and adding the desired actions to be executed within the scope of the schedule entry (see section 7.5.5).

The actions have to be created via the Server and Project Configuration first (see section 7.5.6). Server-related actions are defined in the server properties (see section 7.5.6.2) and project-related actions in the project properties of the individual projects (see section 7.5.6.1).

The following project-related actions can now be selected in the project properties:

▪ Archive old project statuses	see section 7.5.9.1 page 328
▪ Execute generation:	see section 7.5.9.2 page 332
▪ Execute deployment:	see section 7.5.9.3 page 337
▪ Execute project backup	see section 7.5.9.5 page 348
▪ Repair references	see section 7.5.9.7 page 350

Only the two action types

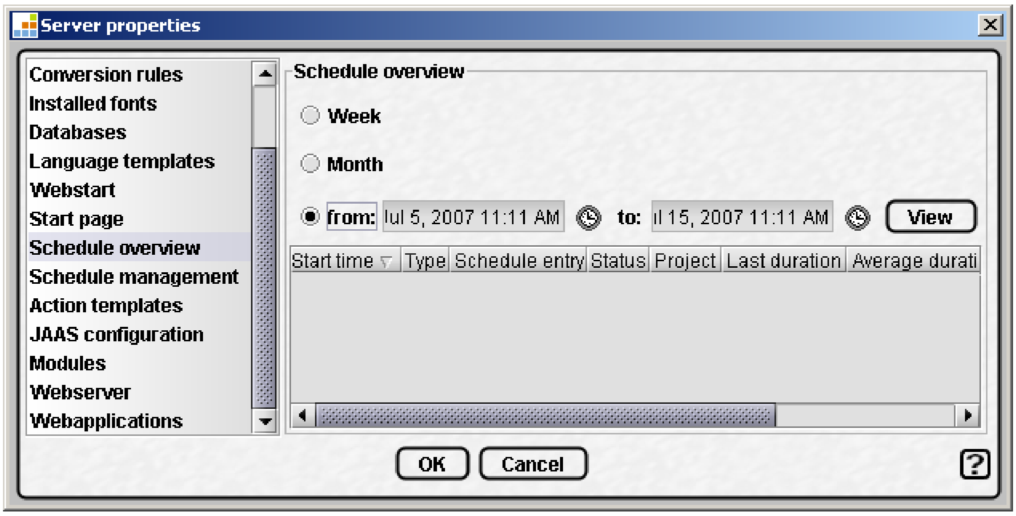
▪ Execute script:	see section 7.5.9.4 page 346
▪ Send email:	see section 7.5.9.6 page 349

can be used in server-related as well as project-related schedule entries.

Created actions can be easily copied from existing schedule entries (see section 7.5.7). If the same actions are to be reused in several schedule entries, it is possible to create an “action template” analogue to the FirstSpirit template concept (see section 7.5.3). An action can be allocated to as many schedule entries as desired via the action template (see section 7.5.8). In this case, action changes are managed centrally via the template.

Besides execution via the Server and Project Configuration, schedule entries can also be executed interactively via the FirstSpirit JavaClient. To achieve this, the user needs the required permissions for interactive execution (see section 7.5.4 page 318), i.e. the user has to be listed in the authorised user list or in a user group from the authorised group list.

7.5.1 Schedule entry overview


Server properties

Conversion rules
Installed fonts
Databases
Language templates
Webstart
Start page
Schedule overview
Schedule management
Action templates
JAAS configuration
Modules
Webserver
Webapplications

Schedule overview

○ Week

○ Month

● from: Jul 5, 2007 11:11 AM to: Jul 15, 2007 11:11 AM View

Start time	Type	Schedule entry	Status	Project	Last duration	Average durati

Figure7-124: Schedule entry overview

The tabular schedule entry overview contains historical dates up to the current date and calculated dates for upcoming, active (not manual) schedule entries (see section 7.5.1.2 page 312). Depending on the location – server properties (see section 7.3.10) or project properties (see section 7.4.9) – only server-dependent or project-dependent schedule entries are displayed.

Schedule entries are displayed in a period of +/- 5 days before and after the current date by default. All the dates of the current week, month or a defined period can be displayed (see section 7.5.1.1 page 311).

Double-click on the desired schedule entry to open the schedule entry details dialogue (see section 7.5.1.3 page 313).

7.5.1.1 Define overview period

Options – week: If this option has been *activated*, all the schedule entries which have already been executed or are still to be executed during the current week (Monday - Sunday) are listed.

Options – month: If this option has been *activated*, all the schedule entries which have already been executed or are still to be executed during the current month are listed.

Options - from ... to: If this option has been *activated*, all the schedule entries which have already been executed or are still to be executed during the defined period are listed. Select the start and end date by clicking on the clock icon 🕒.

[Show schedule entry log file]: If a new option or a different start or end date has been selected, click on the button to confirm the selection. The table is subsequently updated with the desired dates.

7.5.1.2 Tabular schedule entry overview

The table consists of the following columns:

Start time: Start time of the schedule entry execution or start time of the planned execution.

Type: There are various schedule entry execution types (see page 318).

- Manually: Schedule entry to be executed manually
- Schedule entries to be executed automatically are differentiated as described below:

  Once
  Regularly

Schedule entry: Unique name of the schedule entry. The name is defined when adding a new schedule entry.

Status: This column describes the execution status of the schedule entry. The following statuses exist:

- Not started
- Is executed
- Cancelled
- Error
- Finished with errors
- Successful

Depending on the status, the lines of the table appear in various colours.

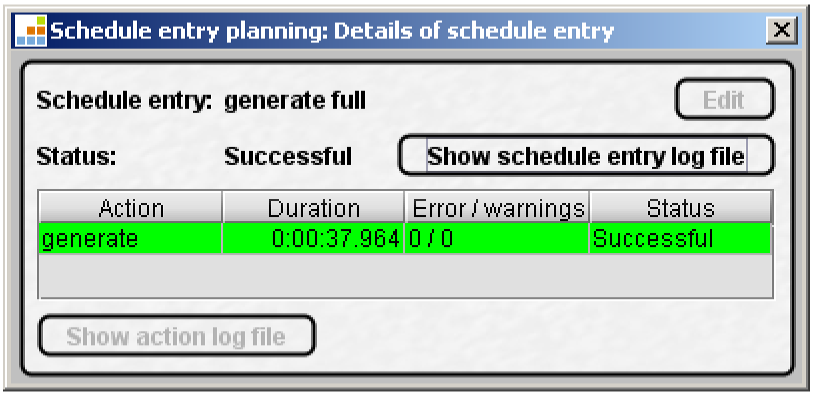
- Green: Successful
- Red: Error

- Yellow: Time conflict

Last duration: The period of the last schedule entry execution is displayed here.

Average duration: The average duration is, just like the “last duration”, statistical information which may be helpful for future scheduling.

7.5.1.3 Schedule entry details


Schedule entry planning: Details of schedule entry
Edit
Show schedule entry log file
Show action log file

Schedule entry: generate full

Status: Successful

Action	Duration	Error / warnings	Status
generate	0:00:37.964	0 / 0	Successful

Figure 7-125: Schedule entry details

Double-click on a schedule entry in the schedule entry overview (see Figure7-124) to open a dialogue with detailed information. This dialogue shows the status of the selected schedule entry as well as the respective actions.

Depending on the execution status, the lines of the table appear in various colours (colour legend: see section 7.5.1.2 page 312).

Edit This button is only activated if the schedule entry has the status “Not started” and the logged-in user has access permissions for the respective project. Click on this button to open the edit dialogue for the schedule entry (see section 7.5.4 page 318).

Show schedule entry log file Click on this button to open the log file of the schedule entry.

Action: The name of the executed action or of the action to be executed.

Duration: Period of the last action execution. If the action has not yet been executed, the cell remains empty.

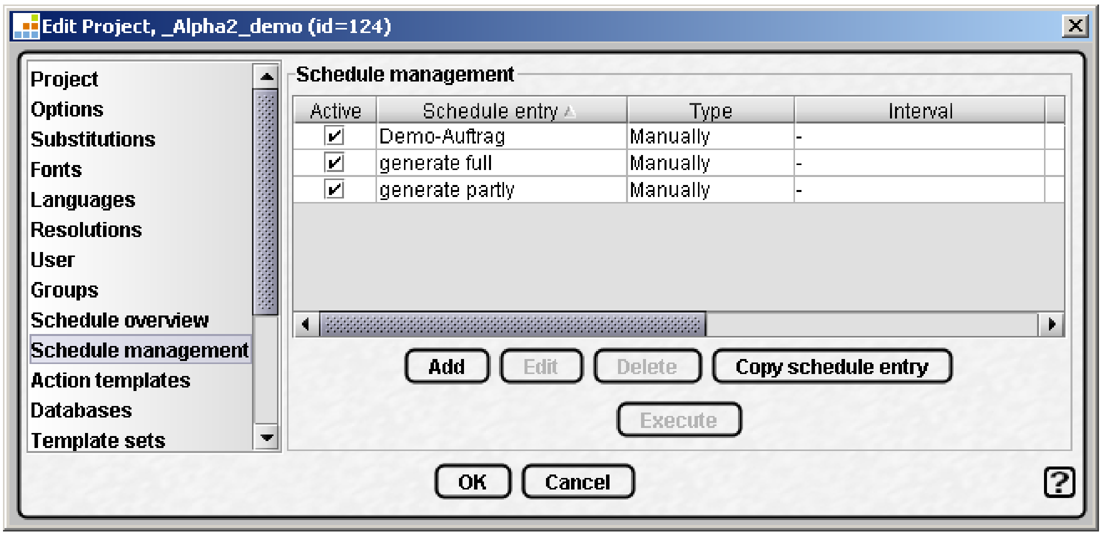
Error / warnings: This column displays the number of errors and warnings which occurred during execution.

Status: This column describes the execution status of the schedule entry (see section 7.5.1.2 page 312).

Show action log file Click on this button to open the log file of the selected action. The log file can also be opened by double-clicking on the desired action.

Schedule entries or actions relating to the schedule entry with the status "Not started" do not have a log file.

7.5.2 Schedule entry management



Active	Schedule entry	Type	Interval
☑	Demo-Auftrag	Manually	-
☑	generate full	Manually	-
☑	generate partly	Manually	-

Figure 7-126: Schedule entry management

Depending on the location (server or project properties), the schedule entry management contains all the schedule entries which have been created either server-dependently (see section 7.3.10) or project-dependently (see section 7.4.9).

Active: If this option has been *deactivated*, this schedule entry will not be automatically executed. From FirstSpirit Version 4.2R4 this applies also for system schedule entries (yellowed out). These schedules can not be started in the JavaClient: the respective menu items are deactivated, namely for project administrators und the server administrator, too. Independent of the selection at "Interactive execution allowed for", schedules can be started by server and project administrators in the JavaClient.

Schedule entry: Unique name of the schedule entry. The name is defined when adding a new schedule entry.

Type: There are various schedule entry execution types (see page 318).

- Manually: Schedule entry to be executed manually
- Schedule entries to be executed automatically are differentiated as described below:

Once
Regularly

Interval: This column displays the execution interval for a schedule entry. This information is only displayed for regularly, automatically executed schedule entries:

- Daily
- Every x minute
- Weekly, *x* times
- Monthly, on each *x day*

> ! *A schedule entry is only saved if a correct start time has been determined. For example: If weekly execution has been configured, a weekday has to be specified.*

Last execution: This column displays the time and date of the last schedule entry execution.

Last duration: This column displays the period of the last schedule entry execution.

Average duration: The average duration is, just like the “last duration”, statistical information which may be helpful for future scheduling.

Schedule entry ID: Unique schedule entry ID. The ID is automatically assigned when creating a new schedule entry.

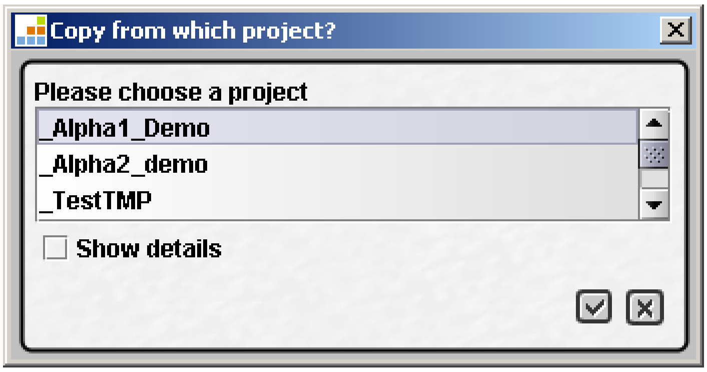
Add Click on this button to create a new schedule entry for this project or the server (see section 7.5.4 page 318).

Edit Click on this button (or double-click on the desired schedule entry) to edit the selected schedule entry. A new dialogue with the respective input mask opens (see section 7.5.4 page 318).

Delete Click on this button to delete the selected schedule entry from this project or the server. Confirm the confirmation prompt. The (yellowed out) system schedule entries cannot be deleted.

Execute Click on the button to execute the desired schedule entry manually, irrespective of the set execution interval.

Copy schedule entry Click on this button to copy a schedule entry from another project and to add it to the current project or to the server-wide schedule entries. At first, select the project from which an existing schedule entry is to be copied. The selection dialog shows all projects on the server for which the logged-in user is recorded as **project administrator**.



Copy from which project?

Please choose a project

_Alpha1_Demo

_Alpha2_demo

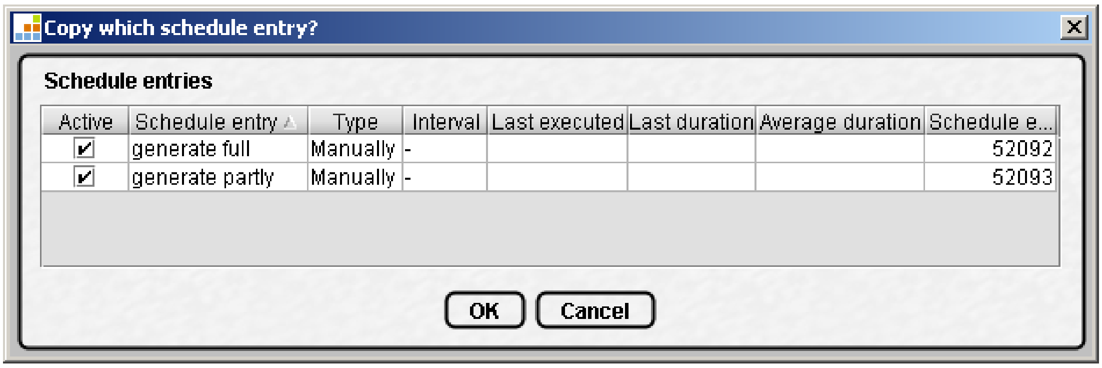
_TestTMP

□ Show details

Figure 7-127: Copy schedule entry – Select a project

This dialogue shows a list of all the projects on the server which can be viewed by the logged-in user. Double-click on the project from which a schedule entry is to be copied.

A new dialogue opens which lists all the schedule entries of the selected project.



Copy which schedule entry?

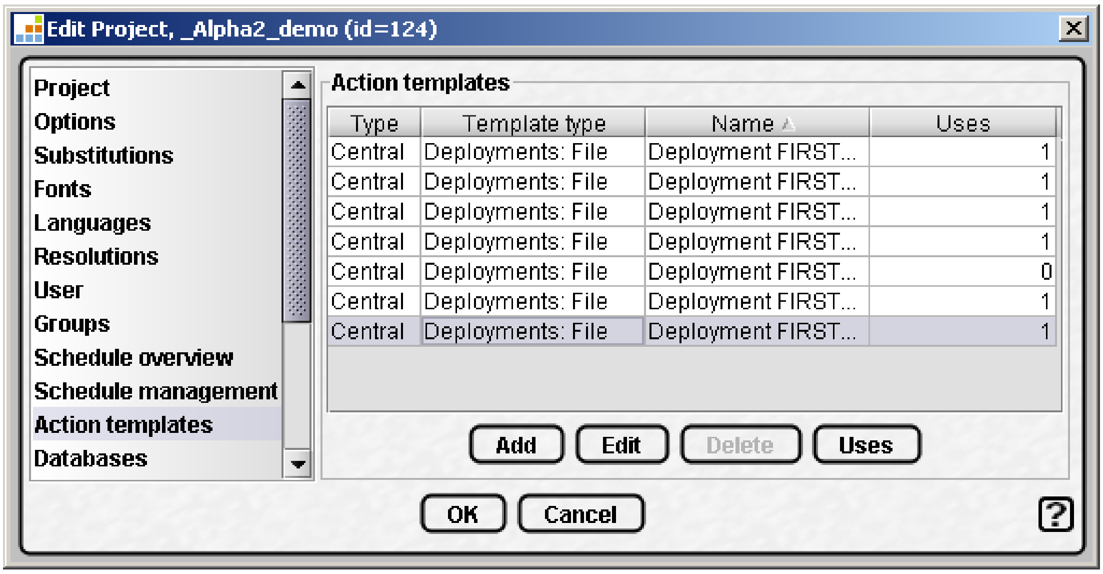
Schedule entries

Active	Schedule entry	Type	Interval	Last executed	Last duration	Average duration	Schedule e...
☑	generate full	Manually	-				52092
☑	generate partly	Manually	-				52093

Figure 7-128: Copy schedule entry – Select the schedule entry to be copied

Double-click on an individual schedule entry to copy it into the edited schedule entry.

7.5.3 Action templates



Type	Template type	Name	Uses
Central	Deployments: File	Deployment FIRST...	1
Central	Deployments: File	Deployment FIRST...	1
Central	Deployments: File	Deployment FIRST...	1
Central	Deployments: File	Deployment FIRST...	1
Central	Deployments: File	Deployment FIRST...	0
Central	Deployments: File	Deployment FIRST...	1
Central	Deployments: File	Deployment FIRST...	1

Figure 7-129: Action templates

Action templates enable the simple management of actions which can be used in several schedule entries – and in other projects – with identical configuration. Action templates can only be edited at the same location as they were created. If action templates should also be available in other projects or server-sided schedule entries, they have to be marked as “public”.

Public: Basically, action templates are always available for the project in which they were created. If this option has been *activated*, the action is additionally provided to all other projects.

Template type: Displays the action type. The following actions are possible:

▪ Execute script:	see section 7.5.9.4 page 346
▪ Execute deployment:	see section 7.5.9.3 page 337
▪ Send email:	see section 7.5.9.6 page 349

Name: Name of the action. Depending on the selected action, the name is automatically assigned (see template type):

- Deployment
- Script
- Email

Uses: The number of schedule entries which use this action template.

Add Click on this button to create a new action template. To achieve this, the action type has to be selected first (see section 7.5.6 page 324).

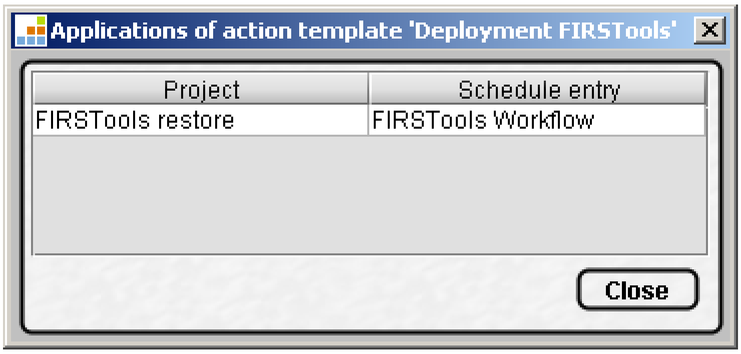
Edit Click on this button to edit the selected action template. A new dialogue with the respective input mask opens (see section 7.5.9 page 328 and section 7.5.10 page 350).

> **!** *Changes to an action template affect all the schedule entries which use this template.*

Delete Click on this button to delete the selected action template. Confirm the confirmation prompt.

> **!** *An action template can only be deleted if it is no longer used in any of the schedule entries.*

Uses Click on this button to open a new dialogue which displays all the schedule entries which use this action template:



Applications of action template 'Deployment FIRSTools'

Project	Schedule entry
FIRSTools restore	FIRSTools Workflow

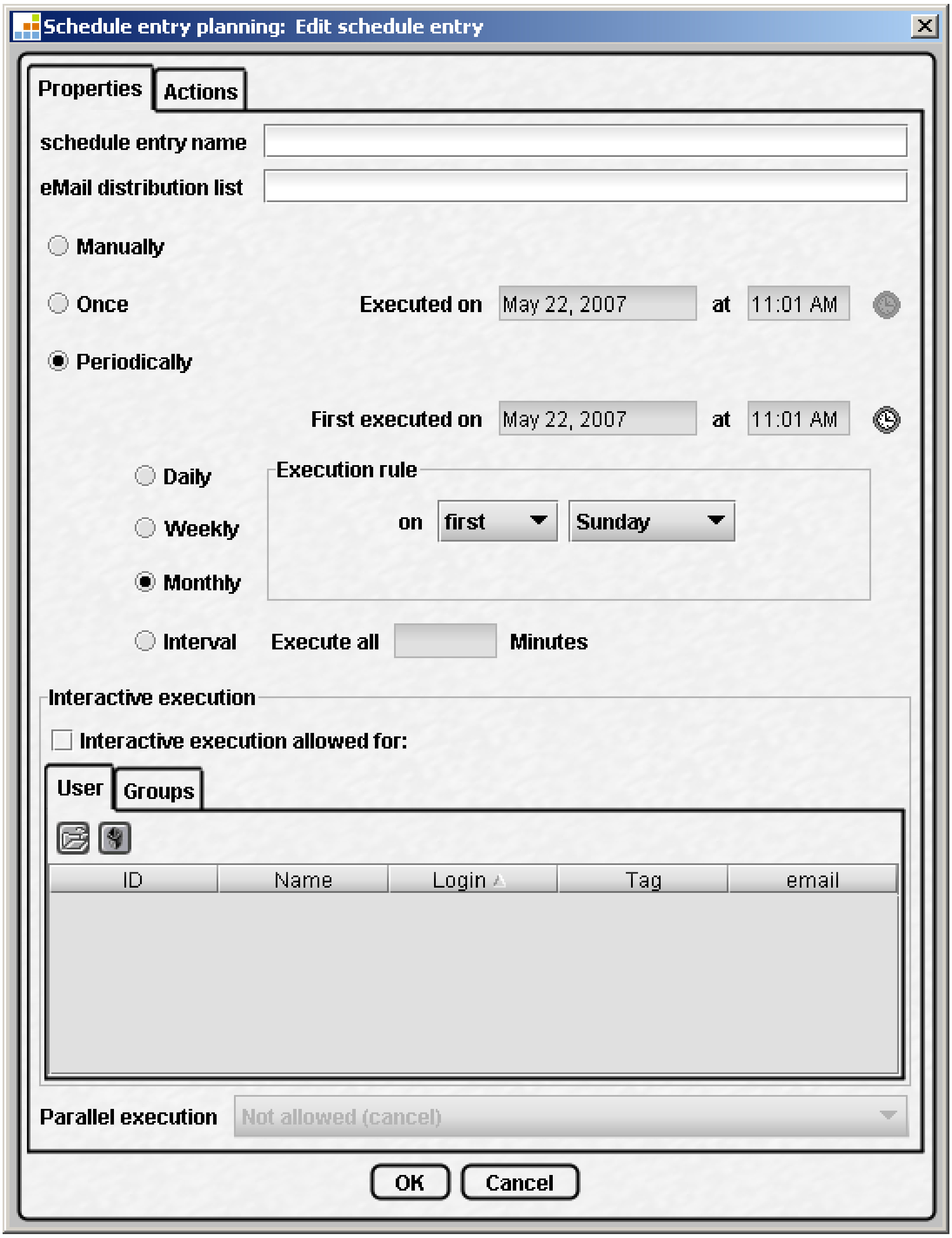
Figure 7-130: Display of selected template utilisation

Project: Name of the project to which the schedule entry belongs.

Schedule entry: Name of the schedule entry which uses the action template.

7.5.4 Add / Edit a schedule entry (properties tab)

Click on “Add” or “Edit” in the schedule entry management (or double-click on the desired table entry) to open the “Edit schedule entry” dialogue window. The following schedule entry properties can be defined in the “Properties” tab.



Schedule entry planning: Edit schedule entry

Properties | Actions

schedule entry name

eMail distribution list

○ Manually

○ Once Executed on May 22, 2007 at 11:01 AM

● Periodically

First executed on May 22, 2007 at 11:01 AM

○ Daily
○ Weekly
● Monthly

Execution rule: on first Sunday

○ Interval Execute all Minutes

Interactive execution

☐ Interactive execution allowed for:

User | Groups

ID	Name	Login	Tag	email

Parallel execution: Not allowed (cancel)

OK Cancel

Figure 7-131: Edit schedule entry – Properties

Schedule entry name: Unique name of the schedule entry. The schedule entry is displayed under this name in the Server and Project Configuration (see Figure7-124 and Figure 7-126) and in the JavaClient.

eMail distribution list: Here it is possible to specify email addresses which are also inherited to embedded "email actions" (see section 7.5.9.6 page 349) and used by them.

Execution type: Use the radio buttons to select one of the following execution types. The execution type "Manually" is selected by default.

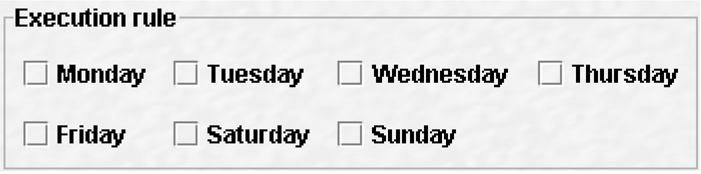
- **Manually**: If this execution type is selected, the schedule entry can only be started manually. An execution time cannot, therefore, be specified. The respective fields are deactivated.

- **Once:** In contrast to manual execution, the once-only schedule entry execution is automatically started at a specific time. Click on the clock icon 🕓 to define the execution time.

- **Periodically:** If this execution type is selected, the schedule entry is automatically executed at regular intervals. Click on the clock icon 🕓 to set the first execution time and the execution rule. A schedule entry can only be saved if a valid time has been determined (for to this reason, there are preset default values which can be changed as desired). Regular schedule entries can be executed according to the following rules:

  - Daily
    A daily executed schedule entry does not require an execution rule since it is always executed at a set time.

  - Weekly
    The weekday on which the schedule entry is to be executed has to be defined for a weekly executed schedule entry. (Default value: Monday).

    Execution rule

    □ Monday □ Tuesday □ Wednesday □ Thursday

    □ Friday □ Saturday □ Sunday

    Figure 7-132: Execution rule– Weekly execution

  - Monthly
    The day of the month on which the schedule entry is to be executed has to be defined for a monthly executed schedule entry. (Default value: On the first Monday of the month).

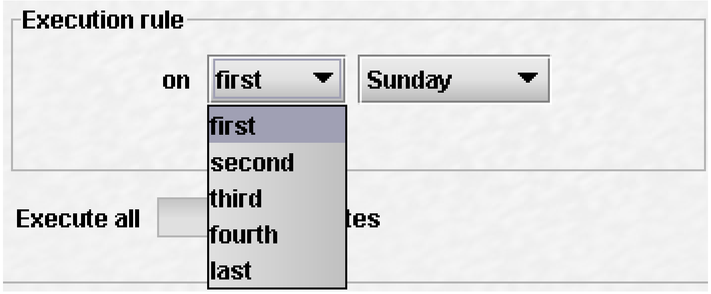


Figure 7-133: Execution rule– Monthly execution

- Interval

  If the radio button “Interval” is selected, only the interval between the actual executions has to be specified in minutes (default value: 60 min).

Figure 7-134: Execution rule – Interval execution

Depending on the selected execution type, further schedule entry properties can be edited (see Figure 7-131):

Interactive execution: If this option is *activated*, all the selected users or users of one of the selected groups can execute this schedule entry interactively. Interactive schedule entry execution is only possible via the JavaClient.

User

This table lists all the users who are allowed to execute this schedule entry interactively (via the JavaClient).

Add Click on this button to open a dialogue which shows all the current project users. Already selected users are highlighted in boldface.

Delete Click on this button to delete all the selected users from the table. These users no longer have the permission to execute this schedule entry via the JavaClient.

Groups

This table lists all the user groups which are allowed to execute this schedule entry interactively.

Add Click on this button to open a dialogue which shows all the current project user groups. Already selected groups are highlighted in boldface.

Delete Click on this button to delete all the selected user groups from the table.

Parallel execution:

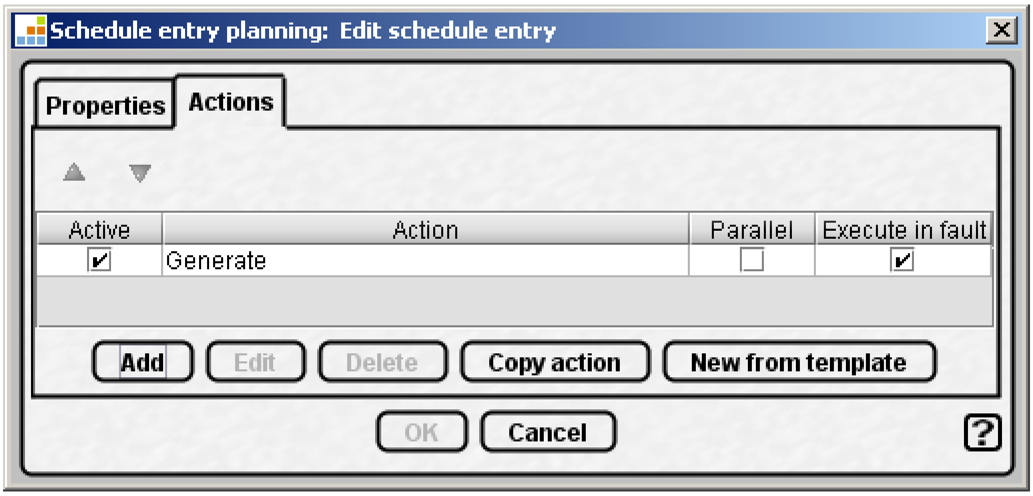
- Not allowed (cancel):

  If this option is selected, parallel execution of this schedule entry is not allowed. If a schedule entry is already running, an attempt to start the execution is blocked.
- Allowed (parallel execution):

  If this option is selected, execution of the schedule entry is started immediately after the request (even if another schedule entry is already running in parallel).
- Not allowed (successive execution):

  If this option is selected, parallel execution of the schedule entry is not allowed. However, the schedule entry is automatically started after completion of the running execution (if available).

7.5.5 Add / Edit a schedule entry (actions tab)



Active	Action	Parallel	Execute in fault
☑	Generate	□	☑

Figure 7-135: Edit schedule entry – Actions

Click on “Add” or “Edit” in the schedule entry management (or double-click on the desired table entry) to open the “Edit schedule entry” dialogue window. Actions can be added to the schedule entry, edited or deleted in the “Actions” tab.

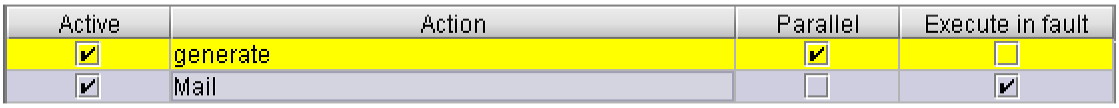
All related actions are displayed in the sequence in which they are executed. This sequence can be changed by selecting an action – i.e. a line of the table - and moving it up or down via the △ ▽ buttons.

The table consists of three columns, which show the name and two action properities important for execution.

Active: If this option is *activated*, the action is also executed during schedule entry execution; if it is *deactivated,* the action is skipped.

Action: Name of the action.

Parallel: If this option is *activated*, the action can be executed at the same time as another action in a schedule entry. This setting is only recommended if several successive actions should also be released for parallel execution and if these actions are logically independent of each other. If this option is activated, the line is highlighted yellow in the overview.



Active	Action	Parallel	Execute in fault
☑	generate	☑	□
☑	Mail	□	☑

Execute in fault: If this option is activated the respective subsequent action is executed even after a preceding faulty action. For example, if the action "generate" fails, the "Mail" action is still executed.

Add Click on this button to create a new action for this schedule entry (see section 7.5.6 page 324).

Edit Click on this button (or double-click on the table entry) to open a new dialogue for editing the selected action (see section 7.5.9 page 328 and section 7.5.10 page 350).

Delete Click on this button to delete the selected action from this schedule entry. Confirm the confirmation prompt.

Copy action Click on this button to copy a schedule entry from another project and to add it to the current schedule entry (see section 7.5.7 page 325).

New from template Click on this button to add an action which has previously been defined in the action templates (see section 7.5.3 page 317) to the current schedule entry (see section 7.5.8 page 326).

Actions which have been added to the schedule entry via a template cannot be edited here. This action can only be changed via the respective template in the "Action templates" area (see section 7.5.3 page 317).

7.5.6 Add actions to a schedule entry

Click on "Add" in the schedule entry management (or double-click on the desired table entry) to open the "New activity" dialogue window.

As already described in the introduction of section 7.5 it is differentiated between project-related and server-related schedule entries when creating an action:

- Add a project-related action (see section 7.5.6.1).
- Add a server-related action (see section 7.5.6.2).

7.5.6.1 Create a project-related action

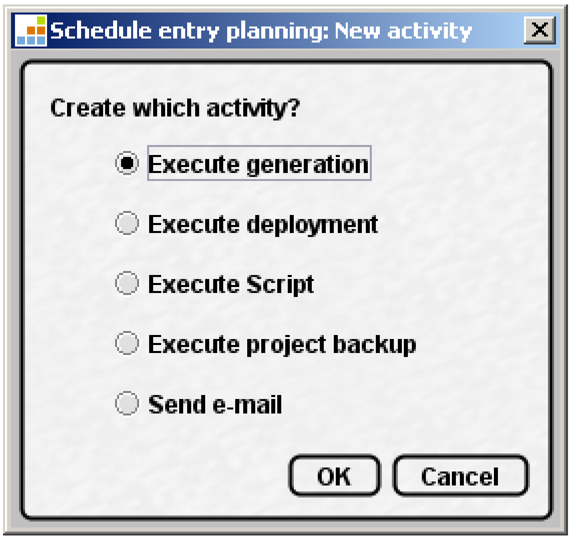


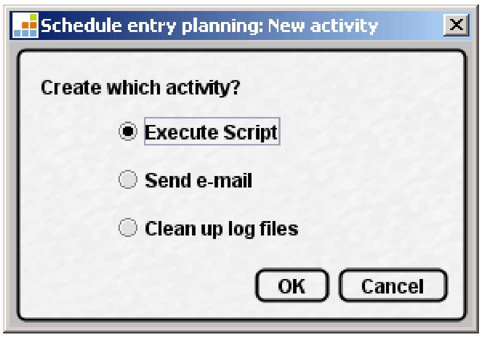
Figure 7-136: Create a project-related action

Depending on the selected action, the input mask of the selected action type opens:

Archive old project statuses	see section 7.5.9.1 page 328
Execute generation:	see section 7.5.9.2 page 332
Execute deployment:	see section 7.5.9.3 page 337
Execute script:	see section 7.5.9.4 page 346
Execute project backup	see section 7.5.9.5 page 348
Send email:	see section 7.5.9.6 page 349

- Repair references: see section 7.5.9.7 page 350

7.5.6.2 Create a server-related action


Schedule entry planning: New activity

Create which activity?

● Execute Script

○ Send e-mail

○ Clean up log files

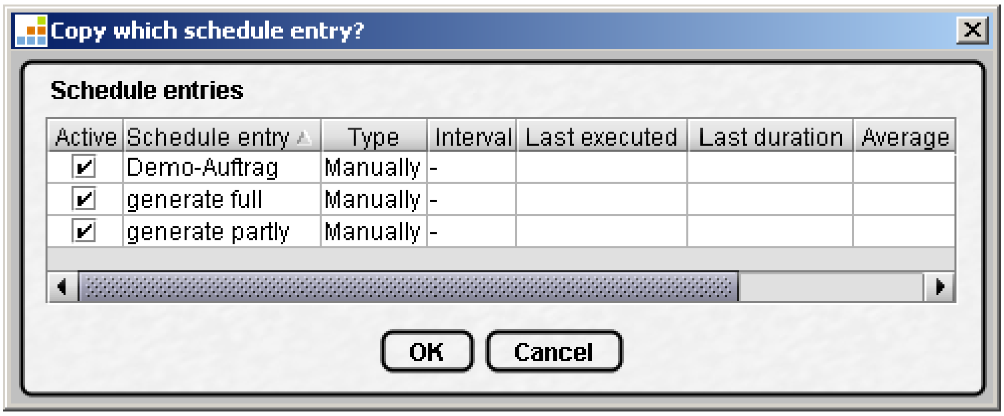
Figure 7-137: Create a server-related action

Depending on the selection, the input mask of the selected action type opens.

- Execute script: see section 7.5.9.4 page 346
- Send email: see section 7.5.9.6 page 349

7.5.7 Copy actions from another schedule entry

Use the “Copy action” button in the “Edit schedule entry” dialogue window (see section 7.5.5 page 322) to copy an already created action from another schedule entry into the currently edited schedule entry. The following dialogue window opens:


Copy which schedule entry?

Schedule entries

Active	Schedule entry	Type	Interval	Last executed	Last duration	Average
☑	Demo-Auftrag	Manually	-			
☑	generate full	Manually	-			
☑	generate partly	Manually	-			

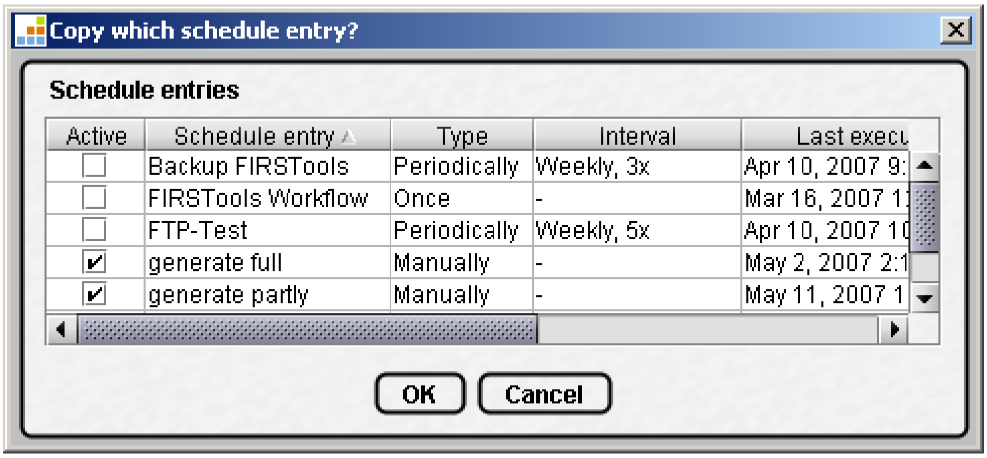
Figure 7-138: Copy action

This dialogue shows a table of all the available schedule entries:

- In the project properties: All project-related schedule entries.
- In the server properties: All server-related schedule entries.

Double-click on the schedule entry from which one or more actions are to be copied.

A new dialogue opens and lists all the actions of the selected schedule entry:



Copy which schedule entry?

Schedule entries

Active	Schedule entry	Type	Interval	Last execu
□	Backup FIRSTools	Periodically	Weekly, 3x	Apr 10, 2007 9:
□	FIRSTools Workflow	Once	-	Mar 16, 2007 1
□	FTP-Test	Periodically	Weekly, 5x	Apr 10, 2007 1
☑	generate full	Manually	-	May 2, 2007 2:1
☑	generate partly	Manually	-	May 11, 2007 1

Figure 7-139: Copy action – Select the action to be copied

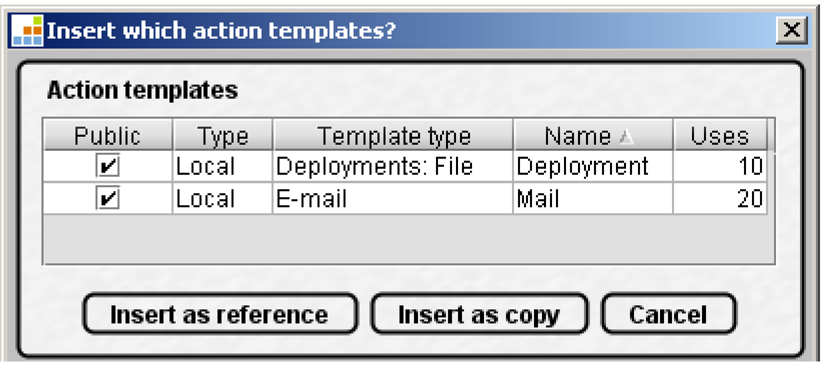
Double-click on an individual schedule entry to copy it into the edited schedule entry.

It is also possible to copy several actions simultaneously. To achieve this, select all the actions while pressing the CTRL key. After clicking on OK, all the selected actions are copied and added to the schedule entry.

7.5.8 Add actions via action templates

If actions are to be reused with the same configuration in several schedule entries, it is possible to create “action templates” analogue to the FirstSpirit template concept (see section 7.5.3). An action can be allocated to as many schedule entries as desired via the action template.

Use the “Copy action” button in the “Edit schedule entry” dialogue window (see section 7.5.5 page 322) to create an action based on an already created action template and to add it to the currently edited schedule entry. The following dialogue window opens:



Public	Type	Template type	Name	Uses
☑	Local	Deployments: File	Deployment	10
☑	Local	E-mail	Mail	20

Figure 7-140: Add action – Select an action template

Figure 7-140 shows the available action templates.

Only action templates which have been defined in the local action template management or, e.g., in the action template management of another project and marked as “public” are available (see section 7.5.3 page 317). Double-click on individual actions to add them to the schedule entry.

! *Besides creating individual actions via the action template, it is also possible to insert several actions simultaneously. Select all the desired templates while pressing the CTRL key.*

Insert as reference Click on the button to add an action to the schedule entry as a reference to an action template. Actions which have been inserted as a reference to an action template can only be edited in the template management and not in a schedule entry. Referenced actions are italicised in the table.

Insert as copy Click on the button to add an action to the schedule entry as a copy of the selected action template. Action copies can be edited as normal.

Cancel Click on the button to close the dialogue window. An action is not added to the schedule entry.

7.5.9 Project-related actions

Project-related actions are created in the project properties of the individual projects and added to the project-related schedule entries.

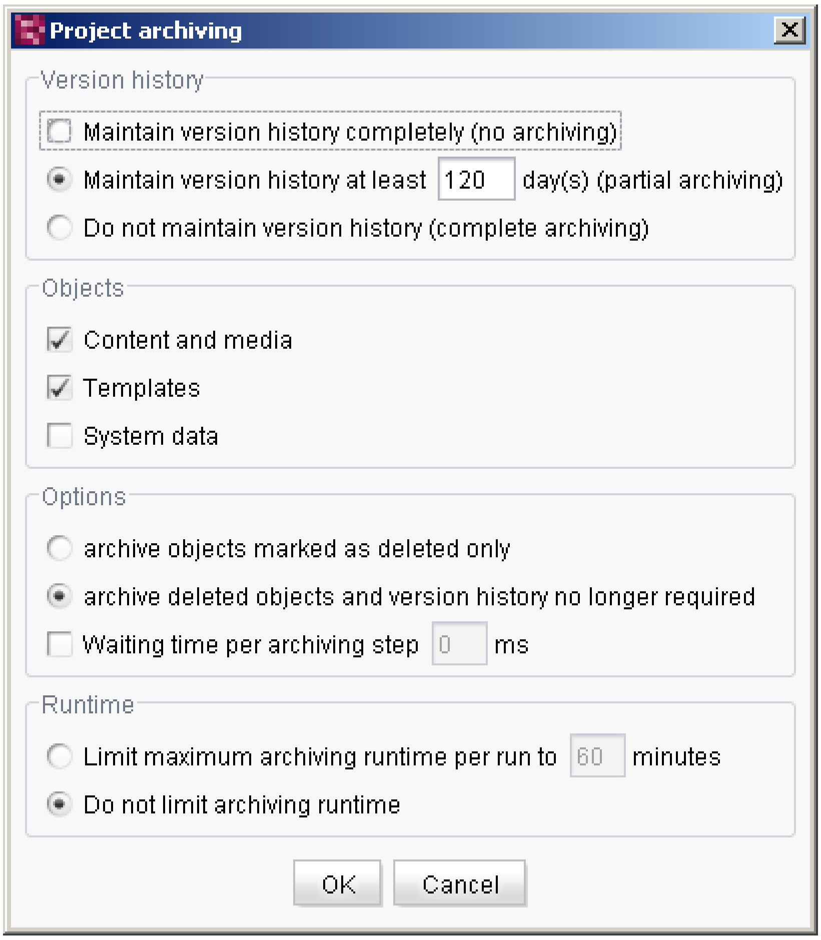
The following actions are available:

- Archive old project states see section 7.5.9.1 page 328
- Execute generation see section 7.5.9.2 page 332
- Execute deployment see section 7.5.9.3 page 337
- Execute script: see section 7.5.9.4 page 346
- Execute project backup see section 7.5.9.5 page 348
- Send email: see section 7.5.9.6 page 349
- Repair references see section 7.5.9.7 page 350

7.5.9.1 Archive old project states (from V4.1)

This function is released for FirstSpirit Version 4.1 and higher only. Screenshots are therefore displayed in the new "LightGray" look & feel. The display can differ slightly in the "Classic" look & feel.

For information about the use of this function in comparison with the use of the module “FirstSpirit EnterpriseBackup“ see also FirstSpirit Release Notes 4.2, Chapter “Long-term archiving and backup in FirstSpirit“.



Project archiving

Version history

☐ Maintain version history completely (no archiving)

● Maintain version history at least 120 day(s) (partial archiving)

○ Do not maintain version history (complete archiving)

Objects

☑ Content and media

☑ Templates

☐ System data

Options

○ archive objects marked as deleted only

● archive deleted objects and version history no longer required

☐ Waiting time per archiving step 0 ms

Runtime

○ Limit maximum archiving runtime per run to 60 minutes

● Do not limit archiving runtime

Figure 7-141: Create schedule – Project archiving

FirstSpirit uses repositories for archiving and versioning of project data. A repository for each project exists in the server directory `data\projects\`. With each action taken in JavaClient, data is written into the repository. This applies both to actions which create new elements and to actions which delete elements. In addition, even deleted elements are not removed from the repository. As this means that new data is continuously added to the repository, it becomes larger and an increasing amount of hard disk space is required.

The "Archive old project states" schedule is used to archive the selected project so that data which is no longer required is swapped out of the project, thereby reducing loading times and increasing the performance of the FirstSpirit server. To this end, data from the repositories is moved into archive files. Archive files which are no longer required can be subsequently deleted to finally release memory space on the hard disk.

During archiving, a folder for each project is created in the server `archive` directory and the data to be archived is moved into an archive file (format `tar.gz`) in the corresponding project folder. However, the data can be subsequently displayed using the "Archive" function in the server and project properties and if necessary can be installed (see Chapter 7.2.3.8 page 198).

Further information on the FirstSpirit archiving concept is given in Chapter 7.9 from page 370*.*

> ! *The archiving function should be used prudently. Because, depending on the settings, the version history is no longer fully available following archiving. It may therefore no longer be possible to flawlessly restore older revisions. During an export (see Chapter 7.2.3.3 page 195) following archiving, only the currently available project status is exported, without any archives or archived project statuses that may exist.*

The following options can be used to define the archiving criteria:

Version history: In this area it is possible to define the period for which archiving is to be performed or the period for which the version history is to be retained.

Maintain version history completely (no archiving): If this option is selected, no archiving is performed and an archive file is not created. The version history is completely retained.

Maintain version history at least 120 day(s) (partial archiving): This option can be used to define the period for which the whole version history is to be retained with all revisions. For example, if 120 days are set, the archiving only includes revisions which are **older than** 120 days at the start of the archiving schedule. All changes which are **younger than** 120 days at the start of the archiving schedule can still be completely traced, even after archiving.

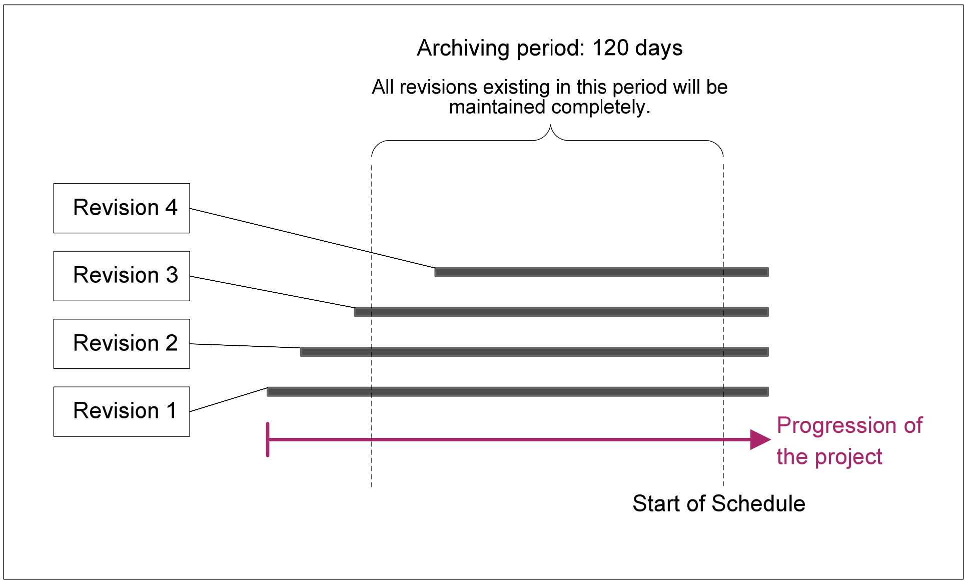

Archiving period: 120 days
All revisions existing in this period will be maintained completely.
Revision 4
Revision 3
Revision 2
Revision 1
Progression of the project
Start of Schedule

Figure 7-142: Partial archiving over 120 days

Do not maintain version history (complete archiving): If this option is selected, the whole version history is included when the schedule is performed. All data no longer required (see Chapter 7.9.3 page 372) is moved into the archive file.

<u>Objects:</u>
This area can be used to define the type of data to be archived.

Content, media and data sources: If this option is selected, all content of the page, media and content store of a project are archived (i.e. everything except templates).

Templates: If this option is selected, templates are archived. The option can only be selected in conjunction with "Content, media and data sources".

System data: System data is information which is generated by the system with each action in JavaClient (e.g. creation or deletion objects, releases, etc.) (see also Chapter 7.9.1 page 370 and Chapter 7.9.2 page 371). If this option is selected, system data which is no longer used is archived. The system data also includes revisions. The archived revisions are subsequently displayed using the "Archive" function on the "Revisions" tab (see Figure 7-20).

! *From FirstSpirit Version 4.2R4, closed tasks are now also archived. To do this, the "Page, Media and Content"* ***and*** *"System Data" checkboxes must be activated. In this case, all files belonging to a task which have been closed by a certain time lying within the period to be archived are archived.*

Options:
This area can be used to define whether objects only or version history entries no longer required too are to be archived.

archive objects marked as deleted only: If this option is selected, only objects which have been deleted are archived.

archive deleted objects and version history no longer required: If this option is selected, in addition to the deleted objects, version history entries no longer required are also archived. This means that the objects' version history is reduced. However, in each case, the complete version history from the partial archiving period is retained (provided this option is selected), the revisions of the last release status (provided available) and the current editing status are also retained.

Waiting time per archiving step: This value can be used to set a pause (in milliseconds) to be introduced between each of the archiving steps of extensive archiving. In this way, the server load is reduced during archiving.

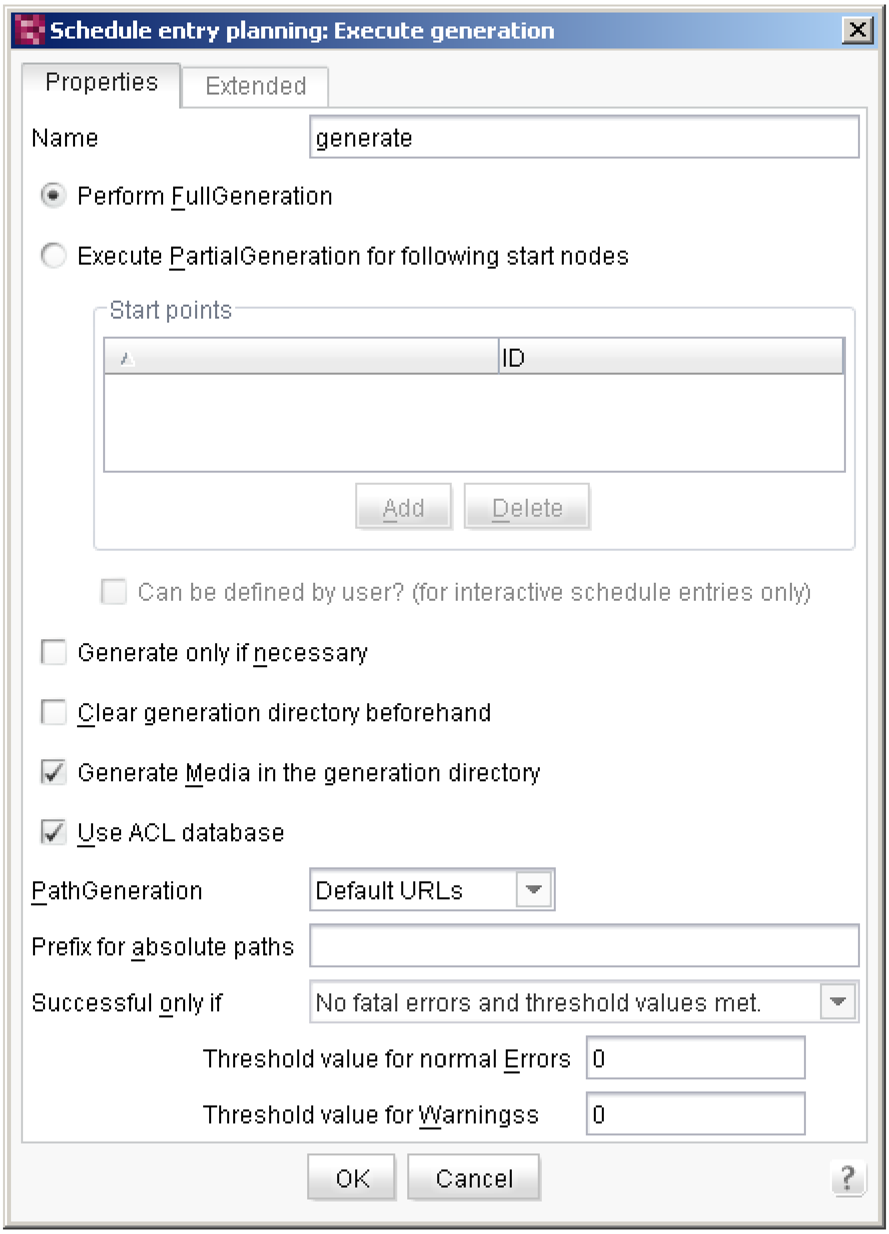
Runtime: As extensive archiving can take a long time to perform, the maximum running time of project archiving can be limited. The "runtime" area can be used to define how much time the archiving may take.

Limit maximum archiving runtime per run to 60 minutes: Use this option to set the limit, i.e. after how may minutes the archiving is to be stopped. The default setting is 60 minutes. The next time the action is started (manually or automatically), the archiving begins at the status reached on stopping. A separate archive file is created for each of these "partial archiving" processes.

Do not limit archiving runtime: If this option is selected the archiving is performed without a time limit until it has finished.

7.5.9.2 Execute generation

This action is used to execute a full or part generation of the selected project (also see section 8.4 page 392). Full generation generates all the project contents whereas part generation only generates the selected “start points” and their children.



Schedule entry planning: Execute generation

Properties | Extended

Name: generate

● Perform FullGeneration

○ Execute PartialGeneration for following start nodes

Start points

	ID

□ Can be defined by user? (for interactive schedule entries only)

□ Generate only if necessary

□ Clear generation directory beforehand

☑ Generate Media in the generation directory

☑ Use ACL database

PathGeneration: Default URLs

Prefix for absolute paths:

Successful only if: No fatal errors and threshold values met.

Threshold value for normal Errors: 0

Threshold value for Warningss: 0

Figure 7-143: Create action – Execute generation

Perform Full Generation: If this option is *activated*, the project is completely generated during schedule entry execution.

Execute Partial Generation for following start nodes: If this option is *activated,* only the nodes displayed in the following “Start points” table are generated (including all subordinate start points).

Starting points: This table shows all the starting points to be generated on performing the action. The selected elements are run through recursively, i.e., for example, if a media folder is added, all subordinate or lower level elements (media and other folders) are also included as part of the generation schedule.

Click this button to open a dialog, which displays all the project's available and not yet selected starting points. Only released objects are shown.

Click this button to delete the selected starting point.

can be defined by user? (for interactive schedule entries only): The "Definable by user" option can be selected for partial generation. By selecting this option, the project administrator can permit the user to define their own starting points for generation on performing the schedule. These changes only affect the current generation.

From FirstSpirit Version 4.1, starting points for the generation of media can also be defined within a generation schedule.

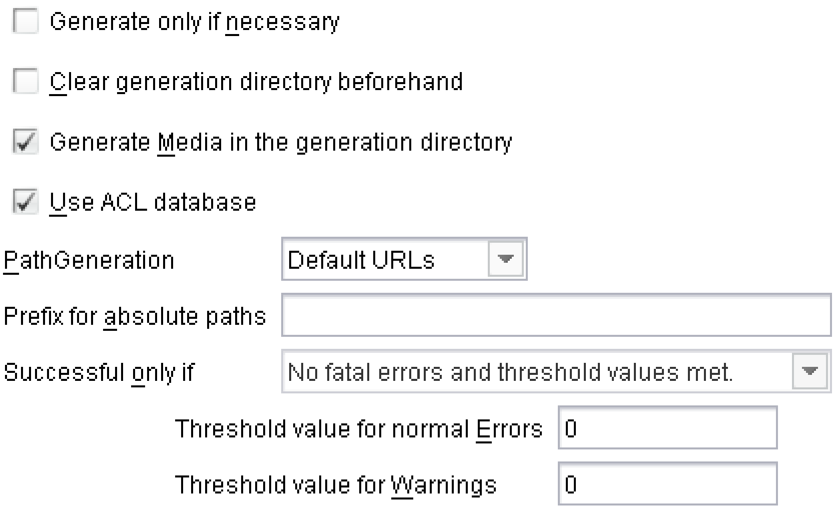
Background: When project content is deployed, referenced media, for example within a picture input component on a page, is also generated and deployed. Media which are not explicitly referenced are not automatically taken into account in the generation. In several application cases, for example when media is used within a script, this media must also be contained in the generation.

Under FirstSpirit Version 4.0, this media could be added to the generation using the "Copy all media in this folder during generation" option. This option was taken into account for all generations. This option is now only available in "Compatibility mode 3.1" (see Chapter 7.4.2 page 257). This option is dropped with Version 4.2 (cf. FirstSpirit Release Notes for Version 4.1 and 4.2).

Media for which the "Parse file" option is selected (see FirstSpirit Manual for Editors), is not generated/deployed if the "Copy all media in this folder during generation" option is selected for it.

From FirstSpirit Version 4.1, the generation of this media can be assigned to individual generation schedules. The objective of the new option is to enable faster deployment of individual sub-areas. Swapping out computing-intensive generation of media into separate generation schedules offers a high performance gain, especially when performing several small sub-generations, as the required media only is taken into account for each generation. This option is also ideally suited, for example, for deploying media in a remote project ("remote media").

Media for which the "Parse file" option is selected (see FirstSpirit Manual for Editors), is not generated/deployed if it is located below a selected starting point and if it is not generated on any page of the generation.



☐ Generate only if necessary

☐ Clear generation directory beforehand

☑ Generate Media in the generation directory

☑ Use ACL database

PathGeneration: Default URLs

Prefix for absolute paths:

Successful only if: No fatal errors and threshold values met.

Threshold value for normal Errors: 0

Threshold value for Warnings: 0

Figure 7-144: Section (see Figure 7-143)

Generate only if necessary: If this option is *activated*, it is checked prior to node generation whether the node has been changed since the last generation. If the node has not been changed, it is not re-generated. If the option is *deactivated*, all the nodes are re-generated even if there have not been changes since the last generation.

Clear generation directory beforehand: If this option is *activated*, the generation directory is emptied prior to generation start.

Generate Media in the generation directory: If this option is selected, media is generated in the generation directory during the generation. If this option is *deselected*, no media is generated during the generation. This setting is useful if deployment is to solely take place via the Site Store and referenced media (on the page) is not to be taken into account, for example, because it is generated in a separate schedule.

Use ACL database: If this option is selected, during generation information - for each page reference in the Site Store and for each medium in the Media Store - is deposited in a local database, the so-called FirstSpirit Access Control database, or ACL database for short. This database is used to provide information about FirstSpirit objects, for example to provide access permissions which have been saved for an object. Synchronisation of the Access Control database with the currently released project status takes place automatically when the content is generated.

For further information on the ACL database, see "Documentation for the FirstSpirit SECURITY module".

Path Generation: Select a path generation method from this list. There are currently three generation types:

- Default URLs: With this method there is an independent subfolder for each project language on the web server (de, en, etc.).
- Multiview URLs: With this method there are no language-specific subfolders. Instead, the files for each language are labelled with the respective language abbreviation. The language abbreviations are inserted *behind* the file extension (e.g. index.html.de, index.html.en) (cf. chap. 7.4.2).
- Infix URLs: There are also no language-specific subfolders for this method. Instead, the files for each language are labelled with the respective language abbreviation. The language abbreviations are inserted *before* the file extension (e.g. index.de.html, index.en.html).

Prefix for absolute paths: The prefix entered here precedes all the links which are allocated with the absolute link property in a template (in the JavaClient).

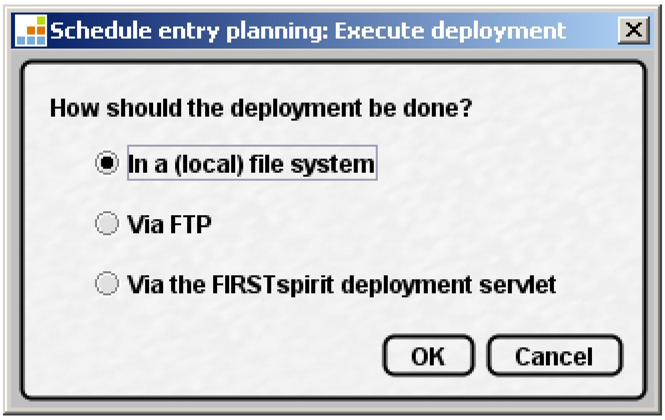
Successful only if: There are two options to define a generation as successful:

- No fatal errors and threshold values met.
- No fatal errors

In both cases the generation is cancelled when a fatal error occurs. If no fatal errors occur, generation is only successful for the options "No fatal errors" and "No fatal errors and threshold values met." if the following threshold values have not been exceeded.

- Threshold value for normal errors
- Threshold value for warnings

7.5.9.3 Execute deployment



Schedule entry planning: Execute deployment

How should the deployment be done?

● In a (local) file system

○ Via FTP

○ Via the FIRSTspirit deployment servlet

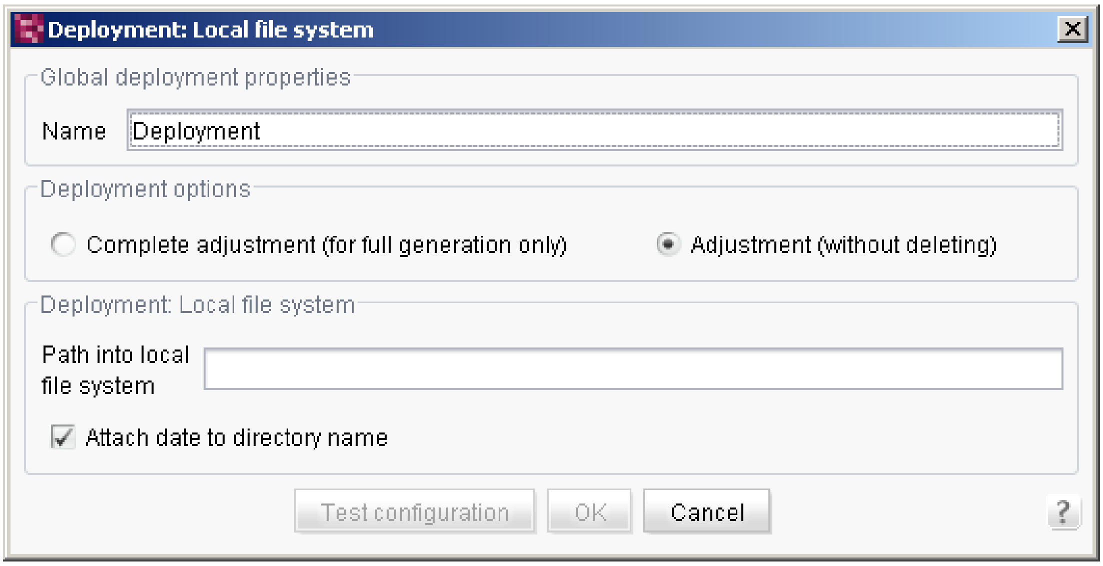
Figure 7-145: Create action – Execute deployment

The deployment type has to be selected before a new deployment can be created. Currently, there are three different possibilities:

- Deploy in a (local) file system (see section 7.5.9.3.1 page 338)
- Deploy via FTP (see section 7.5.9.3.2 page 340)
- Deploy via the FirstSpirit deployment servlet (see section 7.5.9.3.3 page 343)

! *If an existing deployment is to be edited, this dialogue is skipped and the respective type-dependent input mask opens directly.*

7.5.9.3.1 Deploy in a (local) file system



Deployment: Local file system

Global deployment properties

Name: Deployment

Deployment options

○ Complete adjustment (for full generation only) ● Adjustment (without deleting)

Deployment: Local file system

Path into local file system:

☑ Attach date to directory name

Figure 7-146: Create action – Deploy in a (local) file system

This action is used for project deployment in a local file system and, therefore, does not require many settings.

Name: Name of the action which is displayed in the schedule entry overview, the schedule entry management and the action templates.

Deployment options

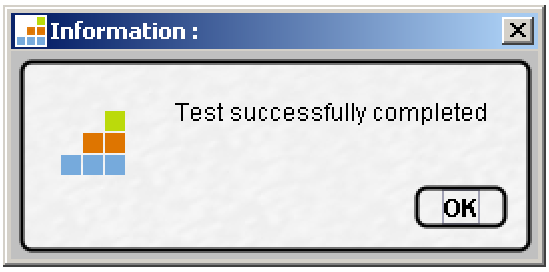
Options – Complete alignment (for full generation only)
This method creates a database on the web server which is completely identical to the generation status. This means: Files which are no longer available on the development server are also deleted on the web server, new files are copied, and old, already existing files are overwritten by the new ones.

Options – Adjustment (without deletion)
This deployment option works in a similar way to the complete alignment; the difference is that files on the web server are not deleted, even if they are no longer available on the development server.

Path into local file system: Specify the local directory for deployment here.

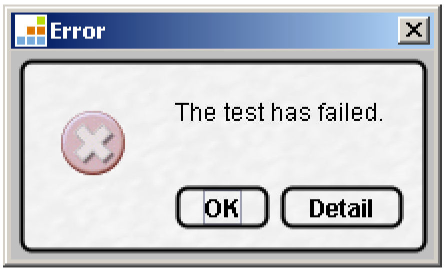
Options – Attach date to directory name: If this option is *activated*, the current date is attached to the paths specified above.

Test configuration Click on this button to test the set configuration. It is checked whether all the required parameters have been specified. The server subsequently tries to create a folder in the specified “Path into the local file system”, to write a file therein, to rename this file and to finally delete this data again. If the test is successful, the following message appears:



Test successfully completed

If the test has failed, the following message appears:

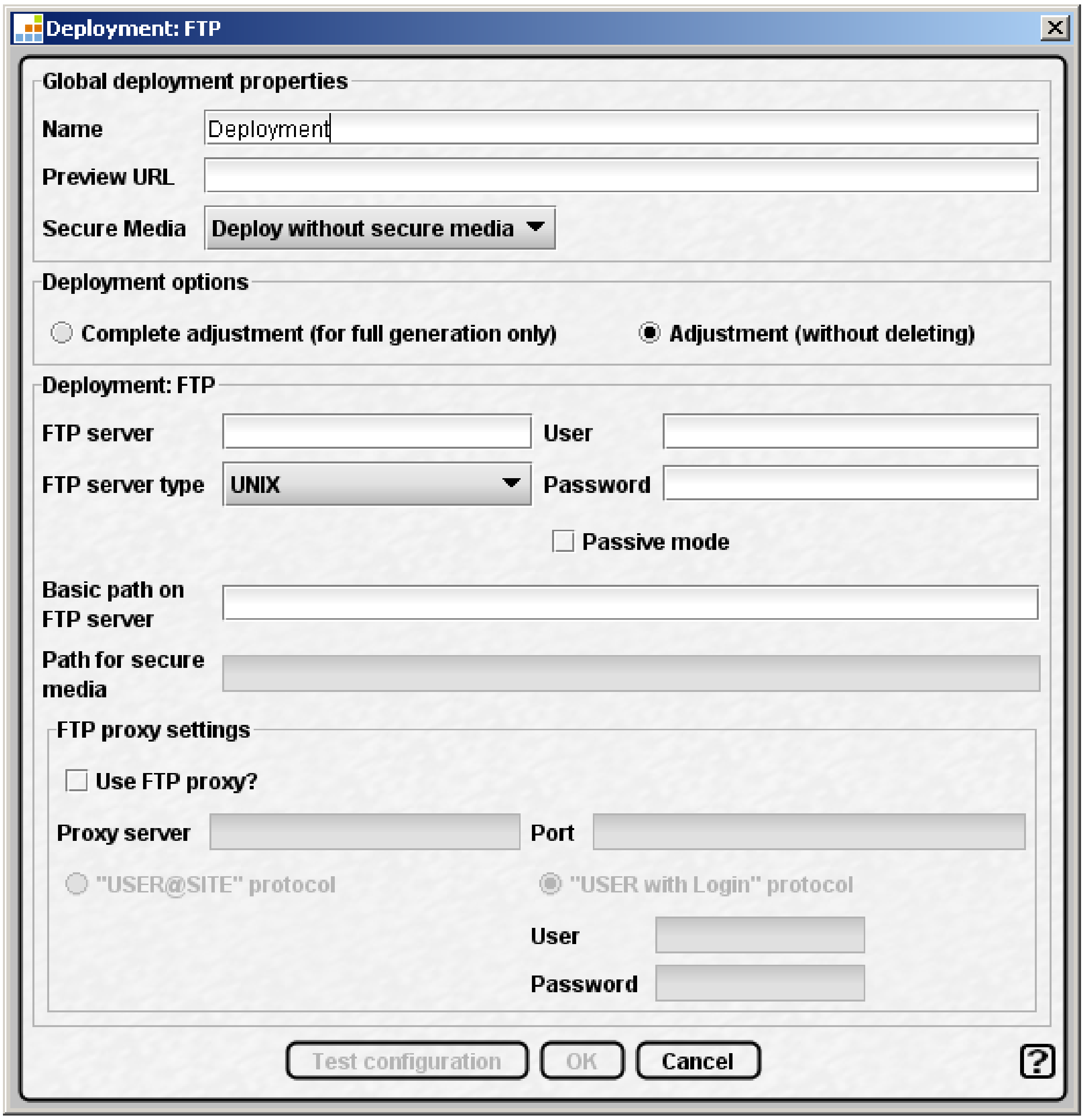


The test has failed.

Click on Detail to open a dialogue with the respective log file which shows the occurred errors.

For more information about the concept “Secure Media“ see “Documentation about Module FirstSpirit SECURITY“.

7.5.9.3.2 Deploy via FTP



Deployment: FTP

Global deployment properties

Name: Deployment

Preview URL:

Secure Media: Deploy without secure media

Deployment options

○ Complete adjustment (for full generation only) ● Adjustment (without deleting)

Deployment: FTP

FTP server: User:

FTP server type: UNIX Password:

□ Passive mode

Basic path on FTP server:

Path for secure media:

FTP proxy settings

□ Use FTP proxy?

Proxy server: Port:

○ "USER@SITE" protocol ● "USER with Login" protocol

User:

Password:

Figure 7-147: Create action – Deploy via FTP

This action is used for project deployment via FTP connection to a remote server and thus requires a number of settings.

Global deployment properties

Name: Name of the action which is displayed in the schedule entry overview, the schedule entry management and the action templates.

Deployment options

Options – Complete adjustment (for full generation only)
With this method a data inventory identical to the generation state is created on the web server. This means: Files which are no longer available on the development server are also deleted on the web server, new files are copied and old, already existing files are overwritten by the new ones.

Options – Adjustment (without deleting)
This deployment option works similar to the complete adjustment, but files are not deleted on the web server, not even if they are no longer available on the development server.

FTP server settings

FTP server: Enter the FTP server address to which the data is to be transferred. This entry is obligatory.

FTP server type: Select the operating system of the FTP server in this combo box.

User: Specify the user for server login at the FTP server.

Password: The password is used in connection with the user name for FTP server login. If a user name is specified, this entry is also obligatory.

Options – Passive mode: If this option is *activated*, connection is established to a port selected by the FTP server. Use this mode if the CMS server is located after a router or if a firewall protects the network against external accesses.

Basic path on FTP server: Specify the path to the FTP server directory to which the generated data is to be transferred.

Path for secure media: Specify the path to the FTP server directory to which the secure media is to be transferred.

FTP proxy settings

Options – Use FTP proxy: If this option is *activated*, connection to the FTP server is established via the proxy server set in this area.

Proxy server: The FTP proxy server address via which the connection is to be established. This entry is obligatory if an FTP proxy is used.

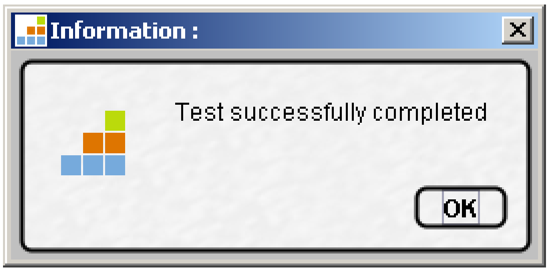
Port: The port to the proxy server specified above. This entry is obligatory if an FTP proxy is used.

Options – “USER@SITE” protocol: If this option is *activated*, proxy authentication takes place via “USER@SITE” protocol, i.e. without separate user ID.

Options – “USER with Login” protocol: If this option is *activated*, proxy authentication takes place via “USER with Login” protocol with the following user ID. In this case, the user name and password have to be specified.

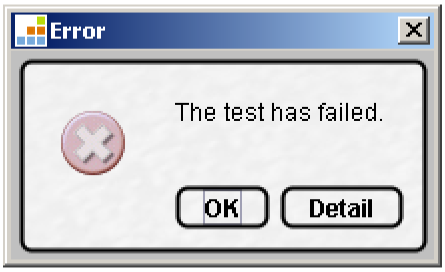
> **User:** The user name used for FTP proxy connection establishment. This entry is obligatory.
>
> **Password:** The password is used in connection with the user name for FTP proxy login and is, therefore, obligatory.

Test configuration Click on this button to test the set configuration. It is checked whether all the required parameters have been specified. The server subsequently tries to establish a connection to the FTP server and to create a folder in the specified “Basic path on FTP server”, to write a file therein, to rename this file and to finally delete this data again. If the test is successful, the following message appears:



Test successfully completed

If the test has failed, the following message appears:

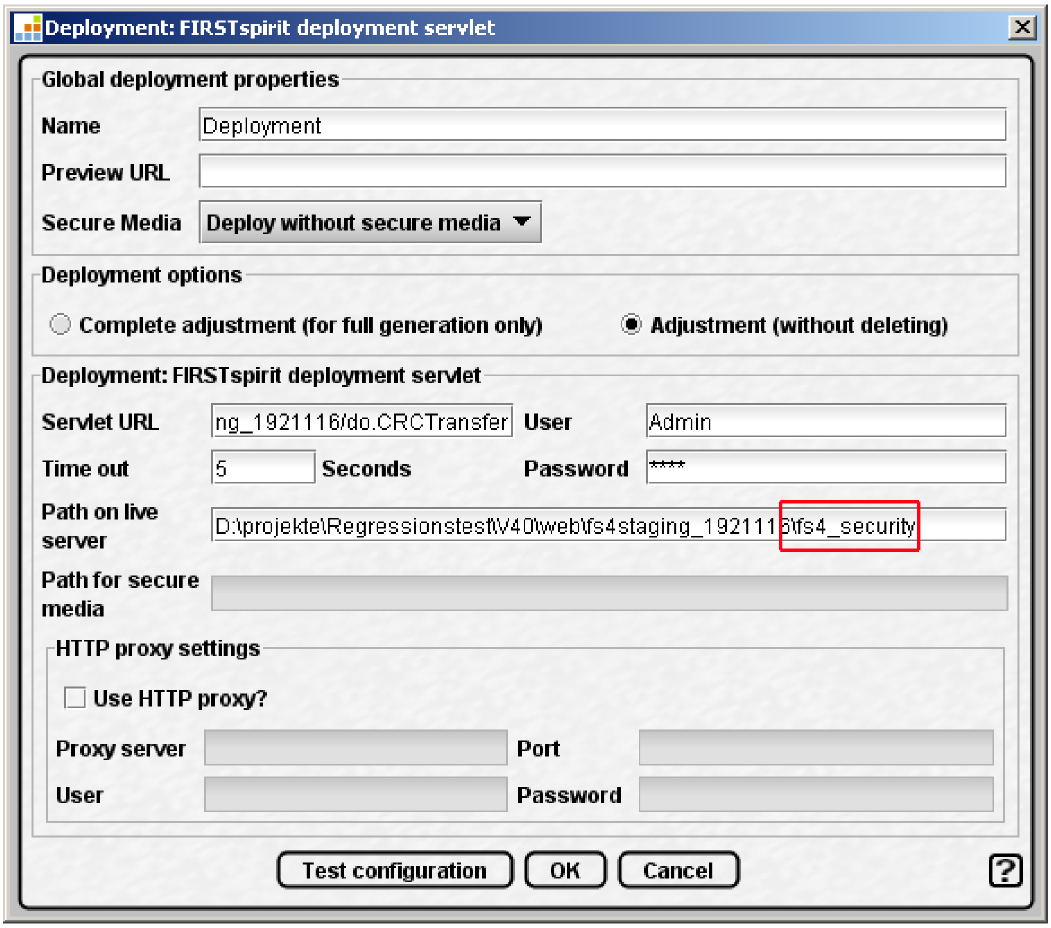


The test has failed.

Click on Detail to open a dialogue with the respective log file which shows the occurred errors.

For more information about the concept “Secure Media” see “Documentation about Module FirstSpirit SECURITY“.

7.5.9.3.3 Deploy via FirstSpirit deployment servlet



Deployment: FIRSTspirit deployment servlet

Global deployment properties

Name	Deployment
Preview URL	
Secure Media	Deploy without secure media

Deployment options

○ Complete adjustment (for full generation only) ● Adjustment (without deleting)

Deployment: FIRSTspirit deployment servlet

Servlet URL	ng_1921116/do.CRCTransfer	User	Admin
Time out	5 Seconds	Password	****
Path on live server	D:\projekte\RegressionstestV40\web\fs4staging_1921116\fs4_security		
Path for secure media			

HTTP proxy settings

□ Use HTTP proxy?

Proxy server		Port	
User		Password	

Figure 7-148: Create action – Deploy via FirstSpirit deployment servlet

This action is used for project deployment via a FirstSpirit deployment servlet. The task of the servlet is to synchronise project files of the FirstSpirit server and the live system. The CRC checksum calculation can be used to determine new, changed or deleted files and only these files are updated. This differential upload accelerates the updating process in the live system. (The information required for this could be read out of an Access Control database[29] which, among other things, manages the CRC checksum of all objects. Using the Access-Control database is not mandatory for the usage of the CRC servlet).

[29] The servlet as well as the Access Control database function is made available via the FirstSpirit SECURITY module and can be adapted to each specific project via the configuration dialog of the corresponding web application "FS Security WebApp".

Global deployment properties

Name: Name of the action which is displayed in the schedule entry overview, the schedule entry management and the action templates.

Deployment options

Options – Complete adjustment (for full generation only)
With this method a data inventory identical to the generation state is created on the web server. This means: Files which are no longer available on the development server are also deleted on the web server, new files are copied and old, already existing files are overwritten by the new ones.

Options – Adjustment (without deleting)
This deployment option works similar to the complete adjustment, but files are not deleted on the web server, not even if they are no longer available on the development server.

Servlet settings

Servlet URL: Enter the complete address to the FirstSpirit deployment servlet here. This entry is obligatory. After configuring the crcTransfer.ini file (via the web application FirstSpirit Security WebApp (FirstSpirit SECURITY module), the servlet mapping is copied into the web.xml file of the application. As a default, the servlet is mapped in *.CRCTransfer.

The servlet can be opened in the web browser to test availability of the servlet, e.g. via:
http://www.mydomain.de/ fs4staging_1921116/do.CRCTransfer
The servlet will display an error message, as no login data has been transferred into the browser.

User: Specify the user for the server login at the servlet. This entry is obligatory.

Password: The password is used in connection with the user name for the servlet login. This entry is obligatory.

Path on live server: Specify the path to the remote server directory to which the generated data is to be transferred. The directory given here can potentially be deleted ("Options – Empty Generation Directory beforehand" – see chapter 7.5.9.2 page 332. The web application directory should therefore never be specified; instead any sub-directory should be given.
To ensure access protection via the FirstSpirit SECURITY module (see chapter 11.3.2 page 456), the prefix for the Access Control database must also be specified too (see

chapter 7.4.16 page 287).

HTTP proxy settings

Options – Use HTTP proxy: If this option is *activated*, the connection to the deployment servlet is established via the proxy server set in this area.

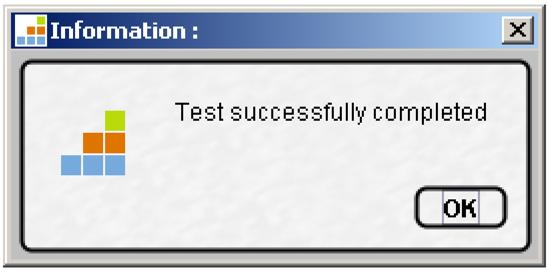
Proxy server: Proxy server address via which the connection is to be established. This entry is obligatory.

Port: The port to the proxy server specified above. This entry is obligatory.

User: The user name used for HTTP proxy connection establishment.

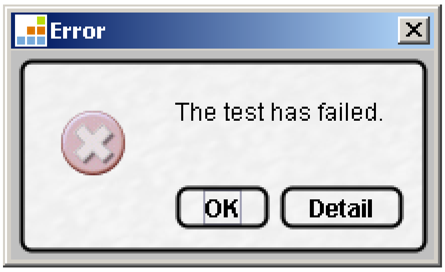
Password: The password is used in connection with the user name for the HTTP proxy login. If a user name has been specified, this entry is obligatory.

Test configuration Click on this button to test the set configuration. It is checked whether all the required parameters have been specified. The server subsequently tries to establish a connection to the deployment servlet and to create a folder in the specified "Path on live server", to write a file therein, to rename this file and to finally delete this data again. If the test is successful, the following message appears:



Test successfully completed

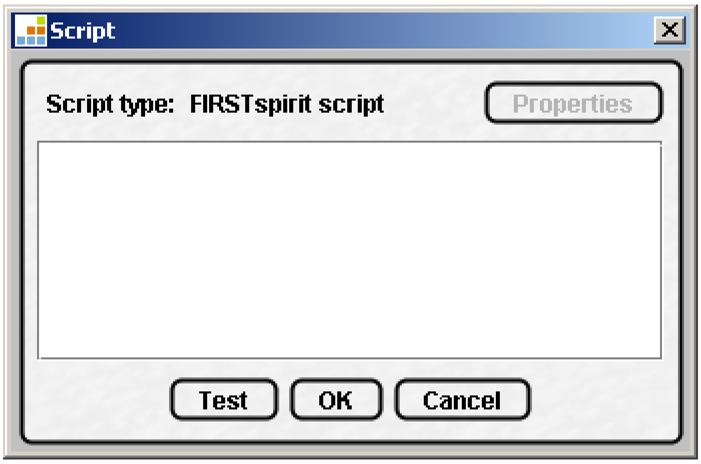
If the test has failed, the following message appears:



The test has failed.

Click on Detail to open a dialogue with the respective log file which shows the occurred errors.

7.5.9.4 Execute script



Script type: FIRSTspirit script

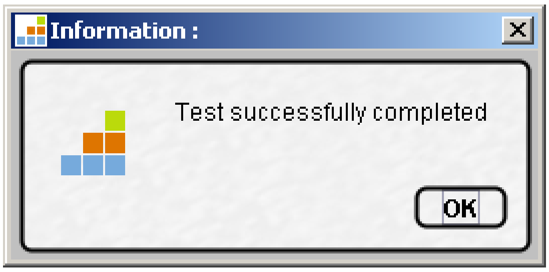
Figure 7-149: Create action – Execute script – Edit script

Name: In this field a name fort he script can be defined.

Text input field: Enter the script code which is to be executed in this action in this field.

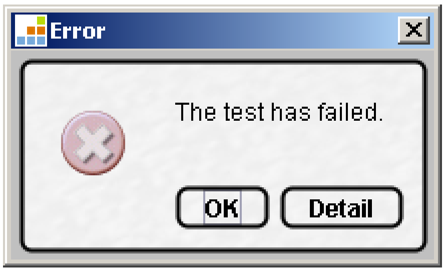
Properties Click on the button to open a new dialogue in which the script properties can be edited (see section 7.5.9.4.1 page 347). This is necessary for example if the script is used to execute changing operations within the project (e.g. creating or changing objects).

Test Click on this button to test the set configuration. It is checked whether all the required parameters have been specified. The server subsequently tries to execute the entered script code with the configured properties. If the test is successful, the following message appears:



Test successfully completed

If the test has failed, the following message appears:



The test has failed.

Click on Detail to open a dialogue with the respective log file which shows the occurred errors.

7.5.9.4.1 Script properties

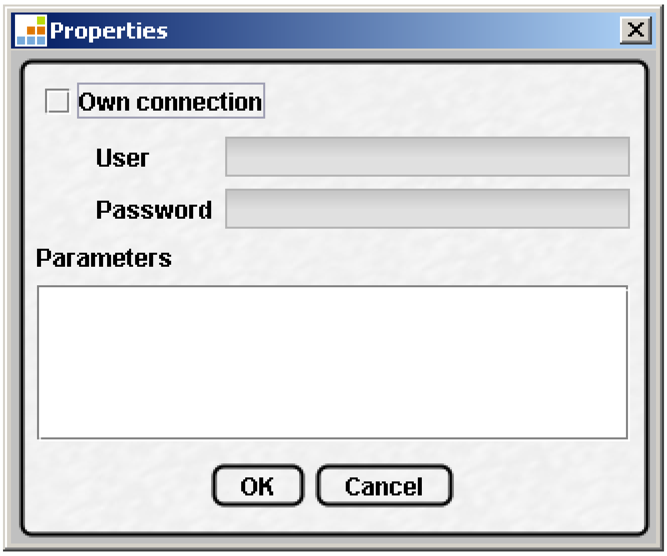


Figure 7-150: Edit script – Script properties

If desired, this dialogue can be used to establish an own connection to the server instead of using the running session for script execution.

Options – Own connection: If this option is *activated*, a new server connection is established based on the following data for script execution.

User: The user name to be used for server login.

Password: The password corresponding to the user specified above.

Parameters: Enter all the parameters to be considered during script execution here.

If the **Own connection** option is selected, the editing privileges/permissions of the user given here can be evaluated and taken into account in the execution of the script.

In older versions of FirstSpirit it was possible to run schedule scripts via the system connection, e.g.:

```
context.getUserService();
```

This could be used to perform modified operations in the project without permission checking. This option generated a "Deprecation" warning in FirstSpirit Version 4.0 and is no longer possible in FirstSpirit Version 4.1.

Scripts which have been configured without special user information can also continue to be run in FirstSpirit Version 4.1. However, this is done using a special connection, which has reading access (via the CAN_SEE, CAN_READ, CAN_META_SEE permissions), but can no longer perform modifying operations. In this case, running of the script is cancelled with a security exception.

The two `context` and `connection` variables are available within a schedule script. From FirstSpirit Version 4.1, the `context` variable provides a special connection (without user information), which now only has reading access (see above). To perform a modifying operation via a schedule script, it is first necessary to define a user via the script properties, who has the relevant permissions (e.g. via the CAN_CHANGE, APPEND, DELETE, etc. permissions).

The `connection` variable must then be used to get a user-specific connection (based on the deposited user information) within the script, e.g.:

```
connection.getProjectById(context.getProject().getId()).getUserService();
```

The schedule script then runs in the permissions context of the given user and evaluates the editorial permissions for each modifying operation (e.g. for create).

> ! *Changing the security model in FirstSpirit Version 4.2:* *If actions modifying the project properties are performed in a script (within a schedule), this script must either be executed in the user context of a server administrator or of a project administrator.*

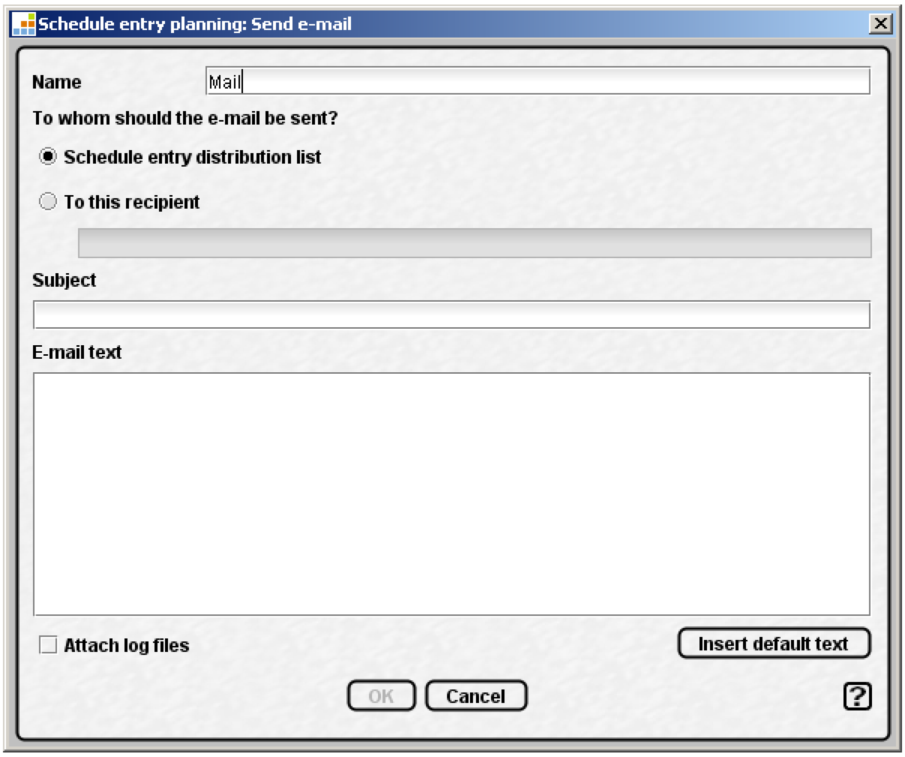
7.5.9.5 Execute project backup

Project backup is used to export a current project status. The project backup is automatically created in the path defined as the backup path in configuration file `fs-server.conf`. (It is also possible to transfer the directory to another hard drive (see section 4.3.1.8 page 46).) Export files which are no longer required for project backup are deleted via the function “Clean up server” (see section 7.2.2.1 page 185).

From FirstSpirit Version 4.2 the licence based module „FirstSpirit Enterprise Backup” allows an efficient method for data backup by means of incremental and differential backups. All changes within a project are saved from a specific starting point (“SnapShot“) only separately. If necessary a complete backup can be created from the initial backup file (“SnapShot” export) and the related differential and / or incremental backup files.

For more information see “FirstSpirit Release Notes Version 4.2” and “Documentation about the module FirstSpirit Enterprise Backup“.

7.5.9.6 Send email


Schedule entry planning: Send e-mail

Name: Mail

To whom should the e-mail be sent?

● Schedule entry distribution list

○ To this recipient

Subject

E-mail text

□ Attach log files

Figure 7-151: Create action – Send e-mail

Emails can be sent via this action. Use the email distribution list of the respective schedule entry or an independent distribution list for this task. The special feature here is that it is possible to attach the log files of previously executed actions in the same schedule entry.

Options – Schedule entry distribution list. If this option is activated, the email is sent to all the recipients defined in the distribution list of the respective schedule entry.

Options – To this recipient: If this option is *activated*, the email is sent to all the email addresses defined in the following text field.

Subject: Subject of the sent emails.

Email text: Enter the email text here. This text can contain template syntax which has been parsed before sending emails. Using the variable `#context` information about the actual schedule (including the actions of this schedule), using the variable `#task` information about the action “Send e-mail” itself can be output[30].
Examples:
`$CMS_VALUE(#context.getStartTime())$`: delivers the starting point in time of the schedule
`$CMS_VALUE(#task.getSubject())$`: delivers the subject of the sent e-mail

Options - Attach log files: If this option is *activated*, log files - if available - of previously executed actions in the same schedule entry are attached to the email.

Click on this button to attach the content of the configuration file for the default template (%ServerDirectory%/conf/server/DefaultMailText.txt) to the email text.

7.5.9.7 Repair references

If a project contains defective references, this action can be used to start recalculation of the references in the project. After the project schedule is started, the following message appears: "The schedule has been successfully performed!". The reference calculation had already started at this time, but does not have to be completed yet. Calculation of the references takes place in the background and, depending on the project size, can take quite a while.

7.5.10 Server-related actions

Server-related actions are created in the server properties and added to the server-related schedule entries.

[30] Methods for `#context` and `#task` can be taken from the FirstSpirit Access-API (*de.espirit.firstspirit.access.schedule.ScheduleContext* or *de.espirit.firstspirit.access.schedule.MailTask*).

The following actions are available:

- Execute script: Analogue to section 7.5.9.4 (see page 346)
- Send email: Analogue to section 7.5.9.6 (see page 349)

7.6 Clustering – load distribution on generation (from V4.1)

This funktionality is released with FirstSpirit Version 4.1.

7.6.1 Introduction (from V4.1)

FirstSpirit supports "vertical scaling", i.e. it is possible to increase the system performance by adding resources, such as more CPUs or by increasing the main memory, as processes such as multi-threading and caching are used to a large extent.

"Horizontal scalability" has now been added with FirstSpirit Version 4.1. It is therefore possible to distribute individual functional components of the FirstSpirit server to different computer systems.

One aspect of this "horizontal scalability" is load distribution to the members of the cluster during generation of the FirstSpirit content. The generation is segmented at schedule level. The generation actions (within one or several schedules) can be distributed to the cluster nodes. A generation action is completely dealt with on a cluster node. Other, parallel pending generation actions can be distributed to other cluster nodes. Scaling of the actions via the cluster nodes takes place automatically. If a valid licence is available for the function the required settings can be defined via the FirstSpirit Server and Project configuration.

Apart from generation of the preview, the generation of content is one of the most time-critical operations in a FirstSpirit environment. Here high computing performance requirements and the wish for the shortest possible waiting time have been addressed together. The aim of the FirstSpirit "Clustering" function is to increase performance in multi-user environments. To this end the computing-intensive generation of the Master server is moved to one (or several) other servers (generation servers).

The following chapters describe the use of the function in the generation of FirstSpirit content.

7.6.2 Concept (from V4.1)

In the following it is assumed that the FirstSpirit server services are to run on different systems and a cluster solution is to be used in the area of the application server.

The architecture shown in the following figure then results:

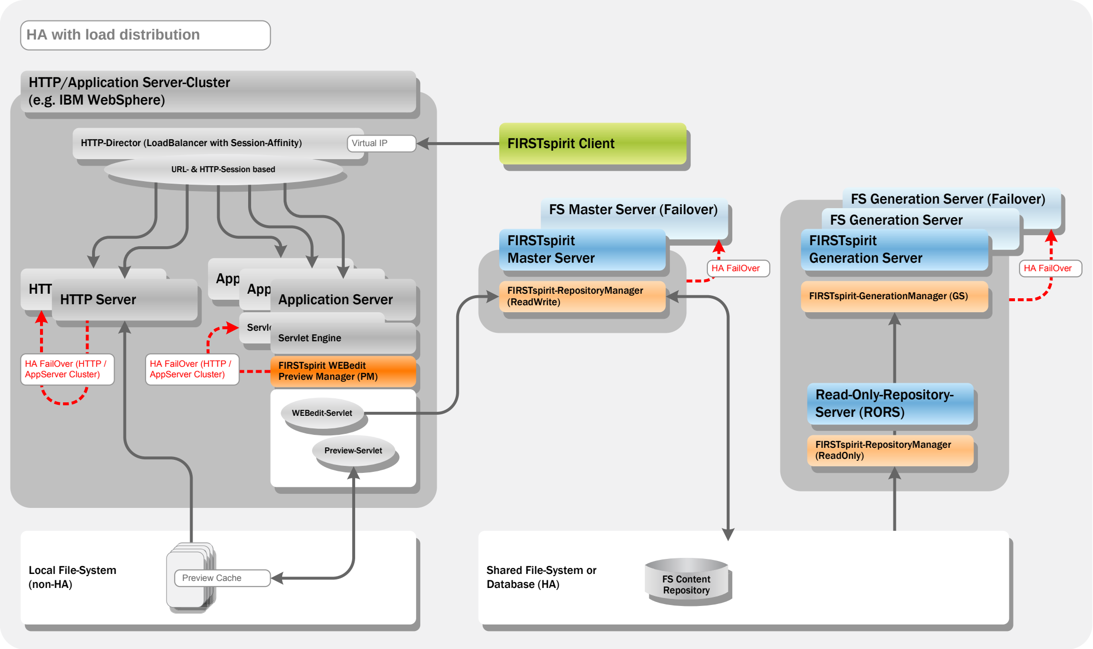

HA with load distribution
HTTP/Application Server-Cluster
(e.g. IBM WebSphere)
HTTP-Director (LoadBalancer with Session-Affinity)
Virtual IP
URL- & HTTP-Session based
HTT
HTTP Server
App
App
Application Server
Servl
Servlet Engine
HA FailOver (HTTP / AppServer Cluster)
HA FailOver (HTTP / AppServer Cluster)
FIRSTspirit WEBedit Preview Manager (PM)
WEBedit-Servlet
Preview-Servlet
FIRSTspirit Client
FS Master Server (Failover)
FIRSTspirit Master Server
HA FailOver
FIRSTspirit-RepositoryManager (ReadWrite)
FS Generation Server (Failover)
FS Generation Server
FIRSTspirit Generation Server
HA FailOver
FIRSTspirit-GenerationManager (GS)
Read-Only-Repository-Server (RORS)
FIRSTspirit-RepositoryManager (ReadOnly)
Local File-System
(non-HA)
Preview Cache
Shared File-System or Database (HA)
FS Content Repository

Figure 7-152: High-availability cluster with load distribution

The FirstSpirit Client (both the JavaClient and the WebClient) communicates with the clustered web application server via HTTP(S). The FirstSpirit sessions are distributed to the cluster's individual application servers via the HTTP(S) Load Balancer. Load distribution can take place on URL basis and/or on HTTP Session basis.

Behind the application server cluster, a range of FirstSpirit services are started on different systems:

FirstSpirit Master server: The FirstSpirit Master server centrally manages all FirstSpirit projects and deals with the users' queries/changes and distributes the tasks, wherever possible, to other FirstSpirit servers.

FirstSpirit Generation servers:
Several FirstSpirit generation servers should be used for complex or frequent deployment processes to move the load from the master when generating the web presence. If necessary, several deployments can also be distributed to different servers in this way. A generation server contains an RORS.

ReadOnlyRepository server (RORS): A special Repository Manager processes the queries from a Generation server.

The individual FirstSpirit servers decide at the start which schedule area they are responsible for and which managers have to be started for this. A FirstSpirit generation service is created, for example, from a "normal" FirstSpirit server, which only starts the servers and generation manager required for communication. The FirstSpirit software architecture is set up so that (in simplified terms) a generation manager is used when a schedule is executed whereby it is not possible to tell whether it runs locally or remotely.

7.6.3 Check licence file (from V4.1)

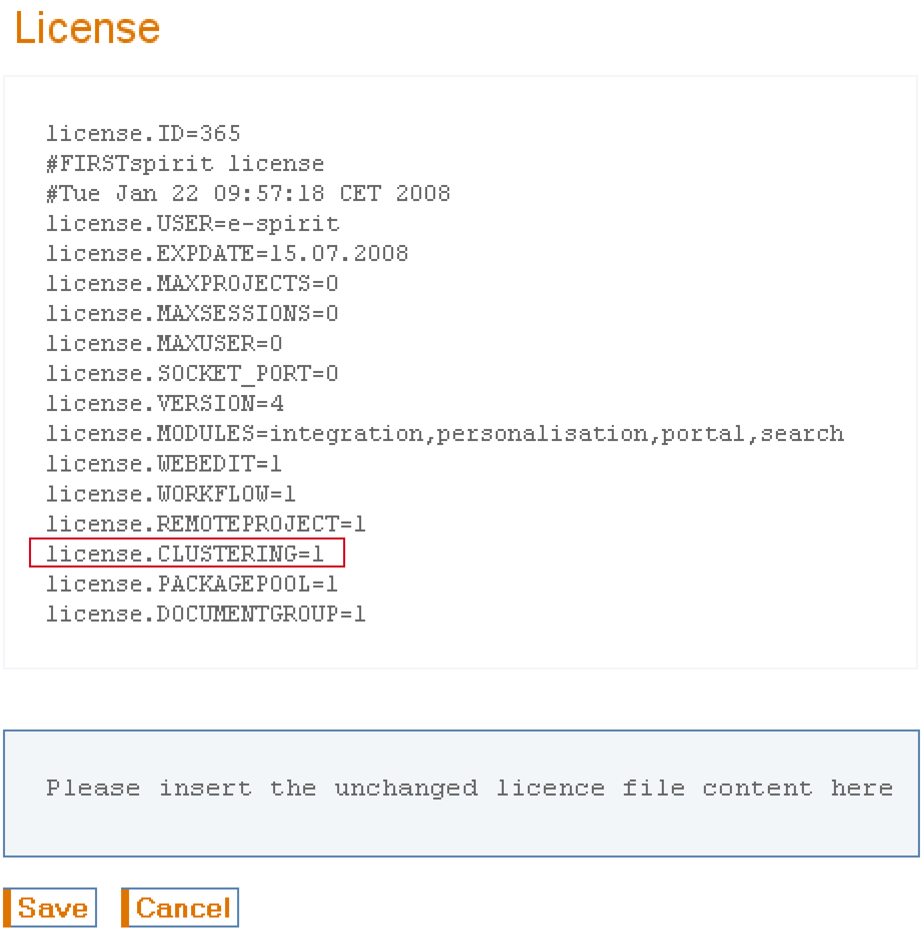
The "Clustering" function is a licence-dependent function. The "Clustering" menu item is only displayed in the FirstSpirit Server and Project configuration application if a valid licence exists for this function.

Use the "FirstSpirit – Configuration – Licence" menu of the FirstSpirit Server Monitoring to display the valid FirstSpirit functions of the `fs-license.conf` licence file (see chapter 8.6.1.2 page 396). The parameter `license.CLUSTERING` parameter must be set to the value `1` for use of the function (see Figure 7-153).

If this is not the case, a new valid licence can be requested and inserted in the blue part of the window. Click the Save button to save the new licence file.

! *Manipulations of the `fs-license.conf` result in an invalid licence. If such changes are necessary, please contact the manufacturer.*

! *When inserting a new configuration file `fs-license.conf` with changed Clustering setting it is necessary to restart the server. The Cluster Manager is started automatically.*



License

```
license.ID=365
#FIRSTspirit license
#Tue Jan 22 09:57:18 CET 2008
license.USER=e-spirit
license.EXPDATE=15.07.2008
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
```

Please insert the unchanged licence file content here

Figure 7-153: Display of the licence file parameters (Server Monitoring)

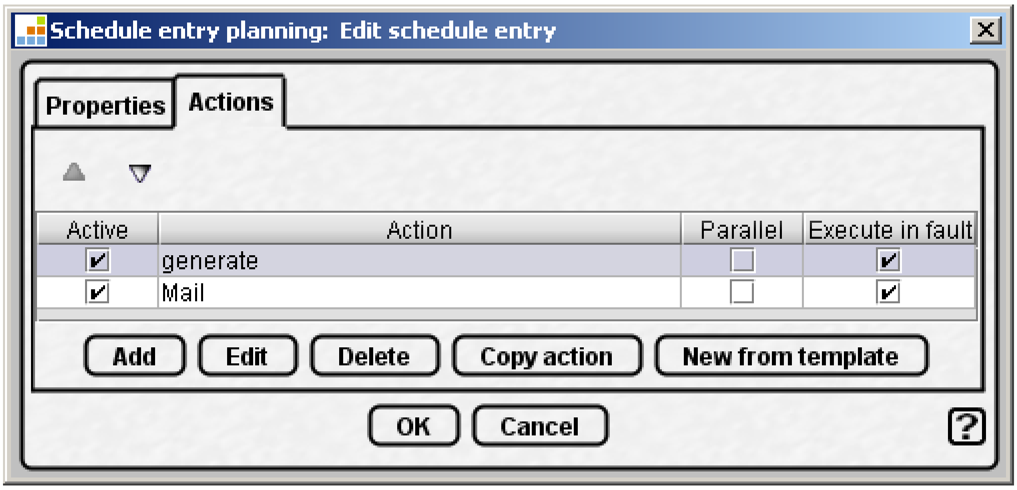
7.6.4 Configuration of the cluster nodes (from V4.1)

The cluster nodes are configured via the FirstSpirit Server and Project configuration in the "Server Properties" area of the “Clustering” menu (see chapter 7.3.17 page 248).

7.6.5 Configuration of the generation schedule (from V4.1)

A generation schedule is created and configured via the Server and Project configuration application. All schedules created for the project are displayed in the “Project Properties” area under the menu item “Schedule Management”. New schedules for the generation (or partial generation) of a project can also be created here or already existing schedules can be processed. (For further information on creating a generation schedule see chapter 7.5.2 and chapter 7.5.9).

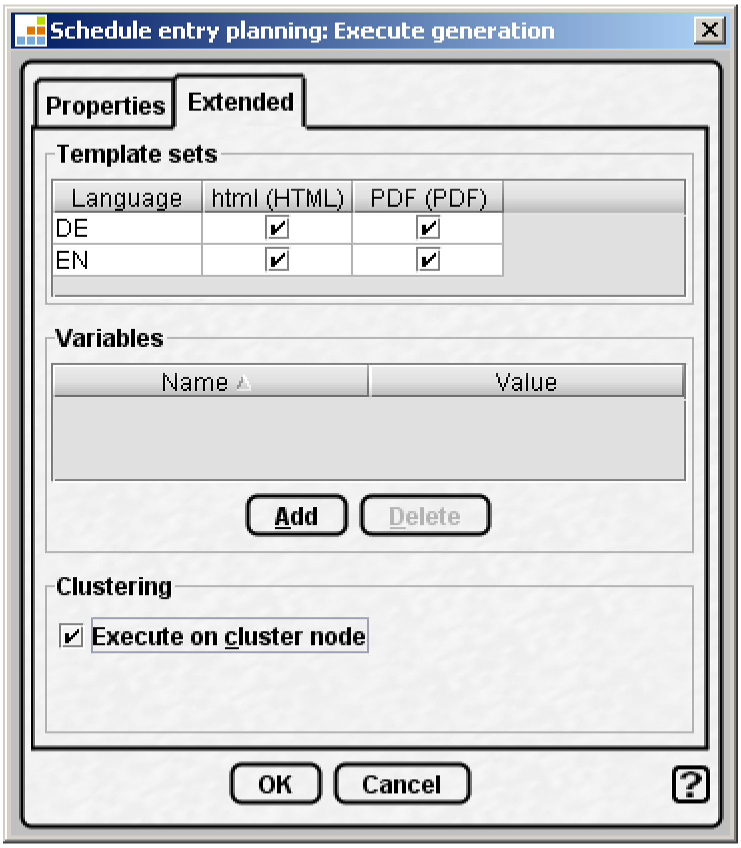
The generation schedule contains the action "generate" (in the Actions tab). This action is used to carry out full or partial generation of the selected project. Full generation generates all content of the project, partial generation the "starting points" and their children only.



Active	Action	Parallel	Execute in fault
☑	generate	☐	☑
☑	Mail	☐	☑

Figure 7-154: Project-related "generate" action of a generation schedule

Edit The configuration setting for the action can be opened by clicking the button. The following dialog opens:



Template sets

Language	html (HTML)	PDF (PDF)
DE	☑	☑
EN	☑	☑

Variables

Name	Value

Clustering

☑ Execute on cluster node

Figure 7-155: Advanced settings for the "generate" action

Apart from the known generation settings (see chapter 7.5.9), the "Extended" tab also contains the “Clustering" area. The "Execute on cluster node" checkbox can be activated here. With activation of this checkbox, generation of the contents (full or partial generation) is distributed among the existing cluster nodes (or generation servers). The cluster node with the lowest capacity utilisation at this time is always used.

An overview of the existing cluster nodes can be opened via FirstSpirit Server Monitoring (see chapter 8.6.6 page 415).

7.7 Configuration of the spelling check

The "SpellService" module is used for spelling checking in the FirstSpirit JavaClient and in the WebClient.

Spelling checking is available in the following input components:

- CMS_INPUT_DOM
- CMS_INPUT_DOMTABLE

The "SpellService" module consists of:

- a local project component: This component can be added to the required projects via their project properties after the module has been installed on the server (see chapter 7.4.17 page 289). It is then possible to configure this component globally (see 7.7.3 page 359) and project-specifically (see chapter 7.7.7 page 365).
- a service: The service is a server component which can be addressed from input components (or scripts) via a public interface.

7.7.1 Install/uninstall SpellService (server properties)

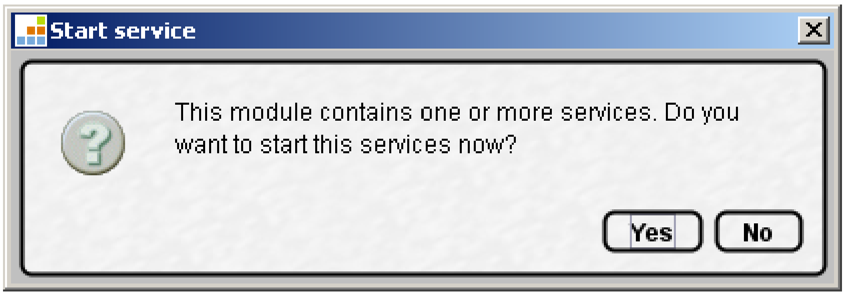
7.7.1.1 Install

The "SpellService" module is made available as an fsm file and must first be installed on the server via Server and Project Configuration.

If the module has already been installed on the server and is only to be replaced by a newer version, it can be updated (see chapter 7.7.2). In this case installation is not necessary.

The installation takes place via the server properties in the "Modules" area (see Chapter 7.3.14 Seite 232).

Click the button to open a dialog to select the fsm file from the local file system. As the "SpellService" module contains a service, the "Start Service" dialog opens first“:

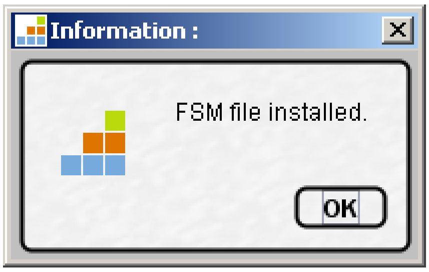

Start service

This module contains one or more services. Do you want to start this services now?

Figure 7-156: Start services automatically

If the dialog is confirmed with "Yes" the "SpellService" service is started automatically. But the service can also be started at a later time, either via Server and Project Configuration or via Server Monitoring (see chapter 7.7.5 page 363).

Successful installation of the fsm file is confirmed by a message. The "SpellService" module is then listed in the "Modules" area.



FSM file installed.

Figure 7-157: SpellService installed

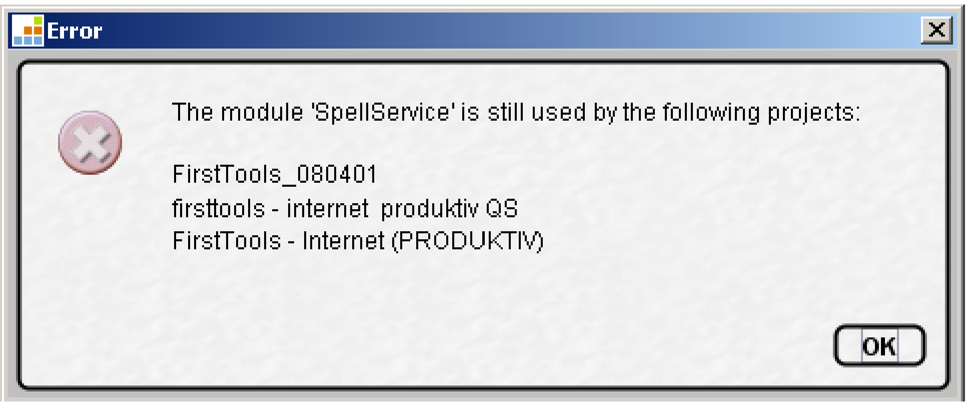
Following the installation the local project component can be configured (see chapter 7.7.6 page 364).

7.7.1.2 Uninstall

The SpellService is uninstalled via the server properties in the "Modules" area (see chapter 7.3.14 page 232).

If the module has already been installed on the server and is only to be replaced by a newer version, it can be updated (see chapter 7.7.2). In this case it is not necessary to uninstall.

The following error message appears if the module is still being used in projects:



The module 'SpellService' is still used by the following projects:

FirstTools_080401
firsttools - internet produktiv QS
FirstTools - Internet (PRODUKTIV)

Figure 7-158: Uninstall SpellService – Error message

In this case the project components within the project properties have to be deleted first (see chapter 7.7.6 page 364). The module can only be successfully uninstalled if no more uses exist.

Uninstall Click the button to uninstall the module.

The JAR File was successfully uninstalled.

Figure 7-159: Uninstall SpellService

7.7.2 Update SpellService (server properties)

The module can always be updated if a more recent version of the fsm file exists. (An error message appears if the file is older or identical with the installed version.)

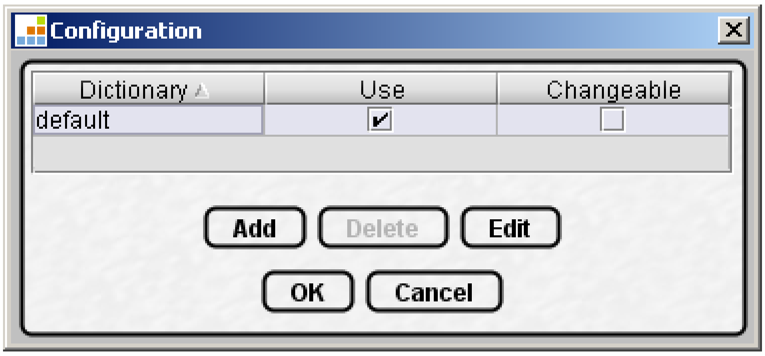
Update Click the button to open a dialog to select the fsm file from the local file system. Following successful installation of the new version the version can also be updated within the project properties.

The configuration settings to date are retained with the update.

7.7.3 Configure Global SpellService

Global configuration of the service is possible via the server properties. Dictionaries can be added to the SpellService in this area. Dictionaries have a unique name and have different language contents.

Configure Click the button to open the "Configuration" dialog. The "default" dictionary is displayed as a default. The dictionary is used server-wide for all projects in which the SpellService is configured and contains the languages German, English, French and Spanish.



Dictionary	Use	Changeable
default	☑	□

Figure 7-160: "SpellService" global configuration settings

The dialog can be used to change the global configuration of the "SpellService". The following information is managed for each dictionary:

Name: server-wide unique name of the dictionary.

Use: If this option is activated the dictionary can be managed centrally from the projects. If use of a dictionary is deactivated the "Changeable" right is also automatically deactivated.

> ! *Global dictionaries are used in each new project created, even if SpellService configuration is NOT available in the project. If use of the dictionary is to be explicitly prevented, the project configuration "SpellService Configuration" must be added first, then the selected global dictionaries or the whole SpellService are explicitly deactivated.*

Changeable: If this option is activated the dictionary can be changed centrally from a project. This means, new entries can be added to the dictionary. Globally defined read/write rights can be changed via the project configuration (into read rights only). The reverse case is of course not possible. The "Changeable" checkbox can only be edited if "Use" is activated.

Add Click the button to configure other global dictionaries. Here a unique name must be assigned for the new dictionary first. The global dictionary is used under this name server-wide for all projects in which SpellService is configured. The globally defined configuration (use/changeable) can also be adjusted in the individual project configurations.
Globally added dictionaries are filed in the server directory `conf\modules\SpellService.SpellService`.

Delete Click the button to remove a previously added dictionary. A confirmation prompt appears before the dictionary is deleted. The "default" dictionary cannot be deleted, but it is possible to deactivate its use.

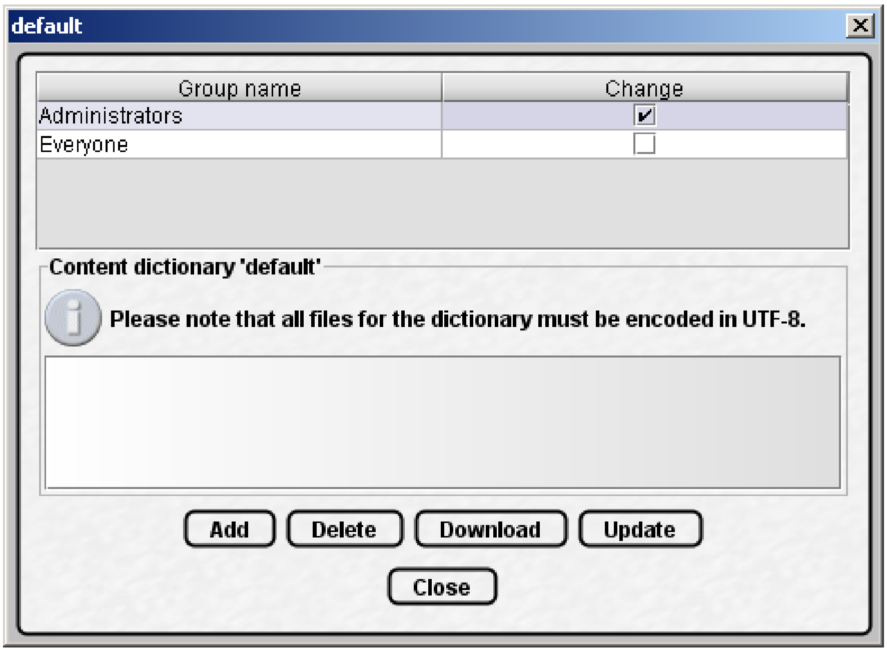
Edit Click the button to edit the configuration for the dictionary. The configuration dialog opens (see chapter 7.7.4 page 361).

OK The changes are saved by clicking the button.

Cancel The changes are not saved if this button is clicked; the window is closed.

7.7.4 Configure global dictionaries

Edit Click the button to open the dialog for configuring the dictionary.



default

Group name	Change
Administrators	☑
Everyone	☐

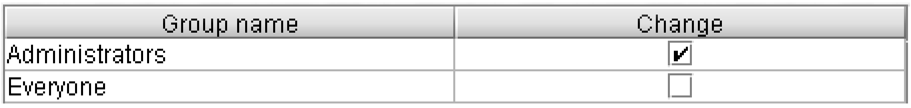
Content dictionary 'default'

Please note that all files for the dictionary must be encoded in UTF-8.

Figure 7-161: Configure dictionary

The dialog is divided into an area for rights definition (see chapter 7.7.4.1 page 361) and an area for optional addition of language-dependent contents to the dictionary (Dict files) (see chapter 7.7.4.2 page 362).

7.7.4.1 Configure permissions for global dictionaries



Group name	Change
Administrators	☑
Everyone	☐

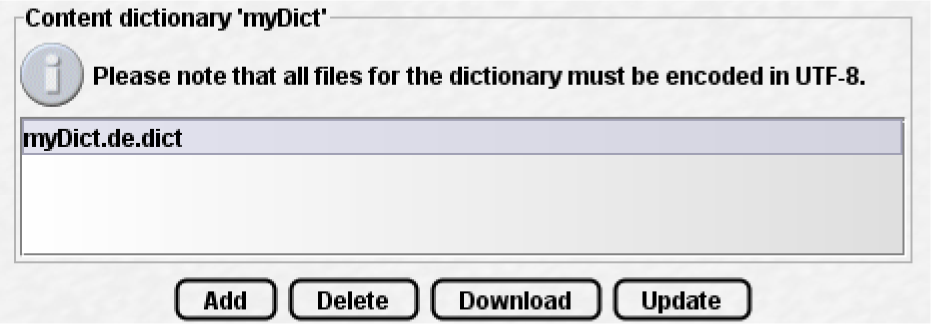
Figure 7-162: Configure permissions

The global write rights for the "Administrators" and "Everyone" groups can be activated or deactivated in the top part of the configuration dialog (cf. Figure 7-161). Other project groups are displayed in this area within the project configuration (see chapter 7.7.8 page 367).

The global rights defined here affect read-write rights within the project configuration; however, can be changed there (see chapter 7.7.8.1 page 367).

! *Exporting projects:* The global dictionaries are managed centrally by SpellService. This means the global dictionaries are NOT exported when a project is exported. The project export takes into account local project dictionaries only.

7.7.4.2 Add language-dependent contents to global dictionaries



Content dictionary 'myDict'

Please note that all files for the dictionary must be encoded in UTF-8.

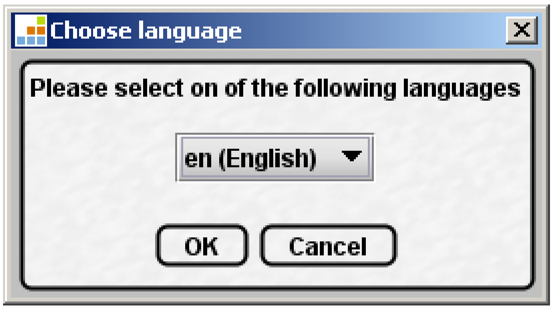
myDict.de.dict

Figure 7-163: Add contents

Language-dependent contents can be added to the dictionary in the bottom part of the configuration dialog (cf. Figure 7-161). Addition of these files is optional. Either word lists already exist for the languages (content languages) which can be added in this area or no files are given. In this case, new lists are automatically created the first time a new word is added in JavaClient or WebClient; these are then subsequently displayed in the configuration dialog too.

For example, if an editor adds an unknown word to the dictionary for English (content language) the corresponding Dict file is automatically created on the server.

Add Click the button to add a new word list (Dict file) to the dictionary. A dialog for selecting the Dict file from the local file system opens. (The button is only active as long as the word has not yet been assigned for all languages). Following the selection, assignment of the file to the required content language (in the projects) must be defined:



Choose language

Please select on of the following languages

en (English)

Figure 7-164: Select target language

The drop-down list displays all languages defined in the "Language Templates" area within the server properties (see chapter 7.3.7 page 224). Each Dict file is therefore assigned to a specific server language. Languages for which assignment already exists are displayed grey in the list.

After assigning to a language the word list is displayed in the dialog with the following name: `Name_of the_Dictionary.LanguageAbbreviation.dict`

Delete Click the button to remove a previously added word list (Dict file). A confirmation prompt appears before the dictionary is deleted. The "default" dictionary cannot be removed, but it is possible to deactivate its use.

Download Click the button to export an existing Dict file. A dialog for selecting a download directory from the local file system opens. The Dict file can be exported into the selected directory under its existing name or under a new name. The "Update" function can be used to store the Dict file locally, edit it and then add it back into the project configuration with the "Update" button.

Update Click the button to replace an already added Dict file with a new file. Unlike "Add" a Dict file to a dictionary, here it is not necessary to assign the file to the required content language. The existing language assignment and the unique name from the original file are adopted and only the file on which it is based is replaced (for example the file written in a local directory for editing by means of a "Download".

7.7.5 Start and configure "SpellService" service

Different options exist for controlling and configuring the "SpellService" service:

- Via the Server and Project configuration within the server properties in the "Modules" area (see chapter 7.3.14 page 232)
- Via the Server Monitoring in the FirstSpirit – Configuration – Services area (see

chapter 8.6.1.7 page 400) and in the FirstSpirit – Configuration – Control area (see chapter 8.6.2.4 page 402).

7.7.6 Add SpellService as project component

Spelling checking can be activated or deactivated for a specific project. In this case the Spell Service module is first installed centrally via the server properties (see chapter 7.7.1 page 357). The global configuration is also carried out via the server properties (see chapter 7.7.3 page 359). Global dictionaries can be created and configured which are then available within the Spell Service project configuration (see chapter 7.7.4 page 361).

Project-specific configuration of the SpellService then takes place via the project properties in the "Project Components" area (see chapter 7.4.17 page 289).

! *The SpellService must be started before a Spell Service project configuration can take place (see Chapter 7.7.5 page 363).*

Add Click the button to add the Spell Service configuration to a project.

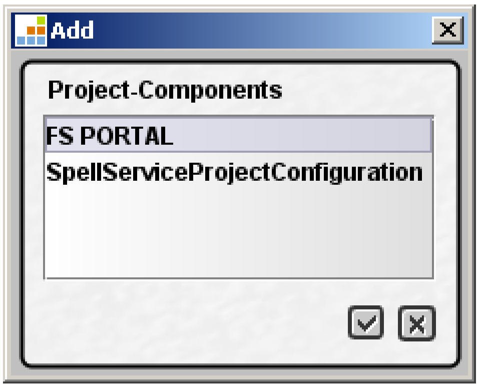
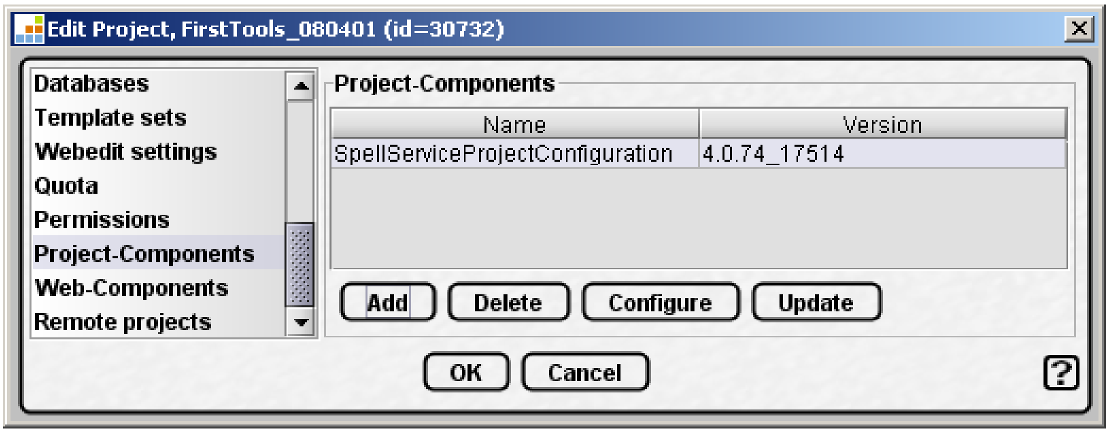


Figure 7-165: Server properties

The globally configured functions of the installed component are then available in the project.



Edit Project, FirstTools_080401 (id=30732)

Databases
Template sets
Webedit settings
Quota
Permissions
Project-Components
Web-Components
Remote projects

Project-Components

Name	Version
SpellServiceProjectConfiguration	4.0.74_17514

Figure 7-166: Project properties – Spell Service Project Configuration

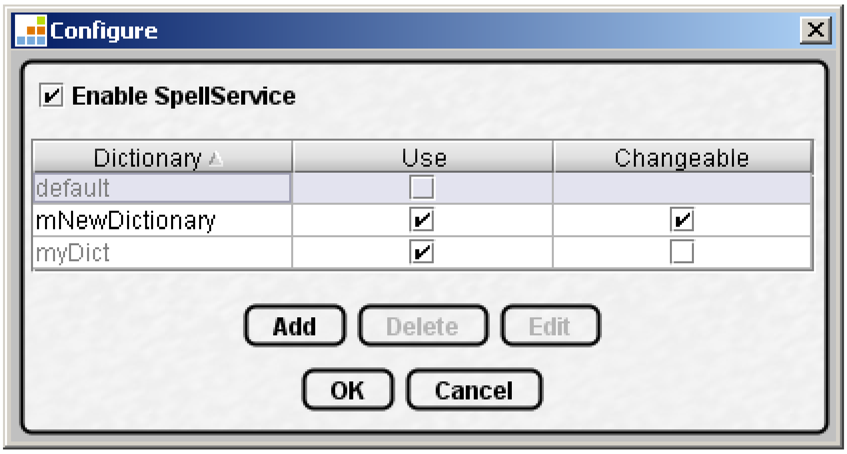
Delete Click the button to remove a previously added Spell Service Project Configuration. All project-specific Spell Service configurations (project-specific dictionaries, permissions) are also removed. A confirmation prompt appears before the dictionary is deleted.

Update Click the button to update an existing Spell Service Configuration. The button is only active if the "SpellService" module has been updated on the server (see chapter 7.7.2 page 359). In this case the version numbers of the global Spell Service project configuration and the project-specific Spell Service project configuration differ. The configuration settings to date (dictionairies, permissions) are retained with the update. (Depending on the new module version however, manual adjustments may be necessary.)

Configure Click the button to define the project-specific Spell Service configuration (see chapter 7.7.7 page 365).

7.7.7 Project-specific Spell Service configuration

Initially only the globally defined dictionaries are displayed in the project-specific configuration dialog (grey coloured in the overview). Depending on the configuration, these global dictionaries can be directly used and (possibly) changed in the project. Within the configuration, other project-specific dictionaries can also be added and edited.



☑ Enable SpellService

Dictionary	Use	Changeable
default	□	
mNewDictionary	☑	☑
myDict	☑	□

Figure 7-167: Configuration of the Spell Service project configuration

The dialog can be used to change the project-specific configuration of the "SpellService". The following information is managed for each dictionary:

Name: Project-wide unique name of the dictionary (for global dictionaries the name is unique server-wide).

Use: If this option is activated the dictionary can be used within the project. If use of a dictionary is deactivated the "Changeable" right is also automatically deactivated.

> ! *Global dictionaries are used in each new project created, even if SpellService configuration is NOT available in the project. If use of the dictionary is to be explicitly prevented, the project configuration "SpellService Configuration" must be added first and then the selected global dictionaries must be explicitly deactivated.*

Changeable: If this option is activated the dictionary can be changed from the project. This means, new entries can be added to the dictionary. Globally defined read/write rights can be changed via the project configuration (into read rights only). The reverse case is of course not possible. The "Changeable" checkbox can only be edited if "Use" is activated.

☑ Enable SpellService If the checkbox is deactivated, spelling checking is deactivated for the project. Neither the contents of the global dictionaries nor the contents of the project-specific dictionaries are evaluated. As a default, the spelling checking tool is activated for all projects.

Add Click this button to add other project-specific dictionaries (see chapter 7.7.8 page 367). Here a project-wide unique name must be assigned for the new dictionary first. The contents of project-specifically added dictionaries are created in the server directory `\data\projects\project_projectID\modules\SpellService.SpellServiceProjectConfiguration` .

Delete A previously added project-specific dictionary can be removed with a click. A confirmation prompt appears before the dictionary is deleted. Global dictionaries cannot be deleted via project configuration, but their use can be deactivated within the project.

Edit Click the button to edit the configuration for the dictionary. The dictionary configuration dialog opens (see Chapter 7.7.8 page 367).

OK The changes are saved by clicking the button.

Cancel The changes are not saved if this button is clicked; the window is closed.

! *Importing projects: When a project with a Spell Service configuration is imported the system checks whether SpellService has been installed and started. If not, the administrator receives a warning but the local configuration of the SpellService is created. If the SpellService is subsequently activated, the configuration and local dictionaries are immediately available.*

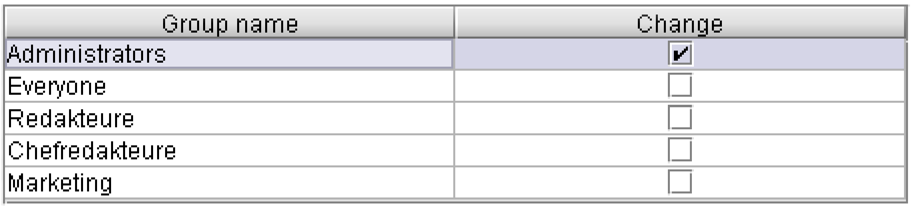
7.7.8 Add project-specific dictionaries

Project-specific dictionaries are configured analogous to the configuration of global dictionaries (see Chapter 7.7.4 page 361).

Edit Click the button to open the dialog for configuring the dictionary (see Figure 7-161).

The dialog is divided into an area for rights definition (see chapter 7.7.8.1 page 367) and an area for optional addition of language-dependent contents to the dictionary (Dict files) (see chapter 7.7.4.2 page 362).

7.7.8.1 Configure permissions for project-specific dictionaries



Group name	Change
Administrators	☑
Everyone	☐
Redakteure	☐
Chefredakteure	☐
Marketing	☐

Figure 7-168: Configure permissions

The project-specific write rights for a project group can be activated or deactivated in the top part of the configuration dialog (cf. Figure 7-161).

Globally set rights are copied into the project configuration as the standard configuration, but can be changed there.

Unlike global configuration (see Chapter 7.7.4.1 Seite 361) ot only the standard groups "Everyone" and "Administrators" are available here but all groups which have access to the project (see Chapter 7.4.8 Seite 273).

7.7.8.2 Add language-dependent contents to global dictionaries

Language-dependent contents can be added to the dictionary in the bottom part of the configuration dialog (cf. Figure 7-161). Addition of these files is optional. Either word lists already exist for the languages (content languages) or new empty lists can be created.

The addition of language-dependent contents to project-specific dictionaries is carried out analogous to the addition of contents to global dictionaries (see Chapter 7.7.4.2 page 362).

The dialog for assignment of a Dict file to a project language (content language) offers the languages known in the project only (see chapter 7.4.5 page 265).

7.8 Support for Apache FOP (from V4.1)

With the release of FirstSpirit Version 4.1, support for Apache FOP was swapped out into a module. The module can be installed, as usual, via the server and project configuration (see Chapter 7.3.14 page 232).

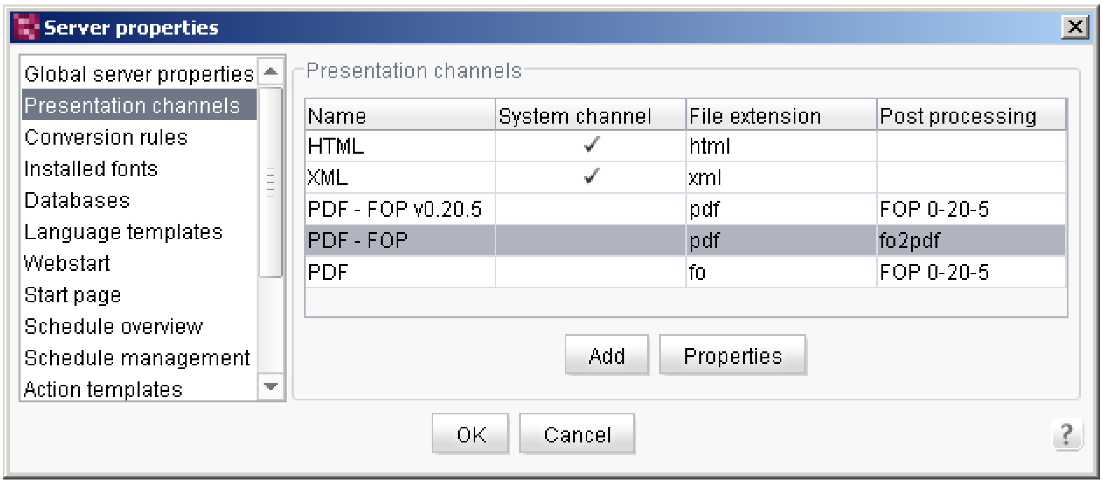
Two modules are available for installation with the release of **FirstSpirit Version 4.1**:

- **fs-fop.fsm:** new module, which includes the current status of Apache FOP. At the time of releasing FirstSpirit Version 4.1, the "fs-fop.fsm" module has Apache

FOP status 0.95. As extensive practical experience with this FOP version is not yet available, the module still has "beta test" status in the initial release of FirstSpirit Version 4.1. Use of Apache FOP 0.95 should only be considered following extensive tests.

- **fs-fop_0_20_5.fsm:** conventional FOP installation (default up until FirstSpirit Version 4.0). This installation is incompatible with Apache FOP 0.95[31].

Following installation of the modules, the respective "presentation channels" are available within the server properties (see also Chapter 7.3.2 page 214):

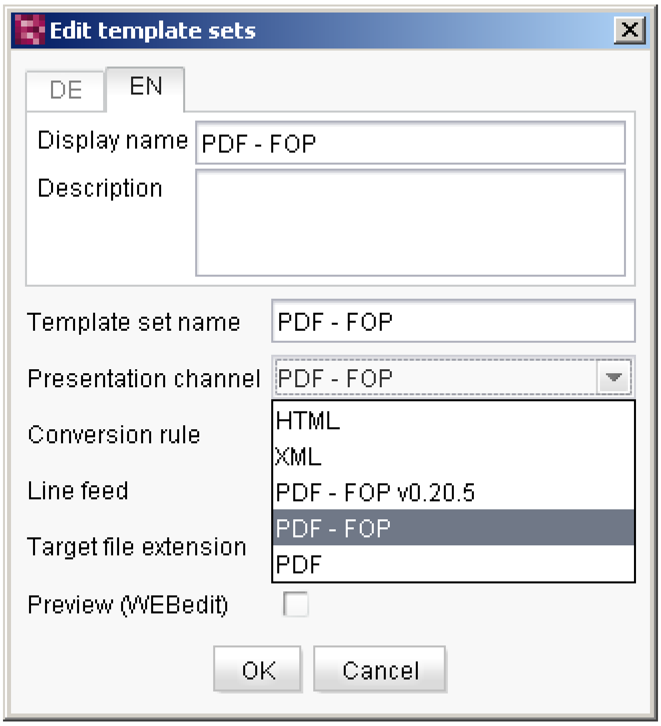


Name	System channel	File extension	Post processing
HTML	✓	html	
XML	✓	xml	
PDF - FOP v0.20.5		pdf	FOP 0-20-5
PDF - FOP		pdf	fo2pdf
PDF		fo	FOP 0-20-5

Figure 7-169: Presentation channels (server properties) following FOP installation

And can then be added into the projects as a new template set (see Chapter 7.4.13 page 282):

[31] Weiterführende Informationen siehe http://xmlgraphics.apache.org/fop/compliance.html



Edit template sets

Display name: PDF - FOP

Description:

Template set name: PDF - FOP

Presentation channel: PDF - FOP

- HTML
- XML
- PDF - FOP v0.20.5
- PDF - FOP
- PDF

Conversion rule

Line feed

Target file extension

Preview (WEBedit) □

Figure 7-170: Template sets (project properties) following FOP installation

7.9 Project archiving

This chapter describes the FirstSpirit versioning and archiving concept in greater detail to provide improved insight into the archiving schedule procedure (see also Chapter 7.5.9.1 page 328).

For information about the use of this function in comparison with the use of the module "FirstSpirit EnterpriseBackup" see also FirstSpirit Release Notes 4.2, Chapter "Long-term archiving and backup in FirstSpirit".

7.9.1 Version history

The FirstSpirit versioning and historicization concept requires that, wherever possible, all changes to objects are completely traceable in JavaClient and therefore earlier system states can be accessed at any time. Each time an object is changed, a new version of the object is created. This means an object has a version history which also provides information about which changes were made by which persons and at which time. As the individual objects in a project are in turn linked to other objects (e.g. pages consist of individual sections and are woven in the Site Store to form a navigation), changes to these linked objects are also incorporated in the version history. Only then can changes be completely traced. The "Version history" function in the context menu of an object (see also *FirstSpirit Manual for Editors*) can be used to show the version history entries relevant for this object.

7.9.2 Revisions

In addition to the version history of objects, changes logically related to the editing of repositories (see Chapter 7.5.9.1 page 328) are also recorded in FirstSpirit. This state of the whole system at a specific time is called a revision. Revisions are consecutively numbered:

- all new objects created in a revision have the same new revision number,
- changed objects are not overwritten in the repository, but instead are each added as a new object (with a higher revision number),
- all unchanged objects keep their old revision numbers.

Example:

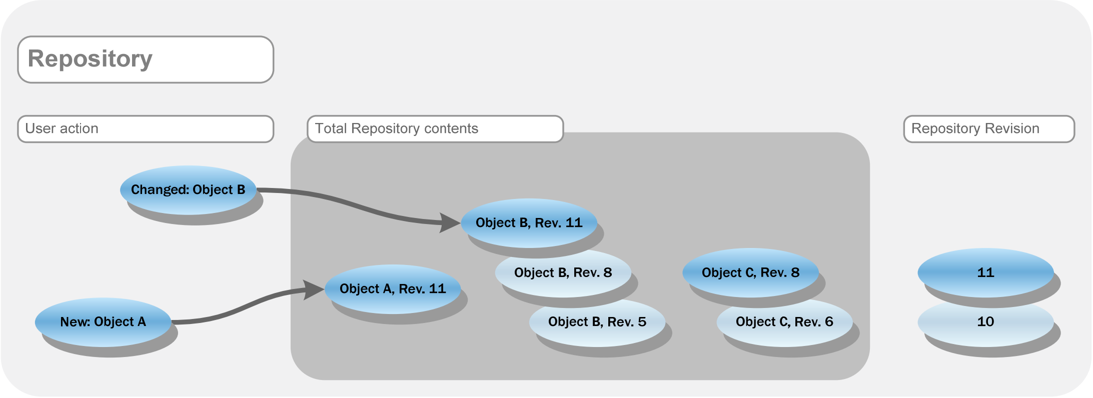


Figure 7-171: Repository revisions example

Initial state: The current repository revision is **10**.

User action: The user creates **Object A** (e.g. a new section on a page) and changes **Object B** (e.g. an existing section). **Object C** (e.g. another page) remains unchanged.

Result: If this action is transferred to the repository (e.g. by exiting the editing process on the page by means of "Save" or using the key shortcut <Ctrl> +E), a new repository revision is created with the number 11. Both Objects A and B are also assigned revision number 11. As Object A is a newly created object, it only exists in precisely one revision (11). As Object B was changed and the old state may not be overwritten, Object B now exists in several revisions (in this example, 5, 8 and 11). All unchanged objects retain their last revisions so that, for example, Object C retains Revision 8 (the last revision in which this object was changed).

7.9.3 Minimum project archiving requirement

Objects consist of different files. If an object is changed in JavaClient, at least one of the files is also changed. Individual files of an object can exist, changed in different revisions, these different revisions are therefore also part of the object.

The archiving is performed on the basis of these files, i.e. depending on the configuration of an archiving schedule by the project administrator (see Chapter 7.5.9.1 page 328), on archiving, files which are no longer required are moved out of the repository and into the archive file. The consequence of this is that following archiving, objects potentially no longer completely exist in all revisions. In this way, for example, so-called "partially archived" revisions are created.

If an **archiving period** is given (partial archiving, see Figure 7-142 page 331), all objects **within this period** including all entries in the version history are retained. All revisions within this period can be restored without restriction, including deleted objects.

All files which changed in the revisions **before the archiving period**, and all files of all revisions in the case of complete archiving, are then examined to see whether they are still currently required in the project. The point in time up until which all files are completely retained is assumed to be the **archiving limit**. Below this archiving limit, the following revisions of a file are at least retained in the project repository:

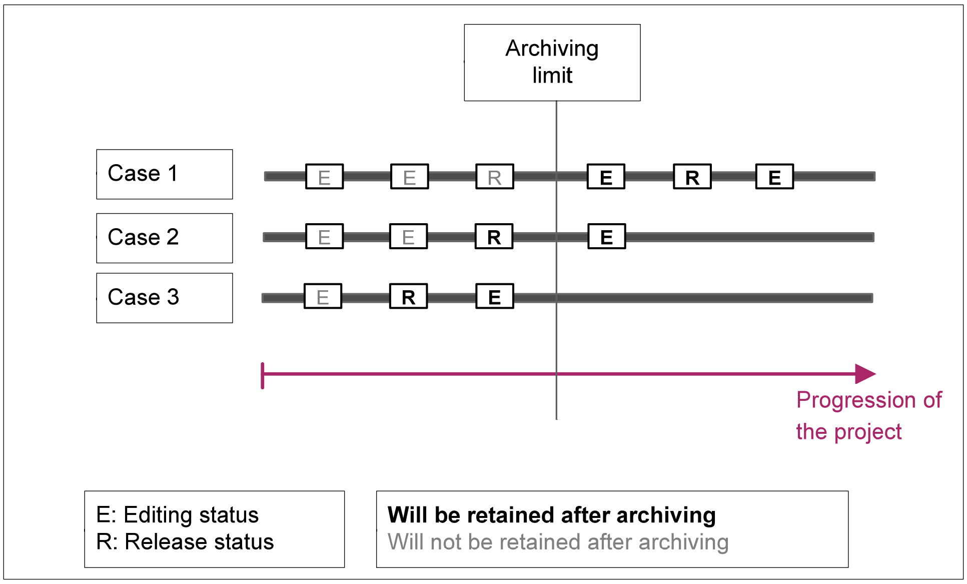

Archiving limit
Case 1
E
E
R
E
R
E
Case 2
E
E
R
E
Case 3
E
R
E
Progression of the project
E: Editing status
R: Release status
Will be retained after archiving
Will not be retained after archiving

Figure 7-172: Minimum project archiving requirement

- Case 1: If the editing and release status of the file is above the archiving limit, the revision of the file with the current editing status, the last release status and the editing status before that are always retained.
- Case 2: If the release status of the file is not available above the archiving limit, in addition to the last editing status, the most recent released status of the file below the archiving limit is always retained.
- Case 3: If neither the release nor the editing status of the file is available above the archiving limit, the most recent editing status and the most recent release status of the file below the archiving limit are retained.

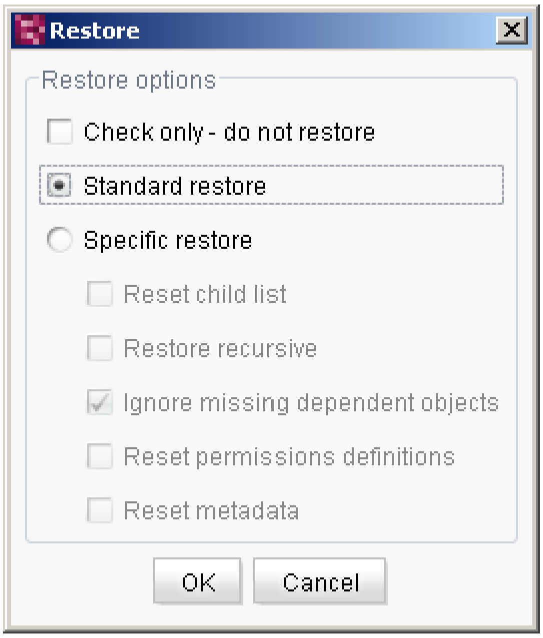
7.9.4 Version history following archiving

If deleted objects (content, media, templates, system data) are swapped out of the project during archiving, the respective revision in which the object existed is no longer complete.

The version history of an object then only displays the oldest completely available revision and all younger revisions. "Partially archived" revisions can be displayed using the "Show concealed revisions" option.

7.9.4.1 Restore following archiving

If the revision to be restored lies within the archived period, the "Specific restoration – Ignore missing dependent objects" option is preset as a default and cannot be deselected. In this way, missing references to the selected object are ignored when the object is restored:



Restore

Restore options

☐ Check only - do not restore

● Standard restore

○ Specific restore

☐ Reset child list

☐ Restore recursive

☑ Ignore missing dependent objects

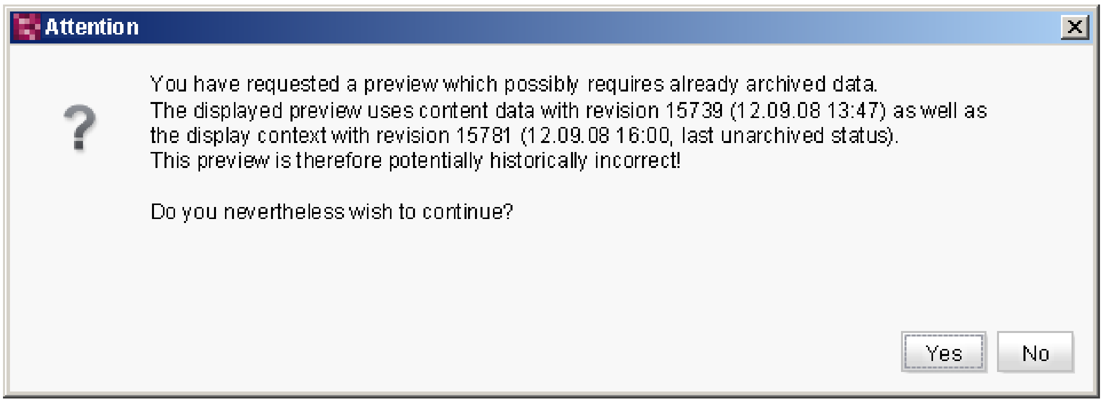
☐ Reset permissions definitions

☐ Reset metadata

Figure 7-173: Restoration following archiving

7.9.4.2 Preview of archived revisions

If data of linked objects (e.g. parent objects) are required for the preview of an object; however, this data has been archived, the most recent complete version is always used for a preview. In this case the following message is displayed:



Attention

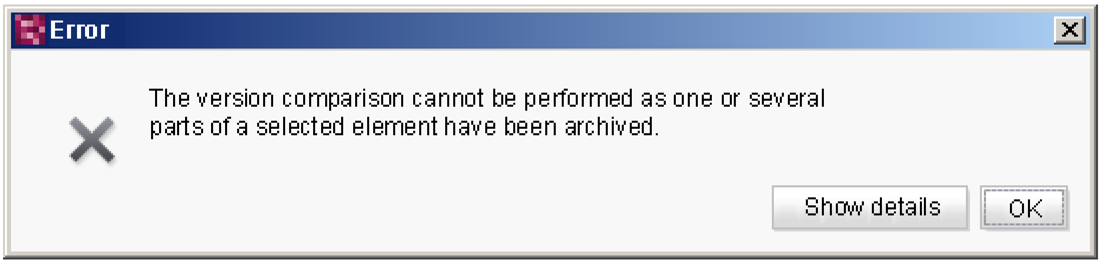
You have requested a preview which possibly requires already archived data.
The displayed preview uses content data with revision 15739 (12.09.08 13:47) as well as
the display context with revision 15781 (12.09.08 16:00, last unarchived status).
This preview is therefore potentially historically incorrect!

Do you nevertheless wish to continue?

Figure 7-174: Message issued for historically incorrect preview

7.9.4.3 Comparison of revisions following archiving

The following message is displayed if all parts of the objects necessary for a comparison are no longer available following archiving:



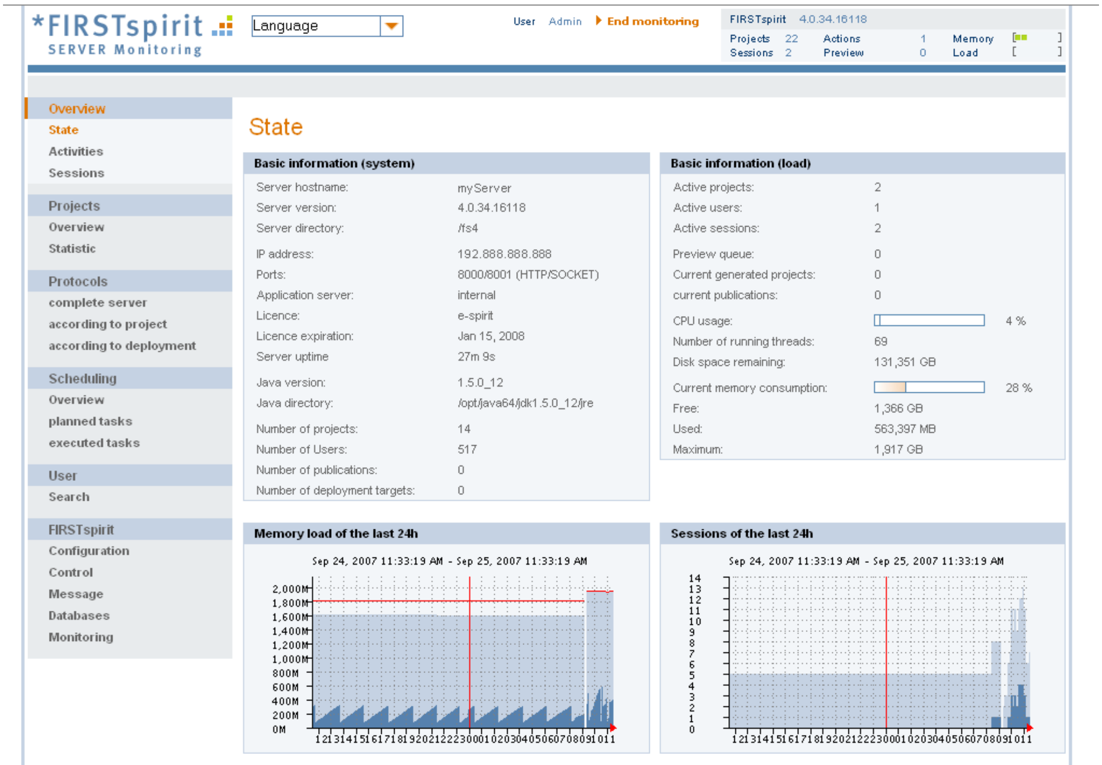
The version comparison cannot be performed as one or several parts of a selected element have been archived.

Figure 7-175: Message issued for version comparison following archiving

8 FirstSpirit Server Monitoring

The browser-based FirstSpirit Server Monitoring is a web application for monitoring the FirstSpirit Server. Analogue to the WebClient, FirstSpirit Server Monitoring is operated via a web browser.

> *The simultaneous login via a web browser (e.g. in different windows or tabs) to several FirstSpirit servers with the same host name (e.g. myServer:8200 and myServer:8400) is not supported.*



Basic information (system)

Server hostname:	myServer
Server version:	4.0.34.16118
Server directory:	/fs4
IP address:	192.888.888.888
Ports:	8000/8001 (HTTP/SOCKET)
Application server:	internal
Licence:	e-spirit
Licence expiration:	Jan 15, 2008
Server uptime	27m 9s
Java version:	1.5.0_12
Java directory:	/opt/java64/jdk1.5.0_12/jre
Number of projects:	14
Number of Users:	517
Number of publications:	0
Number of deployment targets:	0

Basic information (load)

Active projects:	2	
Active users:	1	
Active sessions:	2	
Preview queue:	0	
Current generated projects:	0	
current publications:	0	
CPU usage:		4 %
Number of running threads:	69	
Disk space remaining:	131,351 GB	
Current memory consumption:		28 %
Free:	1,366 GB	
Used:	563,397 MB	
Maximum:	1,917 GB	

Figure 8-1: FirstSpirit Server Monitoring

FirstSpirit Server Monitoring navigation is located on the left. Some menu levels provide further submenus which are displayed in a different navigation at the top of the page.

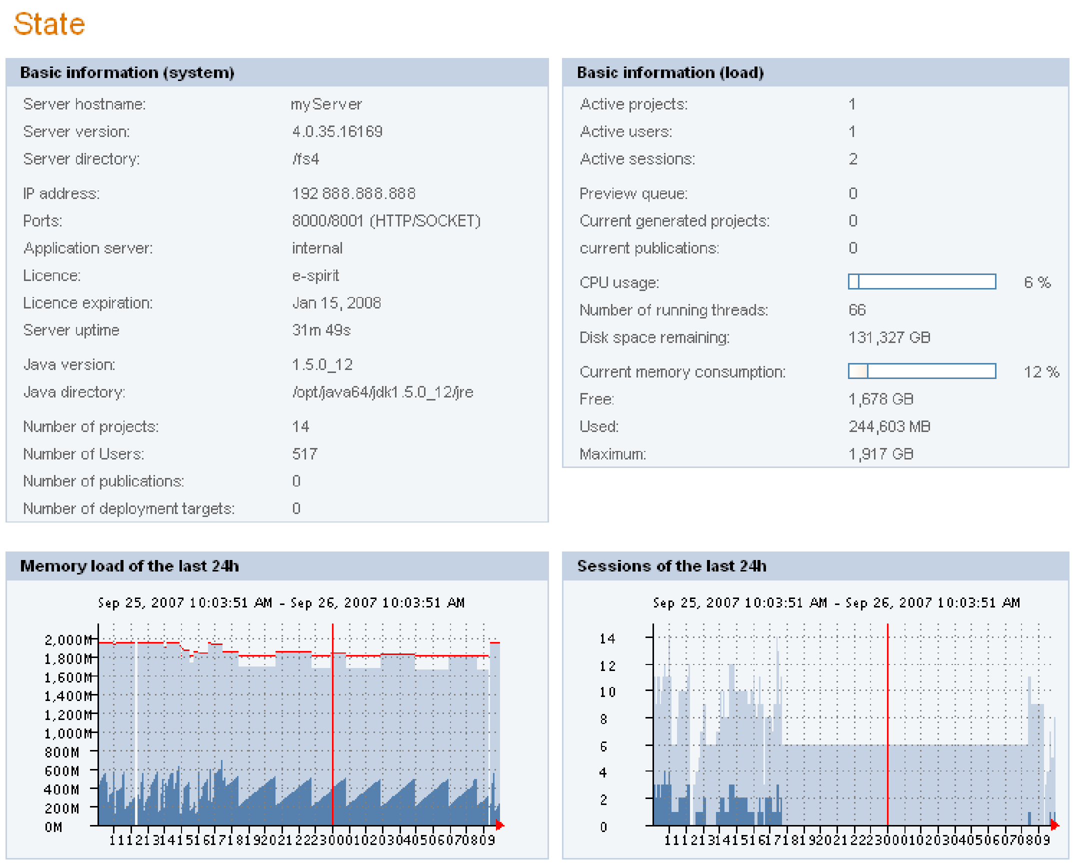
From FirstSpirit Version 4.1: Information about the current server status can also be sent via (text) e-mail (see Chapter 8.3.1.4 page 387).

The full functional scope for server administrators is documented here. Project administrators can also access FirstSpirit Server Monitoring, even though some menu levels are unavailable to them.

From FirstSpirit Version 4.1: Release of the new Look & Feel for the FirstSpirit Server Monitoring (see Chapter 1.3 page 19).

8.1 Overview

8.1.1 Overview – State



State

Basic information (system)	
Server hostname:	myServer
Server version:	4.0.35.16169
Server directory:	/fs4
IP address:	192 888.888.888
Ports:	8000/8001 (HTTP/SOCKET)
Application server:	internal
Licence:	e-spirit
Licence expiration:	Jan 15, 2008
Server uptime	31m 49s
Java version:	1.5.0_12
Java directory:	/opt/java64/jdk1.5.0_12/jre
Number of projects:	14
Number of Users:	517
Number of publications:	0
Number of deployment targets:	0

Basic information (load)		
Active projects:	1	
Active users:	1	
Active sessions:	2	
Preview queue:	0	
Current generated projects:	0	
current publications:	0	
CPU usage:		6 %
Number of running threads:	66	
Disk space remaining:	131,327 GB	
Current memory consumption:		12 %
Free:	1,678 GB	
Used:	244,603 MB	
Maximum:	1,917 GB	

Figure 8-2: FirstSpirit Server Monitoring – Overview – State

This page contains an overview of the most important information. The page is divided into four areas:

Basic information (system): This area displays general information on the server configuration.

Basic information (load): This area displays general information on the current server load.

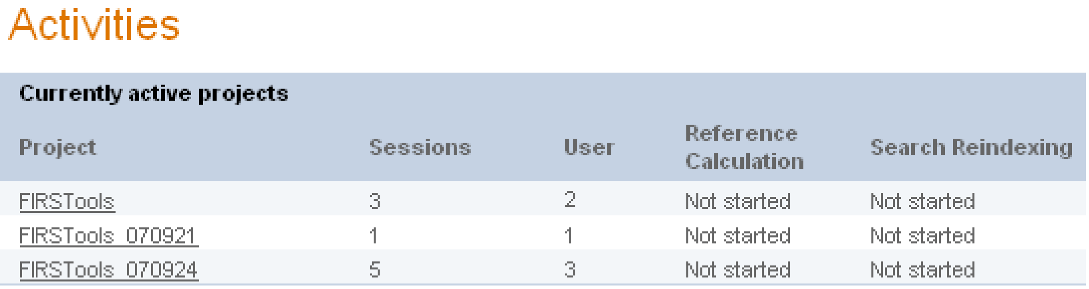
Memory load of the last 24h: This area provides an overview of the server memory load of the last 24 hours. Click on the chart to go to section "FirstSpirit – Monitoring – Memory" (see section 8.6.5.1 page 406 ff).

Sessions of the last 24h: This area provides an overview of the number of sessions during the last 24 hours. Click on the chart to go to section "FirstSpirit – Monitoring – Sessions" (see section 8.6.5.2 page 406 ff).

8.1.2 Overview – Activities

This page provides an overview of certain server activities.

Currently active projects: This table lists projects which are currently active on the server. The following information is displayed:



Activities

Currently active projects

Project	Sessions	User	Reference Calculation	Search Reindexing
FIRSTools	3	2	Not started	Not started
FIRSTools_070921	1	1	Not started	Not started
FIRSTools_070924	5	3	Not started	Not started

Figure 8-3: FirstSpirit Server Monitoring – Overview– Activities

Click on an active project to display a detailed project overview (see section 8.2.1 page 380 ff).

- Sessions: Number of currently active sessions in the project.
- User: Number of currently logged-in users in the project.
- Reference Calculation: State of the reference calculation in the project (see section 9.12).
- Search Reindexing: State of reindexing for the Search function (see chapter 9.14).

Currently running schedule entries: This table lists all the schedule entries which are currently active on the server (see section 7.5 page 310).

FirstSpirit Version 4.2.R4: In this area, information about the recalculation and conversion of references (which are existing in an old version of the reference graph) are displayed, which is executed automatically when projects are opened for the first time after an upgrade to FirstSpirit 4.2R4, too (cf. Chapter 9.12 page 434).

Currently planned schedule entries: This table lists all future, active (not manual) schedule entries (see section 7.5.1.2 page 312).

Scheduling of the last 24 hours (max. 15 entries): This table lists all the schedule entries executed during the last 24 hours.

8.1.3 Overview – Sessions

This page contains a table listing all currently running sessions.

Session ID: Unique session ID

Type: Session type. The following session types are differentiated:

- WEB: The session is established via the start page.
- WebEdit: The session is established via the WebClient.
- Main: The session is established via the JavaClient.
- Child: The session is established via the WebClient.
- WEBmonitor: The session is established via Server Monitoring.
- Dummy: Internal sessions which are, e.g., established during generation execution. These sessions are only displayed if the box ☑ Show dummy sessions has been checked. These sessions are not included for the evaluation of licensed sessions (see section 4.3.5 page 87).
- Remote: The session is established via a remote project.
- Staging: The session is established while generating a preview of the generated project state (via the URL specified during generation). These sessions are not included for the evaluation of licensed sessions (see section 4.3.5 page 87).

> *In a session of the type Child it is possible to send a message to the user of the session. Click the Session Type to open the "FirstSpirit – Message" page (see chapter 8.6.3 page 404 ff.). In this case however the message is not sent to all users on the server but only to the users of the selected session.*

User: User login name. Click on the user of a session to go to the “User – Search” page containing detailed user information (see section 8.5.1 page 394 ff).

Project: Current project of the respective user. Click on the project of a session to go to the “Projects – Overview” page containing detailed project information (see section 8.2.1.1 page 380 ff).

Use the “**Shutdown**” link to terminate a session without prior warning to the user. Please note that all the entries which have not been saved in the respective session will be lost.

Access: Displays the last server access time of the user.

Login time: Displays the server login time of the respective user.

8.2 Projects

8.2.1 Projects – Overview

This page contains a table of all the projects installed on the server. Besides the project name and a project description, the table lists the number of permitted users for this project and whether the project is activated or deactivated on the server. Moreover, it displays the current project state (is generated, is deployed ...) and the project ID.

It is also possible to sort this list according to the project name, user, activation state or project ID. Use the “**Active projects only**” option to hide all deactivated projects. Click on Update projects to update the project overview.

Only projects for which they have been entered as administrators are listed for project administrators.

Click on a listed project to display detailed project information.

8.2.1.1 Detailed project information

Project name, description as defined in the project properties.

The **Project ID** clearly identifies the project on the server.

Activated states whether the project has been activated or not.

Pages states the number of pages generated during the last full project generation.

Generations, deployments state the number of already executed generations (incl. part generations) and deployments.

Max. sessions state the maximum number of sessions.

Average generation duration

Last generation duration

Average page generation duration

Average preview generation duration

Last preview generation duration

The average or last duration of either a project, page or preview generation is displayed. It is, therefore, possible to judge whether recent changes have affected the project performance.

Last access, last change, last release and **start time of the statistic** state when the actions were executed.

Number of users states the number of users specified for this project.

User states the logins of users specified for this project.

Last used states the last 5 project users.

Used fonts states the fonts used in the project. Only the fonts used in function CMS_FONT are listed here.

Furthermore, there is also a table listing all the groups defined for the project together with their respective users.

Show import log Click on the button to display the log file of the project import.

Show import ID map Click on the button to display the import ID map for the project. The “old” IDs are overwritten with new values during a project import. The ID map displays the replacements from “old” to “new”.

8.2.2 Projects – Statistic

This page is subdivided as follows:

- Accesses: Access statistic (see section 8.2.2.1 page 382 ff)
- Deployment: Deployment statistic (see section 8.2.2.2 page 382 ff)
- Resources: Resource statistic (see section 8.2.2.3 page 383 ff)

8.2.2.1 Accesses

This page contains information on accesses to the installed projects. A table lists all the projects installed on the server. Besides the project name, the time of the last access and the last project change are stated. The last user to access the project, the number of permitted project users and the maximum number of simultaneously active sessions are also displayed.

This list can be sorted according to project name, user, last access, last change or maximum sessions. Use the “**Active projects only**” option to hide all deactivated projects.

> ! *Only projects for which they have been entered as administrators are listed for project administrators.*

Click on a listed project to display detailed project information. (See section 8.2.1.1 page 380 ff)

8.2.2.2 Deployment

This page contains information on the executed deployments of the projects. The table lists all the projects installed on the server. Besides the project name, the number of generations, the number of the executions, the last deployment duration and the average deployment duration are displayed.

This list can be sorted according to all the table columns. Use the “**Active projects only**” option to hide all deactivated projects.

> ! *Only projects for which they have been entered as administrators are listed for project administrators.*

Click on a listed project to display detailed project information. (See section 8.2.1.1 page 380 ff)

8.2.2.3 Resources

This page contains resources required by the projects. The table lists all the projects installed on the server. Besides the project name, the number of project pages and the required disk memory space are displayed. The number of executed previews as well as the duration for calculating a preview page and for page generation are also displayed.

This list can be sorted according to all the table columns. Use the "**Active projects only**" option to hide all deactivated projects. Use the "**Update projects"** function to update the project overview.

! *Only projects for which they have been entered as administrators are listed for project administrators.*

Click on a listed project to display detailed project information. (See section 8.2.1.1 page 380 ff)

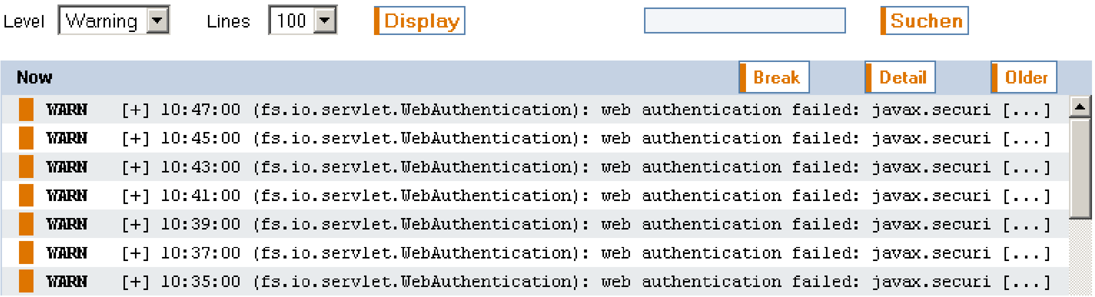
8.3 Logfiles

8.3.1 Logfiles – Complete server

! *This menu level is only available to server administrators.*

8.3.1.1 Current view

This page displays current log file actions.



```
WARN  [+] 10:47:00 (fs.io.servlet.WebAuthentication): web authentication failed: javax.securi [...]
WARN  [+] 10:45:00 (fs.io.servlet.WebAuthentication): web authentication failed: javax.securi [...]
WARN  [+] 10:43:00 (fs.io.servlet.WebAuthentication): web authentication failed: javax.securi [...]
WARN  [+] 10:41:00 (fs.io.servlet.WebAuthentication): web authentication failed: javax.securi [...]
WARN  [+] 10:39:00 (fs.io.servlet.WebAuthentication): web authentication failed: javax.securi [...]
WARN  [+] 10:37:00 (fs.io.servlet.WebAuthentication): web authentication failed: javax.securi [...]
WARN  [+] 10:35:00 (fs.io.servlet.WebAuthentication): web authentication failed: javax.securi [...]
```

Figure 8-4: Logfiles – Current view

The header contains information on the period during which server actions occurred.

"**Now**" means that the list is constantly updated with the most recent actions listed at the top.

Click on Break to stop updating the list and to take a closer look at some of the entries. to stop updating the list and to take a closer look at some of the entries. Click on the plus icon [+] to display the entry in full length.

Click on Detail to display all entries in full length at the same time.

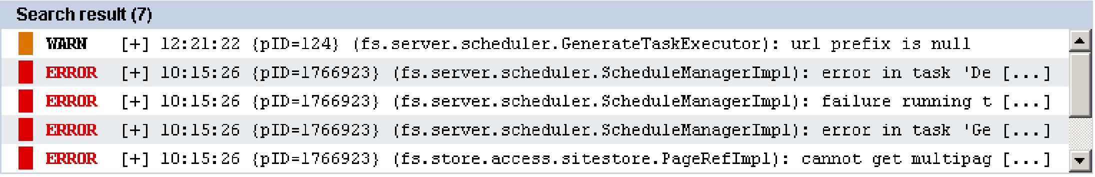
Click on Older to output an older part of the log file. (for this see section 8.3.1.2 page 384 ff)

Above the log list there are various selection options for limiting the current display:

Level: Use this field to select up to which information level the server actions are to be displayed. Level debug, Info, Warning and Error can be selected. Click on Display to update the view.

Lines: The number of server log lines to be displayed simultaneously can be selected in this field. Click on Display to update the view.

FSTicket Suchen Use the search function to search for specific text fragments in the log data. Always observe upper and lower-case spellings. The search results are displayed below the current log list.



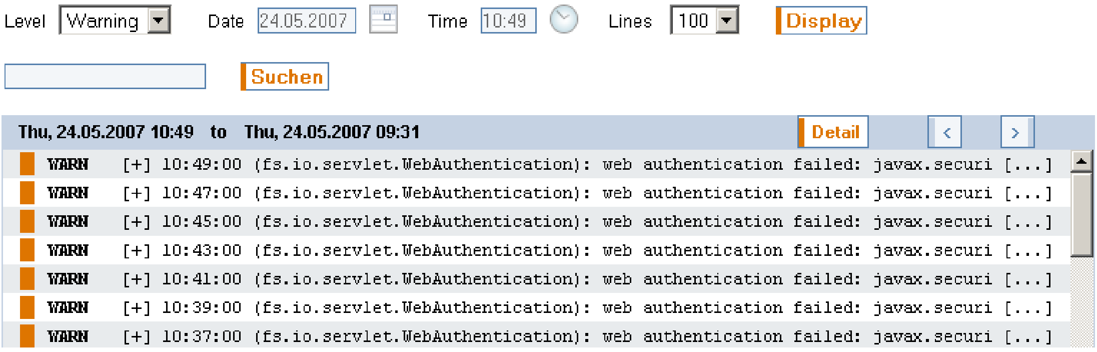
Search result (7)

```
🟧 WARN   [+] 12:21:22 (pID=124) (fs.server.scheduler.GenerateTaskExecutor): url prefix is null
🟥 ERROR  [+] 10:15:26 (pID=1766923) (fs.server.scheduler.ScheduleManagerImpl): error in task 'De [...]
🟥 ERROR  [+] 10:15:26 (pID=1766923) (fs.server.scheduler.ScheduleManagerImpl): failure running t [...]
🟥 ERROR  [+] 10:15:26 (pID=1766923) (fs.server.scheduler.ScheduleManagerImpl): error in task 'Ge [...]
🟥 ERROR  [+] 10:15:26 (pID=1766923) (fs.store.access.sitestore.PageRefImpl): cannot get multipag [...]
```

Figure 8-5: Logs – Current view – Search results

8.3.1.2 History

On this page it is possible to output an older part of the log file.



Level Warning ▾ Date 24.05.2007 Time 10:49 Lines 100 ▾ Display

Suchen

Thu, 24.05.2007 10:49 to Thu, 24.05.2007 09:31 Detail < >

```
WARN  [+] 10:49:00 (fs.io.servlet.WebAuthentication): web authentication failed: javax.securi [...]
WARN  [+] 10:47:00 (fs.io.servlet.WebAuthentication): web authentication failed: javax.securi [...]
WARN  [+] 10:45:00 (fs.io.servlet.WebAuthentication): web authentication failed: javax.securi [...]
WARN  [+] 10:43:00 (fs.io.servlet.WebAuthentication): web authentication failed: javax.securi [...]
WARN  [+] 10:41:00 (fs.io.servlet.WebAuthentication): web authentication failed: javax.securi [...]
WARN  [+] 10:39:00 (fs.io.servlet.WebAuthentication): web authentication failed: javax.securi [...]
WARN  [+] 10:37:00 (fs.io.servlet.WebAuthentication): web authentication failed: javax.securi [...]
```

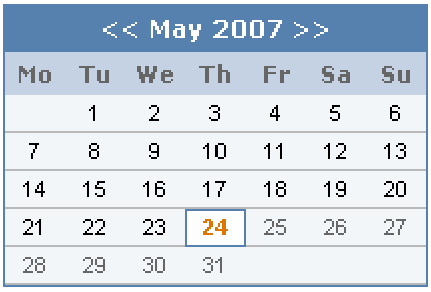
Figure 8-6: Logfiles – History

The header contains information on the period during which server actions occurred. Click on the plus icon [+] to display the entry in full length.

Click on Detail to display all entries in full length at the same time.

Level: Use this field to select up to which information level the server actions are to be displayed. Level debug, Info, Warning and Error can be selected. Click on Display to update the view.

Date: Click on ▦ to open a window for selecting a date.



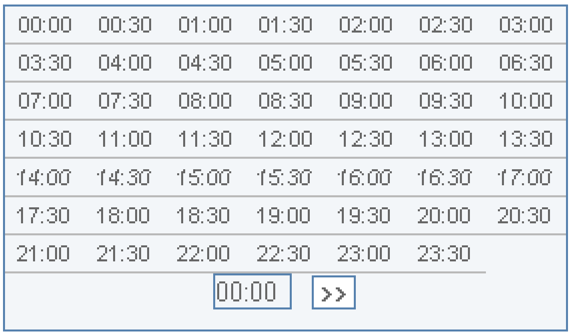
<< May 2007 >>

Mo	Tu	We	Th	Fr	Sa	Su
	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			

Figure 8-7: Date selection

Use the arrows next to the name of the month to go to the next or previous month. Click on the desired day to close the window and to accept the selected date in the field. Click on Display to update the view.

Time: Click on 🕒 to open a window for selecting a time (click the button once again to close this dialogue).



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	

Figure 8-8: Time selection

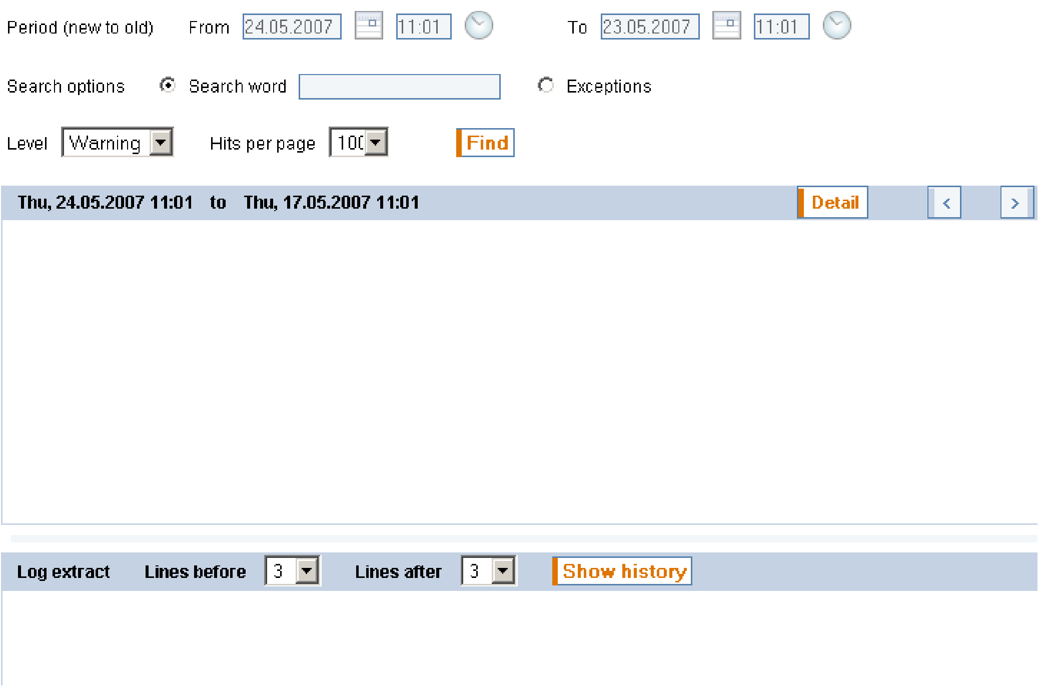
Click on the desired time to close the window and to accept the selected time in the field. Click on Display to update the view.

Lines: The number of server log lines to be displayed simultaneously can be selected in this field. Click on Display to update the view.

FSTicket Suchen Use the search function to search for specific text fragments in the log data. Always observe upper and lower-case spellings. The search results are displayed below the current log list.

8.3.1.3 Search

Use this page to output a specific period of the log file. The output can be limited by search criteria.



Period (new to old) From 24.05.2007 11:01 To 23.05.2007 11:01

Search options ● Search word ○ Exceptions

Level Warning Hits per page 100 Find

Thu, 24.05.2007 11:01 to Thu, 17.05.2007 11:01 Detail < >

Log extract Lines before 3 Lines after 3 Show history

Figure 8-9: Logfiles – Search

Period: Use the two date and time selection fields to set the period for which the log file actions are to be output. Click on 📅 to open a window for selecting the date (see Figure 8-7) and on 🕓 to open a window for selecting the time (see Figure 8-8). A period of one day is preset by default.

Search options: If **Search word** is activated, it is possible to search for specific text fragments in the log data. If **Exceptions** is activated, all specified period actions are output.

Level: Use this field to select up to which information level the server actions are to be displayed. Level debug, Info, Warning and Error can be selected. Click on Display to update the view.

Hits per page: The number of server log lines to be displayed simultaneously can be selected in this field. Click on Find to update the view.

8.3.1.4 Mail transfer (from V4.1)

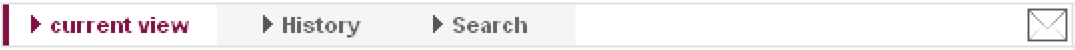
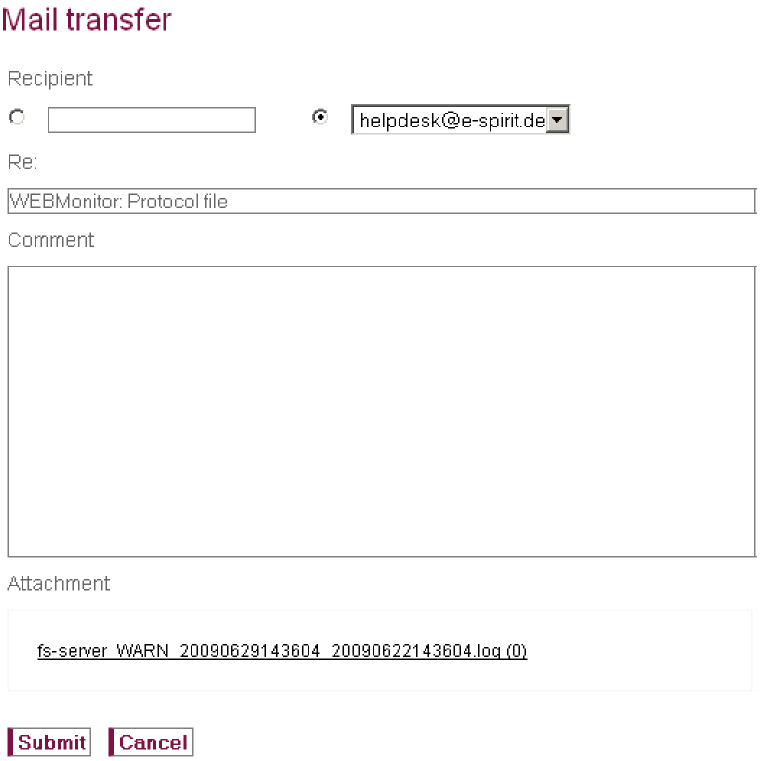


Figure 8-10: Mail transfer of a log file

In FirstSpirit Version 4.1 log files, e.g. of a deployment, can be downloaded within the Server Monitoring application and / or sent as mail attachment. If the mail transfer is opened using the icon ✉ (see Figure 8-10) the respective log files are attached automatically:



Mail transfer

Recipient

○ ________ ● helpdesk@e-spirit.de ▾

Re:

WEBMonitor: Protocol file

Comment

Attachment

fs-server_WARN_20090629143604_20090622143604.log (0)

Figure 8-11: Mail transfer with attached log file

They can also be downloaded and saved with clicking the link within this dialogue.

8.3.2 Logfiles – according to project

This page contains a table of all the projects installed on the server. The project name and a project description are displayed here.

By clicking the Select project link after each project entry the current log view of the selected project is displayed.

By clicking the Project details link further information about the selected project can be displayed (see Chapter 8.2.1.1 from page 380).

Only projects for which they have been entered as administrators are listed for project administrators.

8.3.2.1 Current view

This page displays the current log file actions for the selected project. The name of the selected project is displayed above the log list.

FIRSTools Intranet > Change

Figure 8-12: Logfiles – Change project

A different project can be selected for viewing via the Change link after the project name.

See section 8.3.1.1 page 383 ff for detailed documentation about the structure of the current log view.

8.3.2.2 History

Use this page to output an older part of the log file for the selected project. A different project can be selected for viewing via the Change link after the project name.

See section 8.3.1.2 page 384 ff for detailed documentation about the structure of the historic log view.

8.3.2.3 Search

Use this page to output a specific period of the log file for the selected project. A different project can be selected for viewing via the Change link after the project name.

See section 8.3.1.3 page 386 ff .for detailed documentation about the search for specific actions of the log view.

8.3.2.4 Mail transfer (from V4.1)

In FirstSpirit Version 4.1 log files, e.g. of a deployment, can be downloaded within the Server Monitoring application and / or sent as mail attachment (see Chapter

8.3.1.4 page 387).

8.3.3 Logfiles – according to deployment

This page contains a table of all the projects installed on the server. The project name and a project description are displayed here. The current log view of the selected project is displayed via the Select project link after each project entry.

> ! *Only projects for which they have been entered as administrators are listed for project administrators.*

Click on the project name to display detailed project information. (See section 8.2.1.1 page 380 ff)

8.3.3.1 Current view

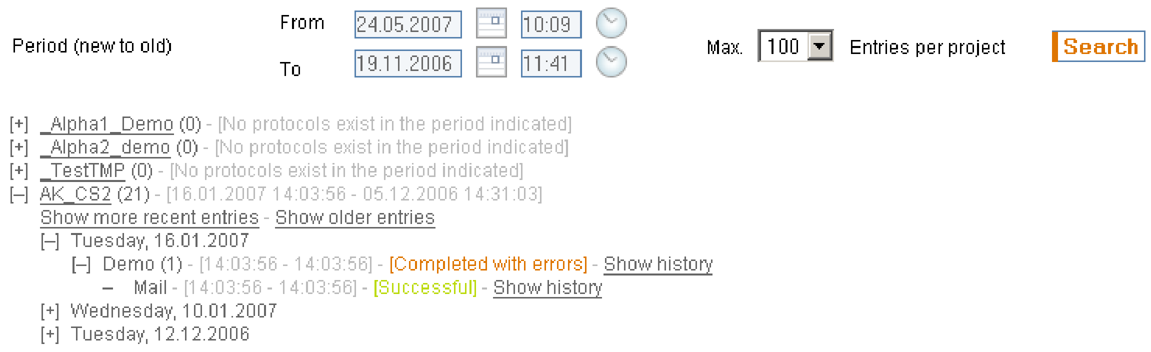
This page outputs the log file of a currently running deployment for the selected project. The name of the selected project is displayed above the log list. A different project can be selected for viewing via the Change link after the project name.

See section 8.3.1.1 page 383 ff for detailed documentation about the structure of the current log view.

8.3.3.2 History

Use this page to output the log file of an older deployment. A list of all the projects installed on the server is displayed.

Above the project list it is possible to set the period during which the searched deployments should have taken place. The maximum number of listed deployments per project can also be specified. Click on Find to update the project list.

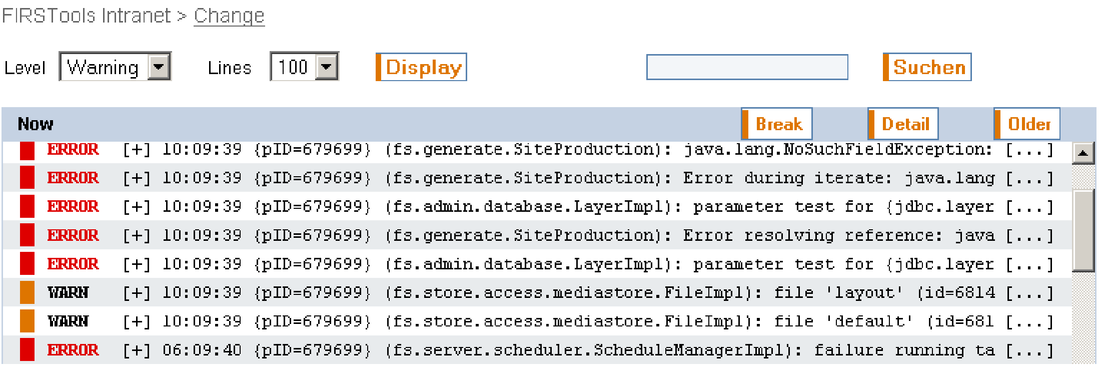


Period (new to old) From 24.05.2007 10:09 To 19.11.2006 11:41 Max. 100 Entries per project Search

```
[+] _Alpha1_Demo (0) - [No protocols exist in the period indicated]
[+] _Alpha2_demo (0) - [No protocols exist in the period indicated]
[+] _TestTMP (0) - [No protocols exist in the period indicated]
[-] AK_CS2 (21) - [16.01.2007 14:03:56 - 05.12.2006 14:31:03]
    Show more recent entries - Show older entries
    [-] Tuesday, 16.01.2007
        [-] Demo (1) - [14:03:56 - 14:03:56] - [Completed with errors] - Show history
            – Mail - [14:03:56 - 14:03:56] - [Successful] - Show history
    [+] Wednesday, 10.01.2007
    [+] Tuesday, 12.12.2006
```

Figure 8-13: Project selection for deployments

The number of log files available to the project and their creation period are displayed after the project name. Click on the plus icon [+] in front of each project name to sort the existing deployments in descending order. Click on the Show history link after each entry to display the respective log file. By clicking on the entry Download the related log file can be downloaded.



FIRSTools Intranet > Change

Level Warning Lines 100 Display Suchen

Now Break Detail Older

```
ERROR  [+] 10:09:39 {pID=679699} (fs.generate.SiteProduction): java.lang.NoSuchFieldException: [...]
ERROR  [+] 10:09:39 {pID=679699} (fs.generate.SiteProduction): Error during iterate: java.lang [...]
ERROR  [+] 10:09:39 {pID=679699} (fs.admin.database.LayerImpl): parameter test for {jdbc.layer [...]
ERROR  [+] 10:09:39 {pID=679699} (fs.generate.SiteProduction): Error resolving reference: java [...]
ERROR  [+] 10:09:39 {pID=679699} (fs.admin.database.LayerImpl): parameter test for {jdbc.layer [...]
WARN   [+] 10:09:39 {pID=679699} (fs.store.access.mediastore.FileImpl): file 'layout' (id=6814 [...]
WARN   [+] 10:09:39 {pID=679699} (fs.store.access.mediastore.FileImpl): file 'default' (id=681 [...]
ERROR  [+] 06:09:40 {pID=679699} (fs.server.scheduler.ScheduleManagerImpl): failure running ta [...]
```

Figure 8-14: Logfiles – Deployment history

Use the Change link after the project name (incl. time of deployment) to select a different project or a different deployment of the same project for viewing.

See section 8.3.1.2 page 384 ff for detailed documentation about the structure of the historic log view.

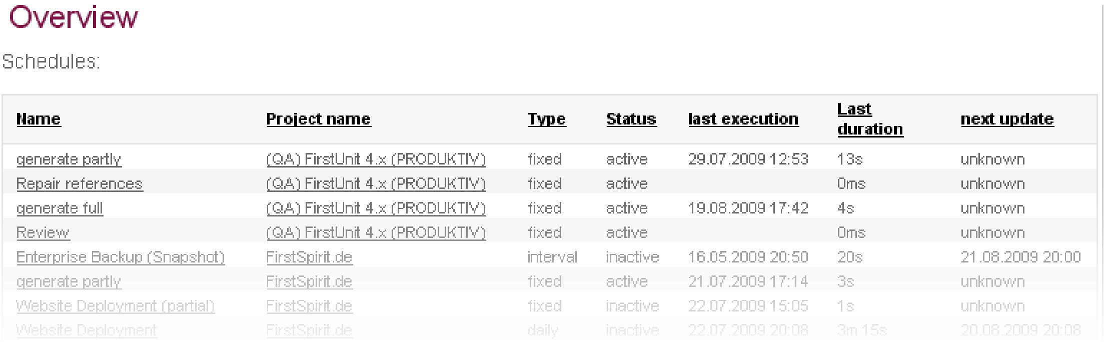
8.3.3.3 Mail transfer (from V4.1)

In FirstSpirit Version 4.1 log files, e.g. of a deployment, can be downloaded within the Server Monitoring application and / or sent as mail attachment (see Chapter 8.3.1.4 page 387).

8.4 Scheduling

8.4.1 Scheduling – Overview

This page contains an overview of the entire scheduling.



Overview

Schedules:

Name	Project name	Type	Status	last execution	Last duration	next update
generate partly	(QA) FirstUnit 4.x (PRODUKTIV)	fixed	active	29.07.2009 12:53	13s	unknown
Repair references	(QA) FirstUnit 4.x (PRODUKTIV)	fixed	active		0ms	unknown
generate full	(QA) FirstUnit 4.x (PRODUKTIV)	fixed	active	19.08.2009 17:42	4s	unknown
Review	(QA) FirstUnit 4.x (PRODUKTIV)	fixed	active		0ms	unknown
Enterprise Backup (Snapshot)	FirstSpirit.de	interval	inactive	16.05.2009 20:50	20s	21.08.2009 20:00
generate partly	FirstSpirit.de	fixed	active	21.07.2009 17:14	3s	unknown
Website Deployment (partial)	FirstSpirit.de	fixed	inactive	22.07.2009 15:05	1s	unknown
Website Deployment	FirstSpirit.de	daily	inactive	22.07.2009 20:08	3m 15s	20.08.2009 20:08

Figure 8-15: Overview of schedules on the server

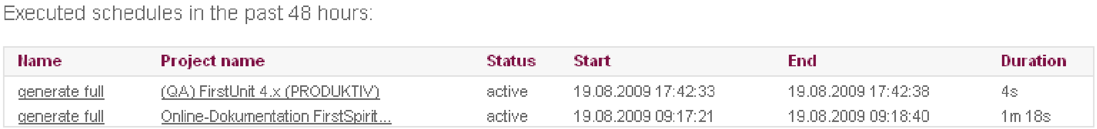
The table lists all the schedule entries set on the server (see section 7.5.1 page 311). Besides the project name, the schedule entry execution interval and the current action state are displayed. Furthermore, information on the time of the last planned execution and its duration as well as the time of the next planned execution is presented.

This list can be sorted according to all the available columns.

> ! *Only schedule entry projects for which they have been entered as administrators are listed for project administrators.*

Click on the project name of a listed project to display detailed project information (see section 8.2.1.1 page 380 ff). Click on the schedule name to display detailed information about the schedule.

Below the first table there is another table which lists the schedule entries executed during the last 48 hours, e.g. deployments.



Executed schedules in the past 48 hours:

Name	Project name	Status	Start	End	Duration
generate full	(QA) FirstUnit 4.x (PRODUKTIV)	active	19.08.2009 17:42:33	19.08.2009 17:42:38	4s
generate full	Online-Dokumentation FirstSpirit...	active	19.08.2009 09:17:21	19.08.2009 09:18:40	1m 18s

Figure 8-16: Scheduling – Execution in the last 48 hours

Besides the schedule's name, information is displayed here about the status of the executed actions and the project name for which the schedule has been executed. Furthermore, the start and end time of the schedule's execution are displayed as well as the duration of the last execution.

Log files of the deployment schedules are available via the menu item „Logfiles – according to deployment " (see Chapter 8.3.3 Seite 390).

8.4.2 Scheduling – Planned schedule entries

A table lists all the schedule entries for which an execution is still planned. Besides the project name, the schedule ID and the planned action type are stated. The execution interval of the schedule entries and the planned time of the next execution are also displayed.

This list can be sorted according to all the available columns.

Click on the project name of a listed project to display detailed project information. (See section 8.2.1.1 page 380 ff)

8.4.3 Scheduling – Executed schedule entries

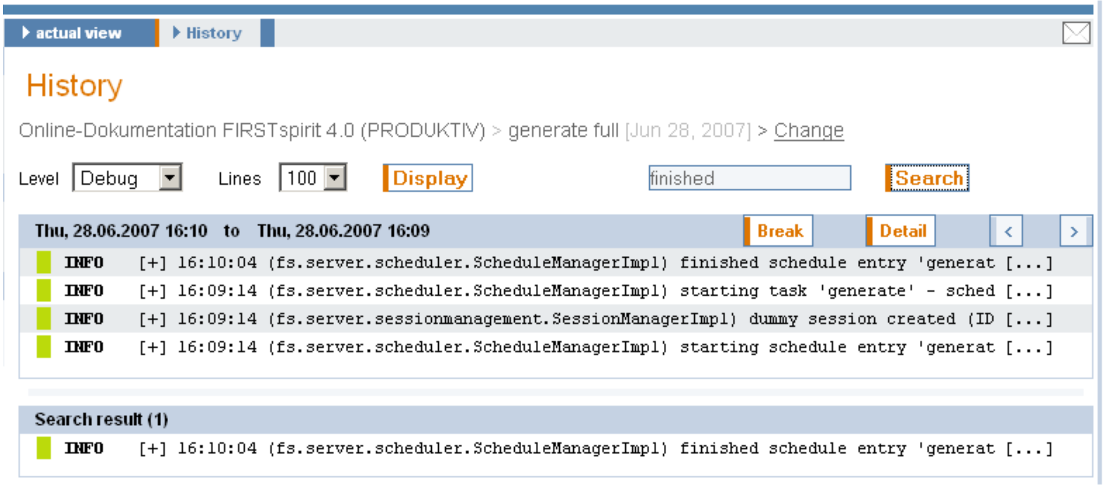
This page contains an overview of previously executed schedule entries. A maximum of 25 schedule entries is presented at one time. However, it is possible to use the arrow above the table to leaf through the individual pages.

A table lists all the schedule entries which have already been executed. Besides the schedule entry name, the name of the project for which the schedule entry has been executed is stated. The start time of schedule entry execution and the duration of the last execution are also displayed.

The list can be sorted according to project name, schedule entry name or execution start.

Click on the project name of a listed project to display detailed project information. (See section 8.2.1.1 page 380 ff)

Click on the Show history link to display log outputs at the time of schedule entry execution:



History

Online-Dokumentation FIRSTspirit 4.0 (PRODUKTIV) > generate full [Jun 28, 2007] > Change

Level: Debug   Lines: 100   Search field: finished

Thu, 28.06.2007 16:10 to Thu, 28.06.2007 16:09

```
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 [...]
```

Search result (1)

```
INFO   [+] 16:10:04 (fs.server.scheduler.ScheduleManagerImpl) finished schedule entry 'generat [...]
```

Figure 8-17: Scheduling – History of executed schedule entries

The log output can be configured via the check boxes:

Therefore, it is possible to set various **log levels**. For example, if you wish to search for error messages during generation, set the log level to “Error” (for this see section 4.3.6 page 89).

The maximum number of displayed **lines** of the log output can also be configured. Click on Display to update the display.

Break Click on the button to stop the line output.

Detail Click on the button to change from line-limited output to extended output.

Besides a filtered output, it is also possible to search directly for specific terms in the log. Click on Find to display the search result at the bottom of the page under **Search results**.

8.5 User

8.5.1 User – Search

This page displays a sorted list of all the users specified on the server. A maximum of 25 users is presented at one time. However, it is possible to use the arrow above the table to leaf through the individual pages.

Besides the user name, the abbreviation, time of the last login and whether the user is an LDAP user are also stated. Furthermore, the email address and telephone number are displayed.

Use the search function to limit the selection of displayed users. Words or parts of a word which are searched for in the user name column can be entered in the search field. Click on Submit to execute the search.

Click on a user to display his/her detailed information. This also includes, besides specifications set during user creation (see section 7.2.4.1 ab page 204 ff), a table of all the projects available to the user on this server. The user's group membership is stated for each project. Click again on a project of the selected user to display a detailed overview of this project. (See section 8.2.1.1 page 380 ff)

Click on the user's email address to open an email window for sending a message to this user.

8.6 FirstSpirit

8.6.1 FirstSpirit – Configuration

Configuration of the FirstSpirit Server, the database connection, the login process and numerous settings takes place via special configuration files in the FirstSpirit Server installation directory. It is not recommended to edit these configuration files directly (see section 4.2 page 31).

The FirstSpirit Server and Project Configuration facilitates editing of the configuration settings when configuring the database connection (fs-database.conf) and the login process (fs-jaas.conf) (see section 5.3 page 164). Further configuration settings can be carried out via the JMX console (see chapter 9 page 390).

Further configuration settings can be carried out via the JMX console (see:

• Server	(File: fs-server.conf)	Section 8.6.1.1 Page 396
• Licence	(File: fs-license.conf)	Section 8.6.1.2 Page 396
• Logging	(File: fs-logging.conf)	Section 8.6.1.3 page 397
• System	(no configuration)	Section 8.6.1.4 page 398
• Start options	(File: fs-wrapper.conf)	Section 8.6.1.5 page 398
• Web applications	(File: fs-webapps.xml)	Section 8.6.1.6 page 399
• Services	(config. system services)	Section 8.6.1.7 page 400
• Login configuration	(File: fs-jaas.conf)	Section 8.6.1.8 page 401

This menu level is only available to server administrators.

8.6.1.1 Server

FirstSpirit Server configuration occurs via the configuration files `fs-server.conf` located in the FirstSpirit Server installation directory (see section 4.3, page 33).

Click on "Server" to open the file for editing. The individual parameters for configuring the FirstSpirit Server are described in section 4.3.1 ff. A server restart might be necessary to implement certain changes (e.g. a port change), further changes can be carried out during server operation. The changes have to be saved and the configuration file reloaded on the server.

Click on Save to save the current server configuration changes and to import them into the running server operation.

Click on Cancel to return to the configuration overview. Current configuration file changes which have not yet been saved are reset.

Restart server: If this option is activated, click on Save to save the current server configuration changes and to subsequently restart the server. (The FirstSpirit Server start is ensured via the configuration file `fs-wrapper.conf` see section 8.6.1.5).

8.6.1.2 Licence

The configuration file `fs-license.conf` is located in the FirstSpirit Server subdirectory `conf`. The file contains the current FirstSpirit licence and can be displayed via FirstSpirit Server Monitoring by clicking on "Licence". The respective parameters are described in section 4.3.5, page 87 fft.

If a new licence file is to be imported onto the server, the content has to be inserted on this page in full and unchanged. Click on Save to save the new licence file.

Click on Cancel to return to the configuration overview. Current licence file changes which have not yet been saved are reset.

Configuration file changes are automatically updated on the server at set intervals during runtime. Therefore, the FirstSpirit Server does not have to be restarted.

Tampering with the `fs-license.conf` *results in an invalid licence. If changes are necessary (e.g. IP address change), please contact the manufacturer.*

8.6.1.3 Logging

The file `fs-logging.conf` is located in the FirstSpirit Server subdirectory `conf` and contains important configuration settings for the "log" outputs. It must be adapted, if necessary. The respective parameters are described in section 4.3.6, page 89 ff.

Occurred errors and info messages are transferred to the logging system "log4j"[32] by default. The log outputs can be categorised via the framework. The categories (log levels) `DEBUG`, `INFO`, `ERROR` are available. It is, however, also possible to configure additional categories (e.g. `FATAL`, `WARN`). The two stages `ALL` and `OFF`, which either deactivate the logging completely (`OFF`) or output all messages unfiltered (`ALL`), are an exception. The filter and type of output can be configured during runtime.

Besides the default logging, further log files can be configured. The status of a file (active | inactive) is stated in brackets after each project file name. Click on Verwenden to activate an inactive log file. The previously activated log file is then deactivated, since only one logging can be active at a time.

Click on Edit to change the log file.

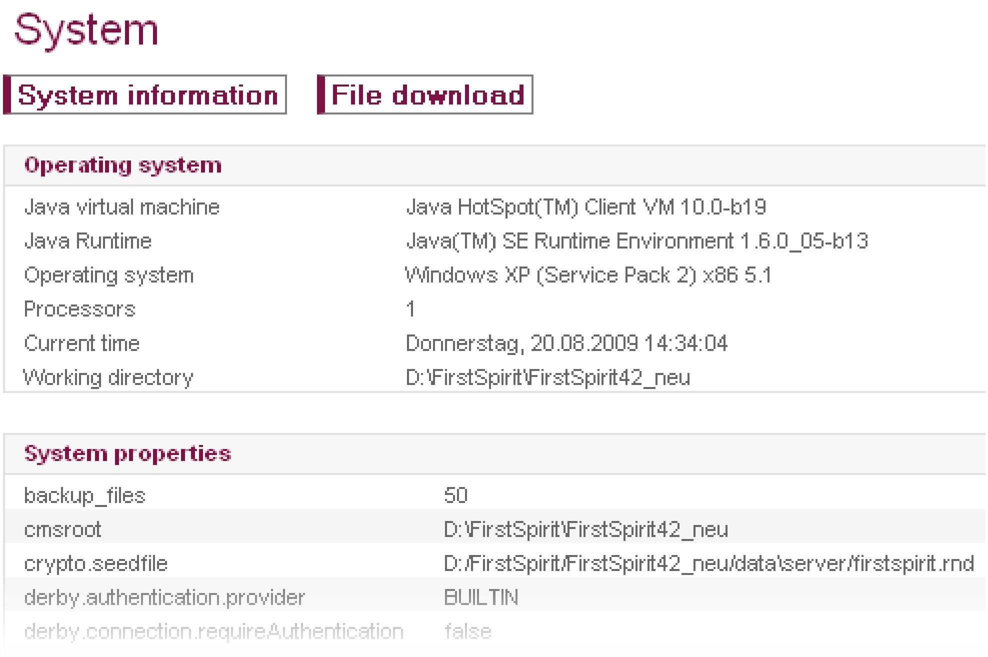
Click on Save to save the changes. Configuration file changes are automatically updated on the server at set intervals during runtime. Therefore, the FirstSpirit Server does not have to be restarted.

Click on Cancel to return to the configuration overview. Current configuration file changes which have not yet been saved are reset.

[32] Further information http://logging.apache.org/log4j/docs/index.html

8.6.1.4 System

This page contains all relevant server system information.



System

System information | File download

Operating system	
Java virtual machine	Java HotSpot(TM) Client VM 10.0-b19
Java Runtime	Java(TM) SE Runtime Environment 1.6.0_05-b13
Operating system	Windows XP (Service Pack 2) x86 5.1
Processors	1
Current time	Donnerstag, 20.08.2009 14:34:04
Working directory	D:\FirstSpirit\FirstSpirit42_neu

System properties	
backup_files	50
cmsroot	D:\FirstSpirit\FirstSpirit42_neu
crypto.seedfile	D:/FirstSpirit/FirstSpirit42_neu/data\server/firstspirit.rnd
derby.authentication.provider	BUILTIN
derby.connection.requireAuthentication	false

Figure 8-18: Server Monitoring – System information and file download

Operating system: Information on the operating system is displayed here, e.g. used JDK, server operating system, number of processors.

System properties: System properties (information on directory paths, port numbers, class path, etc.) are displayed in a clearly structured list here.

The displayed system properties cannot be edited in this overview, but are for information purposes only.

System information Click on this button to display further system information.

File download Click on this button to open the file selection dialogue of the server log directory (.../server/log). Here you can download the server log files (fs-server.log) and other log files which are not accessible via the Server Monitoring nicht (e.g. fs-gc.log).

8.6.1.5 Start options

The file `fs-wrapper.conf` is located in the FirstSpirit Server subdirectory `conf` and contains important configuration settings for the server start and has to be adapted, if necessary. The respective parameters are described in section 4.3.2, page 62 ff.

The configuration file is responsible for starting and stopping the Java process. Depending on the configuration, it is possible to define the VM start and to additionally write respective log outputs into the default output.

Click on Edit to edit the existing configuration. A window opens for editing the contents of the configuration file `fs-wrapper.conf`.

List Click on the button to exit the editing mode and to return to the list view.

Click on Save to save the changes in the configuration file `fs-wrapper.conf`. If the modification is not valid an error is displayed in Server Monitoring and the save operation will not been executed:

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

Click on Cancel to return to the configuration overview. Current configuration file changes which have not yet been saved are reset.

Each change to the configuration file `fs-wrapper.conf` requires a server restart. (In this special case, the server restart cannot be controlled via the GUI of the Server Monitoring but has to take place via the command line.)

8.6.1.6 Web applications

The file `fs-webapp.xml` is located in the FirstSpirit Server subdirectory `conf` and contains important configuration settings for the internal web server and has to be adapted, if necessary. The respective parameters are described in section 4.3.7, page 92 ff.

Click on Save to save changes in the file `fs-webapp.xml`.

Click on Cancel to return to the configuration overview. Current configuration file changes which have not yet been saved are reset.

The Jetty has to be restarted to update the changes on the server. After saving, a message and the link to the menu item Control / Web applications appear (see section 8.6.2.2 page 402). Here it is possible to restart the Jetty (in the Server Monitoring environment).

8.6.1.7 Services

The FirstSpirit system services can be configured via this entry. A service is a server component which can be addressed via a public interface consisting of input components or scripts. (Examples for this are the spell check or the permission service input component CMS_INPUT_PERMISSION) (see section 7.3.14 page 232).

The table lists all the services available on the server.

Overview of the configurable services:

Name: Name of the system service. Click on the entry to open another window with configuration options for the respective service (see “Service configuration” below).

Comment: Optional comment about the service.

Type: Name of the module to which the service belongs. Services described as “system” belong to the FirstSpirit default system module.

Autostart: “Activate” or “deactivate” a service in this area. The function can be used analogue to “Start service” or “Stop service” in the Server and Project Configuration (see section 7.3.14 page 232).

Service configuration:

File name: File name of the service. Click on the file name to open another window for editing the file, e.g. the groups.xml file of the PermissionService.

Click on Save to save the changes in the file.

Click on Cancel to return to the file overview. Current changes which have not yet been saved are reset.

Size: File size

Last change: Date of the last saved file version.

Create directory A directory for the respective service can be specified here. The directory is created with the entered name in the file system underneath the module directory.

Create file It is possible to create a file for the respective service here. The file is created with the entered name in the file system underneath the module directory.

8.6.1.8 Login configuration

The file `fs-jaas.conf` is located in the FirstSpirit Server subdirectory `conf` and contains important configuration settings for the login process and can be adapted, if necessary. The respective parameters are described in section 4.3.4, page 72 ff.

Click on Save to save the changes in the file `fs-jaas.conf`. The file is automatically updated on the FirstSpirit Server. The server does not have to be restarted.

Click on Cancel to return to the configuration overview. Current configuration file changes which have not yet been saved are reset.

8.6.2 FirstSpirit – Control

After server configuration changes or an update, the FirstSpirit Server can be controlled via the menu entry Control.

This menu level is only available to server administrators.

8.6.2.1 Maintenance mode

This function is used to activate or deactivate the FirstSpirit server maintenance mode. If the maintenance mode is activated, new users cannot login at the FirstSpirit Server. When activated, a message stating that the server will be shut down is automatically sent to all currently logged-in FirstSpirit users. Therefore, users have the opportunity to save current changes and to logout.

The server should only be stopped (see section 8.5 page 394) once all the users have logged out (see chapter 3 page 23).

This function is currently unavailable and will probably be provided with a later FirstSpirit version.

8.6.2.2 Web applications

FirstSpirit web applications can be configured via this function.

This function is currently unavailable and will probably be provided with a later FirstSpirit version.

Click on the button to restart the internal Jetty. For example, a restart is required to update changes to the file `fs-webapp.xml` on the server (see section 8.6.1.6 page 399).

8.6.2.3 Update

This function is used to update the FirstSpirit Server.

This function is currently unavailable and will probably be provided with a later FirstSpirit version.

8.6.2.4 Services

This function is used to control the FirstSpirit Server services. A service is a server component which can be addressed via a public interface consisting of input components or scripts. Examples are the spell check or the permission service input component CMS_INPUT_PERMISSION. The system module with the default services (e.g. PermissionService) is already included in the FirstSpirit standard delivery and is available after installation. It is also possible to display further services which have been installed on the server at a later date (see section 7.3.14 page 232). Services are valid throughout the system.

Name: Name of the service.

Comment: Description for the respective service.

Type: Name of the module to which the service belongs.

Start/Stop: Function for starting or stopping the service. If the service has already been started, it can only be "stopped". If the service has not yet been started, it can only be "started". This function can be used analogue to starting and stopping a service via the FirstSpirit Server and Project Configuration (see section 7.3.14 page 232).

Autostart: Function for enabling or disabling an automatic service start. Use this function analogue to "enable autostart / disable autostart" in the Server and Project Configuration (see section 7.3.14 page 232).

After the update of modules with dependencies to other modules with services, these services must restarted manually.

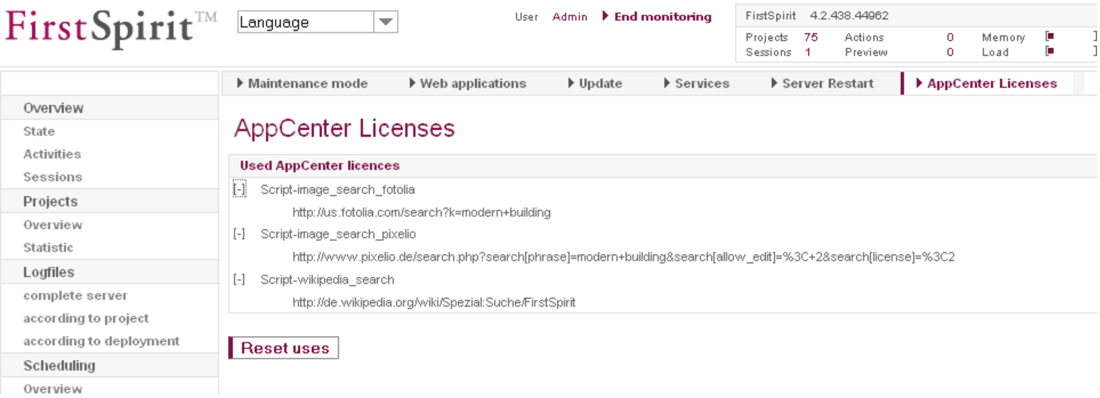
8.6.2.5 Server Restart

Click on the button Restart server to restart the server.

Please notice that no extra prompt for confirmation will be displayed before the restart of the server. The Server Monitoring should be necessarily closed after the restart.

8.6.2.6 AppCenter Licenses

In this area there is a list with all used AppCenter licenses:



FirstSpirit 4.2.438.44962
Projects 75 Actions 0 Memory
Sessions 1 Preview 0 Load

AppCenter Licenses

Used AppCenter licences

[-] 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

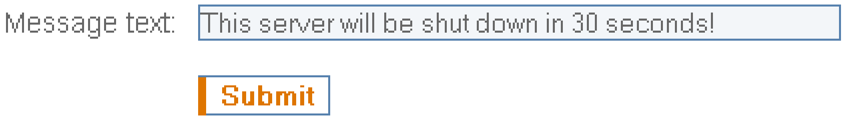
[-] Script-wikipedia_search
http://de.wikipedia.org/wiki/Spezial:Suche/FirstSpirit

Figure 8-19: Server Monitoring – AppCenter Licenses

If necessary, use the button Reset uses to reset the number of registered applications to 0. Registered applications, which are currently open in Clients, can continue to be used until the application or the corresponding tab is closed. The server does not have to be restarted.

8.6.3 FirstSpirit – Message

A message text which is transferred to all active clients (start page, Server and Project Configuration, Server Monitoring, JavaClient and WebClient) as a popup window can be edited on this page.



Message text: This server will be shut down in 30 seconds!

Figure 8-20: Send message

This is particularly useful if the server is to be shut down. This means, editors have enough time to save their work, ensuring data is not lost.

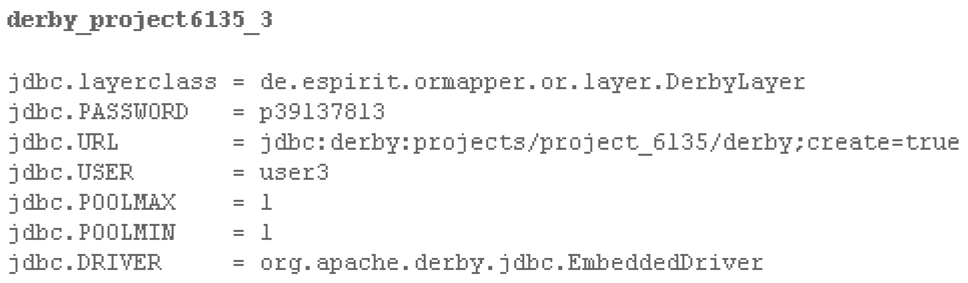
This menu level is only available to server administrators.

8.6.4 FirstSpirit – Databases

FirstSpirit saves highly structured contents of the Content-Store in a database. See section 4.8, page 124 ff.

8.6.4.1 Overview

Overview of all currently configured database connections on the 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
```

Figure 8-21: Database overview

Database connection configuration is carried out via the configuration file `fs-database.conf`. See section 4.3.3, page 71 for a parameter description. Changes to the configuration file `fs-database.conf` can be carried out via the FirstSpirit Server and Project Configuration (see section 7.3.6 page 220).

> ! *The parameters `PASSWORD` and `USER` are only displayed for server administrators.*

8.6.4.2 Status

The menu entry “Status” displays further information on a connected database. The projects which access the database are shown below the displayed database connection.

> ! *This function is currently unavailable and will probably be provided with a later FirstSpirit version.*

Use the project link below the database layer to open the detailed project page (see section 8.2.1.1 page 380).

8.6.5 FirstSpirit – Monitoring

8.6.5.1 Memory

This page displays a chart with the memory consumption of the last 24 hours.

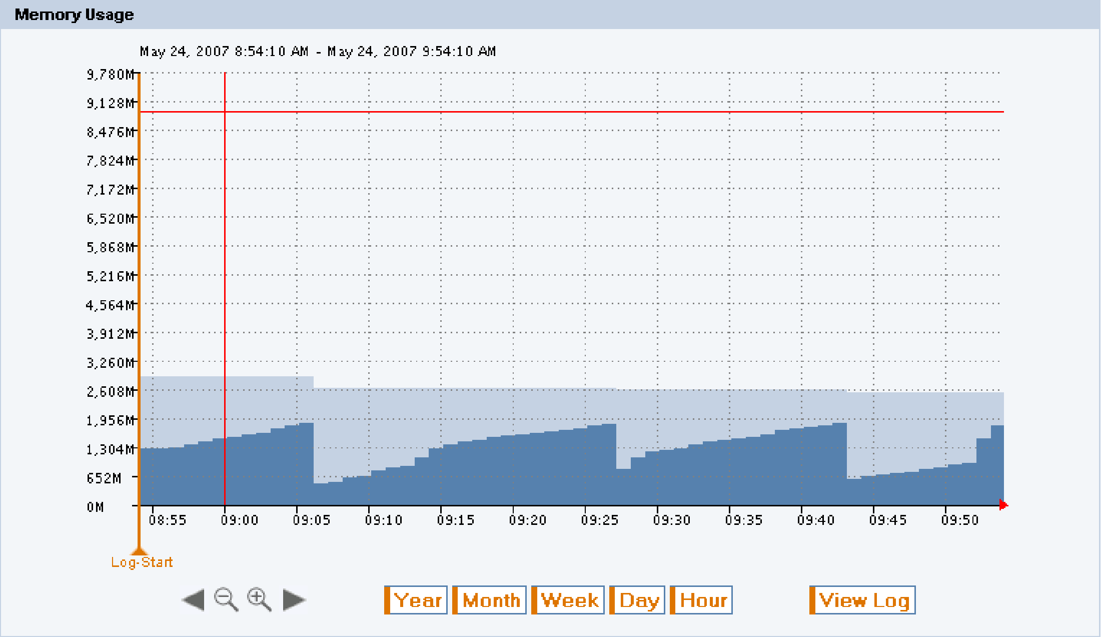


Figure 8-22: Monitoring – Memory consumption

Use the **Year**, **Month**, **Week**, **Day** and **Hour** buttons to change the memory consumption display period. Use ⊖ and ⊕ to move one category towards the year and hour button respectively. Depending on the set category, use the arrows to shift the displayed period backwards (◀) or forwards (▶) by one year (month, week…).

Click on View Log to display the server log file for the selected period. (See section 8.3.1.2 page 389 ff)

8.6.5.2 Sessions

This page displays a chart of the number of sessions which were simultaneously active on the server during the last 24 hours.

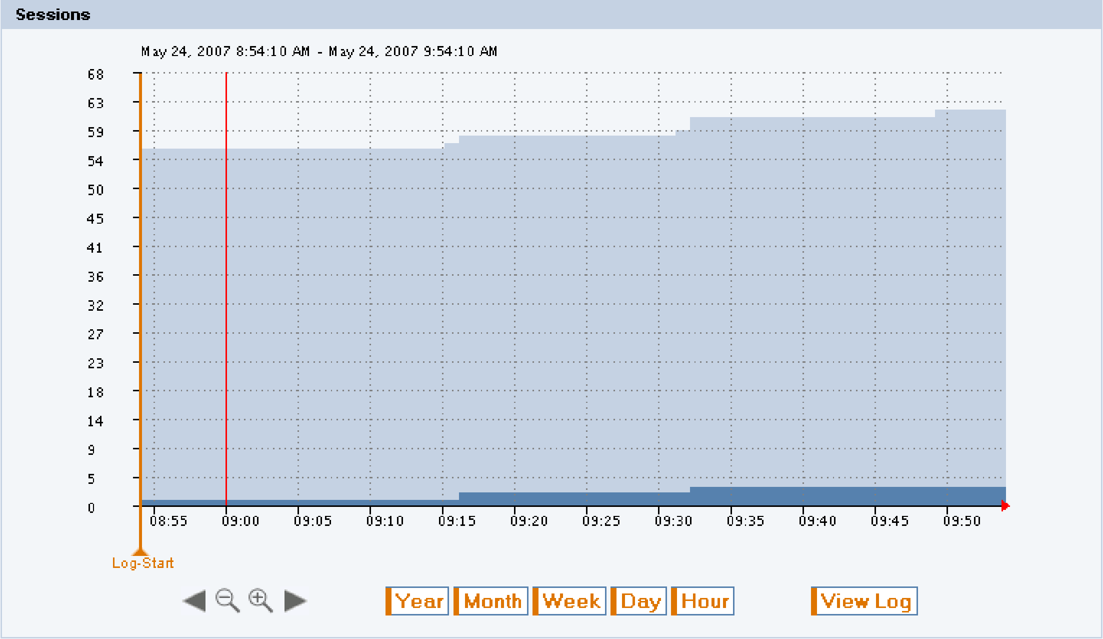


Figure 8-23: Monitoring – Sessions

Use the **Year**, **Month**, **Week**, **Day** and **Hour** button to change the display period for the active sessions. Use 🔍 and 🔍 to move one category towards the year and hour button respectively. Depending on the set category, use the arrows to shift the displayed period backwards (◀) or forwards (▶) by one year (month, week…).

Click on View Log to display the server log file for the selected period. (See section 8.3.1.2 page 389 ff)

8.6.5.3 Active projects

This page displays a chart of the number of the projects which were simultaneously accessed during the last 24 hours.

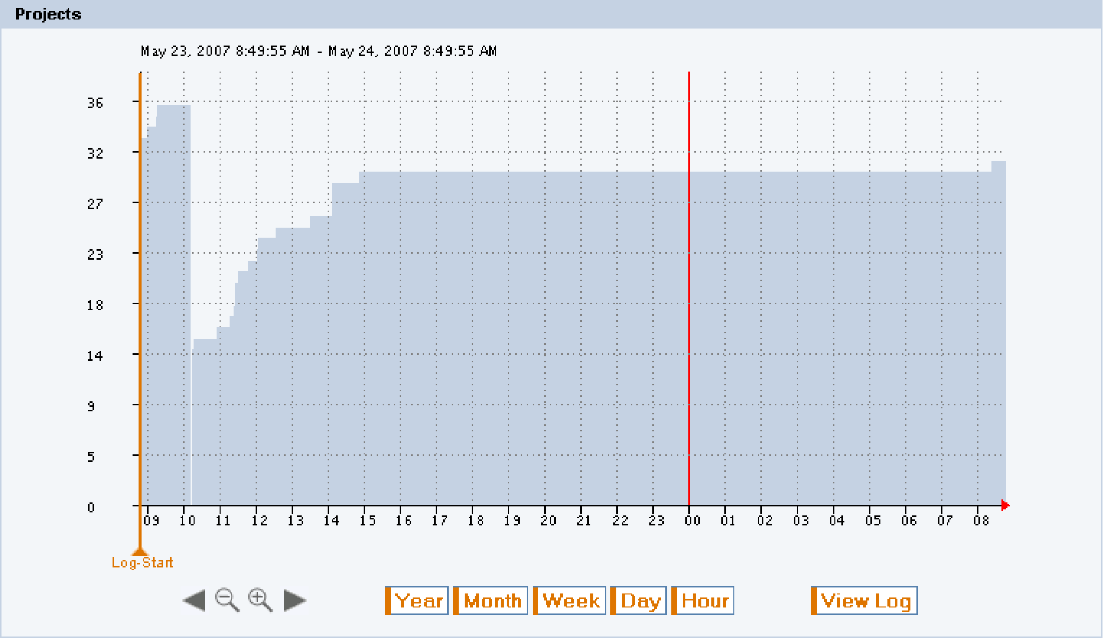


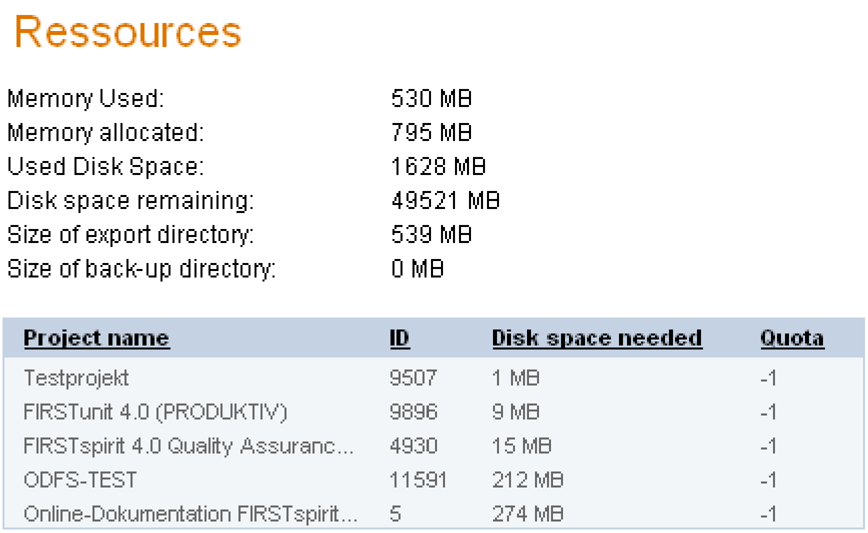
Figure 8-24: Monitoring – Projects

Use the **Year**, **Month**, **Week**, **Day** and **Hour** button to change the display period for the active sessions. Use ⊖ and ⊕ to move one category towards the year and hour button respectively. Depending on the set category, use the arrows to shift the displayed period backwards (◀) or forwards (▶) by one year (month, week…).

Click on View Log to display the server log file for the selected period. (See section 8.3.1.2 page 389 ff)

8.6.5.4 Resources

This page displays an overview of available and already used server memory capacities.



Ressources

Memory Used:	530 MB
Memory allocated:	795 MB
Used Disk Space:	1628 MB
Disk space remaining:	49521 MB
Size of export directory:	539 MB
Size of back-up directory:	0 MB

Project name	ID	Disk space needed	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

Figure 8-25: Monitoring – Resources

The table lists all the projects installed on the server. Besides the project name, the project ID and the required disk memory space are displayed. Furthermore, the maximum disk space available to the project (quota) is displayed (if no limit has been specified, the value -1 is displayed).

This list can be sorted according to all the available columns.

8.6.5.5 Threads

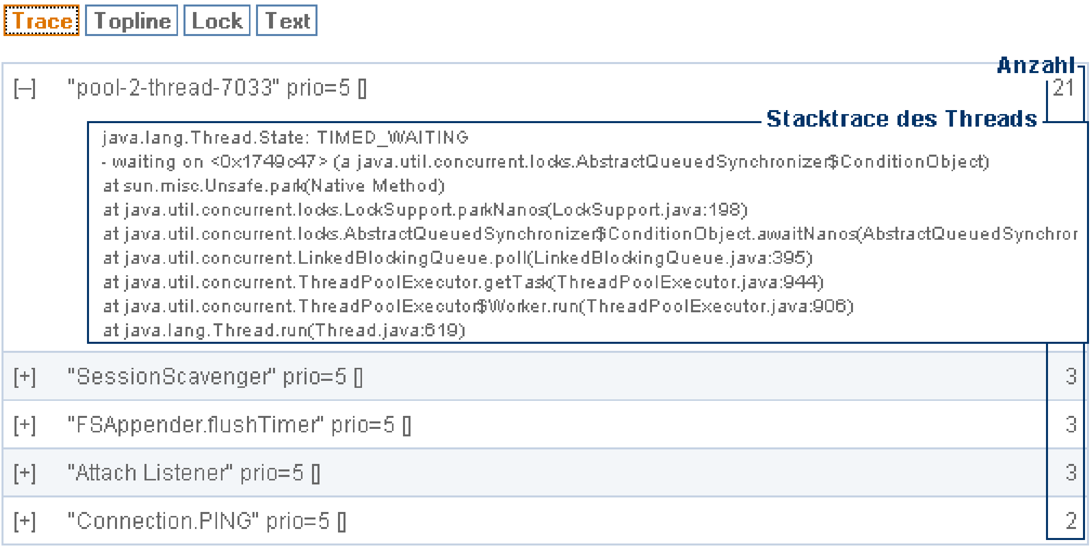
! *This menu level is only available to server administrators.*

This page contains information for monitoring the current system state.

Thread dump Click on the button to display a current thread dump in the following window:

Click on [+] to expand the displayed layer.

The current thread dump can be analysed and displayed in a formatted view via the buttons at the top of the window.



```
[–]   "pool-2-thread-7033" prio=5 []                                   21
      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(AbstractQueuedSynchror
      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)
[+]   "SessionScavenger" prio=5 []                                      3
[+]   "FSAppender.flushTimer" prio=5 []                                 3
[+]   "Attach Listener" prio=5 []                                       3
[+]   "Connection.PING" prio=5 []                                       2
```

Figure 8-26: Threads – View sorted according to "Trace"

Trace Click on the button to group the current threads according to similar stack traces. Stack traces are grouped if they are identical except for the object addresses. The number on the right shows the number of similar traces found. Due to the formatted view it is relatively easy to determine which actions are currently taking place on the server.

Topline Click on the button to group the current threads in the first five lines according to similarities. Similar threads are summarised to one entry in the formatted view. The number on the right shows the number of similar traces found. Due to the formatted view it is relatively easy to determine the current thread position. This view facilitates the tracing of bottlenecks, e.g. if a large number of threads always wait in a specific code position.

Lock Click on the button to check the current threads for locks. If several threads have the status BLOCKED (see “Status” below), they are waiting for the release of an object by another thread. If a large number of objects are continuously waiting and the process speed is slow, it might indicate a bottleneck.

Text Click on the button to display the fully unprocessed thread dump with all the information.

Besides the individual creation of a thread dump, it is possible to create further dumps at certain time intervals. To achieve this, enter the total number of desired thread dumps in the first field and the time intervals in the second input field. Click on “Generate” to start execution.

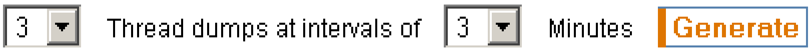


Table: The tabular overview of the page shows the status of the current threads. The thread information displayed here is a snapshot or an extract of the current status.

ID: Each thread has a thread ID via which it can be clearly identified.

Name: Name of the thread.

Status: The threads can have numerous statuses:

- NEW – newly generated thread which has not yet been started.
- RUNNABLE – the thread is being processed or is runnable (waiting for the CPU).
- BLOCKED – the thread is waiting for a lock release.
- WAITING – the thread is waiting for another thread.
- TIMED_WAITING – the thread is waiting for a set time to elapse (e.g. after calling sleep() or wait() with timeout)
- TERMINATED – the thread has been terminated

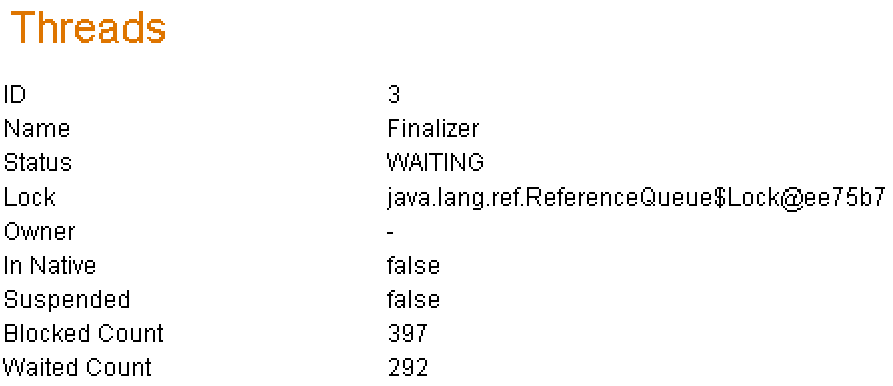
IN (In Native): Provides information on whether the thread executes a native code via the Java Native Interface (JNI) (true) or not (false)

SP (Suspended): Provides information on whether a thread has currently been started (false) or not (true).

BC (Blocked Count): The number provides information on how often a thread has already been in the BLOCKED status during execution.

WC (Waited Count): The number provides information on how often a thread has already been in the WAITING status during execution.

Click on a thread in the table to open a window with detailed thread information. All tabular information is listed in a clear structure and the lock object and the owner are also displayed:



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

Figure 8-27: Threads – Detailed information

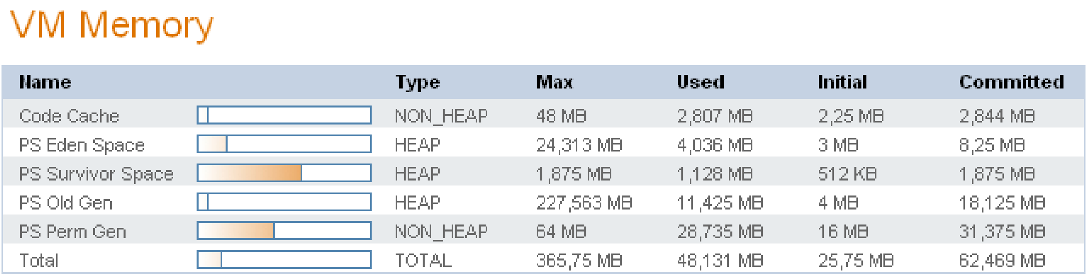
Lock: If a thread is BLOCKED, it is waiting for the release of the stated lock object.

Owner: If a thread is BLOCKED, the name of the thread which currently locks the object is output.

The further outputs are described in the current section (see above).

8.6.5.6 VM memory

This area displays information on the current memory consumption of the Java VM.



VM Memory

Name		Type	Max	Used	Initial	Committed
Code Cache		NON_HEAP	48 MB	2,807 MB	2,25 MB	2,844 MB
PS Eden Space		HEAP	24,313 MB	4,036 MB	3 MB	8,25 MB
PS Survivor Space		HEAP	1,875 MB	1,128 MB	512 KB	1,875 MB
PS Old Gen		HEAP	227,563 MB	11,425 MB	4 MB	18,125 MB
PS Perm Gen		NON_HEAP	64 MB	28,735 MB	16 MB	31,375 MB
Total		TOTAL	365,75 MB	48,131 MB	25,75 MB	62,469 MB

Figure 8-28: Monitoring – Java VM memory (table)

! *This menu level is only available to server administrators.*

Table:

Name: Name of the displayed memory pool.

- **Code Cache:** Memory space used for the internal evaluation, e.g. for compiling.
- **Eden Space:** Memory space in which most objects are initially generated. As soon as the Garbage Collector (GC) cleans up the Eden Space, the surviving objects are transferred to the Survivor Space.
- **Survivor Space:** Memory space to which short-lived objects are shifted from the Eden Space after GC (Garbage Collection).
- **Tenured Gen:** Memory space for long-lived objects, which are shifted here from the Survivor Space after a certain amount of time.
- **Perm Gen:** Memory space for permanently required JVM objects.
- **Total:** No memory space. Provides a total overview of all available memory spaces.

Type: Memory type (HEAP || NON_HEAP).

Max: Maximum possible memory (in bytes) used for storage management.

Used: Currently used memory (in bytes).

Initial: Initially allocated memory (in bytes) during JVM start.

Committed: Memory (in bytes) guaranteed for the JVM.

Besides the tabular overview at the top of the window, a chart is provided for each memory space.

Charts:

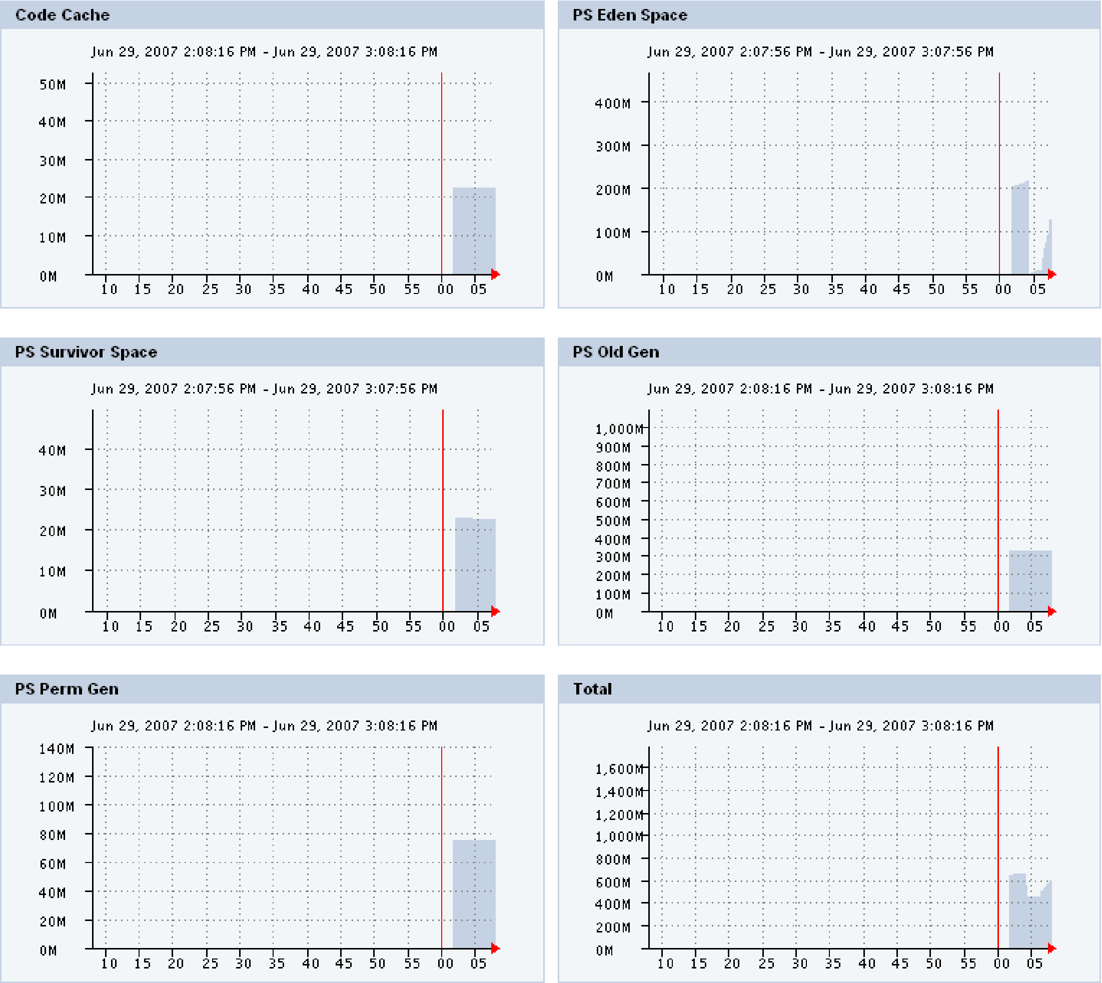


Figure 8-29: Monitoring – Java VM memory (chart)

8.6.5.7 JMX values

The Java Management Extensions (JMX) provides a uniform interface for Java application management. Using the JMX console it is possible to monitor and manage the FirstSpirit Server during runtime (see chapter 9 page 417).

Some information is also provided as a chart via JMX values in the Server Monitoring.

> **!** *This menu level is only available to server administrators.*

This menu level is only available to server administrators.

TasksWaiting: Number of actions which have already been transferred from the respective execution queue to the internal thread pool for execution, but are still waiting (see Figure 9-13) (see section 9.8 page 427).

TasksRunning: Number of actions currently running (for example an indexing job or an event (see section 9.8 page 427).

TasksQueued: If an action is triggered internally, e.g. an indexing, the respective task is at first put into a waiting queue (exception: high priority tasks) (see Figure 9-13). The value displayed here shows the number of currently buffered actions (see section 9.8 page 427).

ExecutionRate: Number of tasks executed during the last 60 seconds.

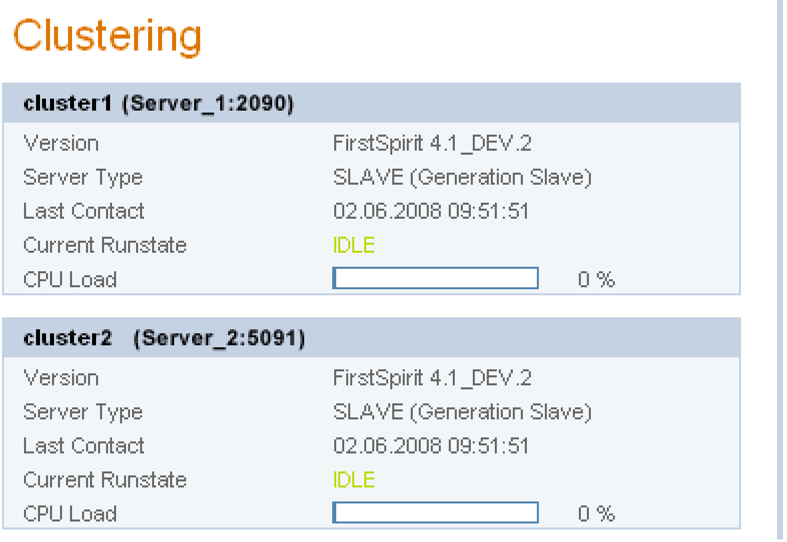
AveragePreviewCount: Number of pages or page references for which a new preview has been calculated during the last 60 seconds (see section 9.11 page 433).

! *The structure and function of the MBeans described below are subject to constant changes. These constant updates can only be reflected in the documentation to a limited degree. Therefore, it might be possible that some figures or descriptions in this documentation differ from the current JMX value view due to small time delays.*

8.6.6 FirstSpirit – Clustering

The “Clustering" menu item can be used to open an overview of the existing cluster nodes for the generation. Initially the overview is empty (if only one FirstSpirit Master server is available). Cluster nodes are not displayed until they exist in the cluster (see chapter 7.3.17 page 248).

Any number of "Generation Slaves" can be displayed for distributed processing of the generation schedules (see Figure 8-30).



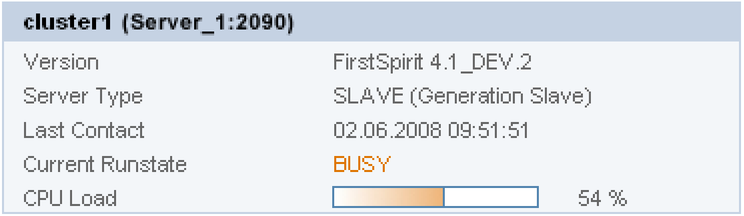
Clustering

cluster1 (Server_1:2090)	
Version	FirstSpirit 4.1_DEV.2
Server Type	SLAVE (Generation Slave)
Last Contact	02.06.2008 09:51:51
Current Runstate	IDLE
CPU Load	0 %

cluster2 (Server_2:5091)	
Version	FirstSpirit 4.1_DEV.2
Server Type	SLAVE (Generation Slave)
Last Contact	02.06.2008 09:51:51
Current Runstate	IDLE
CPU Load	0 %

Figure 8-30: "Clustering" overview

The following information is displayed for each generation server:



cluster1 (Server_1:2090)	
Version	FirstSpirit 4.1_DEV.2
Server Type	SLAVE (Generation Slave)
Last Contact	02.06.2008 09:51:51
Current Runstate	BUSY
CPU Load	54 %

Figure 8-31: Overview of a generation server

Version: Version number of the FirstSpirit server

Server type: The server type of the cluster nodes is given here. For example, the overview displays the type "SLAVE (Generation Slave)" which is responsible for the deployment processes. Several generation servers can of course be used here. If necessary, several deployments can be distributed to different servers in this way.

Last contact: Time of the last server contact (ping).

Current status: Current status of the server:

- IDLE Running with no load
- BUSY The server is currently occupied with working through processes .

Capacity utilisation: Displays the percentage capacity utilisation of the server, for example, during processing of a generation schedule.

9 FirstSpirit JMX Console

The Java Management Extensions (JMX) provide a uniform interface for Java application management. Use the JMX console to monitor and manage the FirstSpirit Server during runtime. While FirstSpirit Server Monitoring primarily aims to manually monitor a FirstSpirit Server, the JMX interface is used for automatic monitoring and is able to integrate itself perfectly into an existing, company-wide monitoring system, if necessary. Interactive use of the JMX Console is also possible.

In contrast to Server Monitoring (see chapter 8 page 376), the information provided by the JMX Console is much finer grained. JMX offers a standardised method for viewing and managing application or JVM resources. All values and operations are provided via managed beans (MBeans). Certain MBeans and the information they provide are presented in the following sections (section 9.3).

The structure and function of the MBeans described below are subject to change. These constant updates can only be reflected in the documentation to a limited degree. Therefore, some figures or descriptions in this documentation may differ from the current JMX Console view due to small time delays.

9.1 JMX Console start

A remote connection to the application to be monitored is required for effective JMX Console utilisation. From JDK 1.5 or higher, a platform function enables a JMX agent start. Set the following system property for the server start to activate this function:

```
-Dcom.sun.management.jmxremote
```

The following parameter is also important:

```
-Dsun.rmi.dgc.client.gcInterval=3600000
```

The interval for the RMI Garbage Collection is set to 1 hour here.

The following parameters should always be set when operating the JMX Console in a production environment:

```
-Dcom.sun.management.jmxremote.authenticate=true
```

and, if applicable, (for encoded transfer):

```
   -Dcom.sun.management.jmxremote.ssl=true
```

A password for the JMX port is assigned via `jmxremote.authenticate` and the SSL authentication is activated via `jmxremote.ssl`.

> ! *If both parameters are deactivated (default setting), JMX port access is unprotected and unauthorised users can terminate the server via the JMX port.*

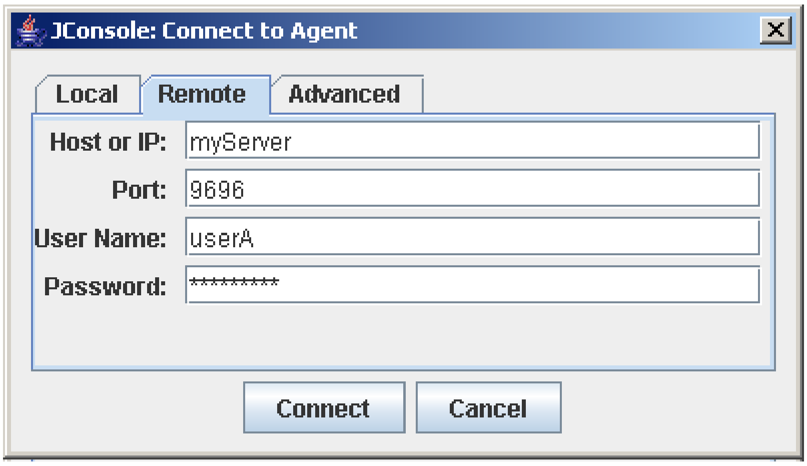
Example configuration:

```
#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
```

Each free port can be entered as a port unless it is blocked by a local firewall.

For further information on JMX port configuration with secured access see: http://java.sun.com/j2se/1.5.0/docs/guide/management/agent.html

The JConsole is started via the command line call “jconsole” for application connection:



Host or IP: myServer
Port: 9696
User Name: userA
Password: *********

Figure 9-1: JConsole start (remote)

Remote access is started via the “Remote” tab. Besides the host name, the JMX port of the target application can also be entered here. Once the respective parameters have been configured, authentication via “User name” and “Password” should occur. Click on “Connect” to start the JConsole. (Presentation of the JMX Console depends on the JConsole being used.)

For a detailed description of the GUI see: http://java.sun.com.

The following sections describe how to monitor the FirstSpirit Server with the information provided by MBeans (section 9.3 ff.). All the information is in the "MBean" tab (see Figure 9-2).

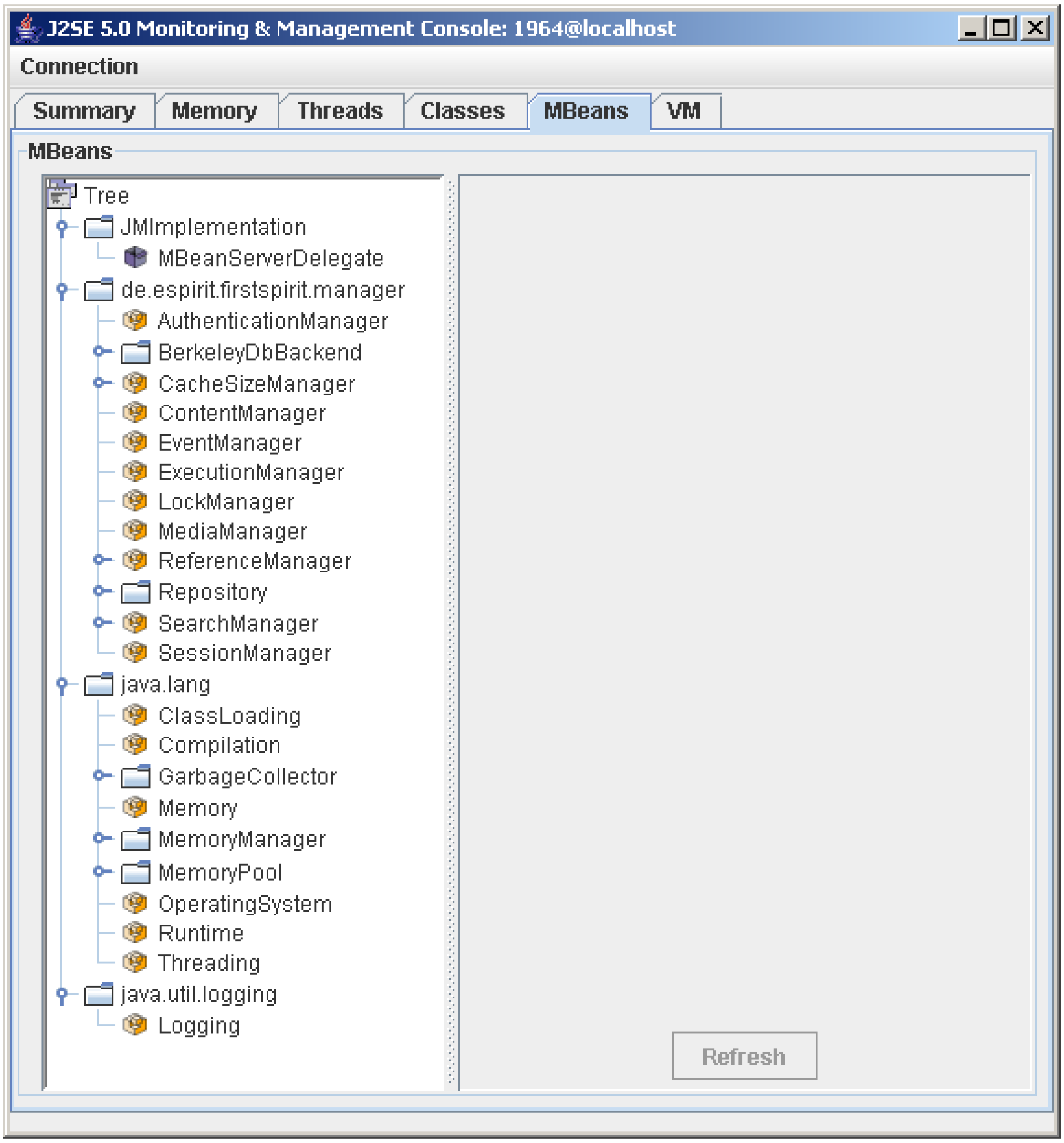


Figure 9-2: "MBean" tab of the JMX Console

9.2 Tabs

Besides the information in the “Attributes” tab, the MBeans provided by FirstSpirit also offer editing options in the “Operation” tab (see section 9.4.1 page 421 and section 9.4.2 page 422).

Figure 9-3: JMX Console tabs

Attributes: Information and statistical values which can be presented in tables (default display) and charts (double-click on the respective value to toggle) (see Click on “Discard chart” to close the chart.

Operations: Operations which can be called on certain managers, e.g. to show further statistical values (see Figure 9-7) or to trigger specific actions (see Figure 9-6). Click on the respective button to start the operations.

Info: General manager information (name and class of the manager for manager implementation).

The respective tabs are only displayed in managers for which the information or operations are relevant.

9.3 AuthenticationManager

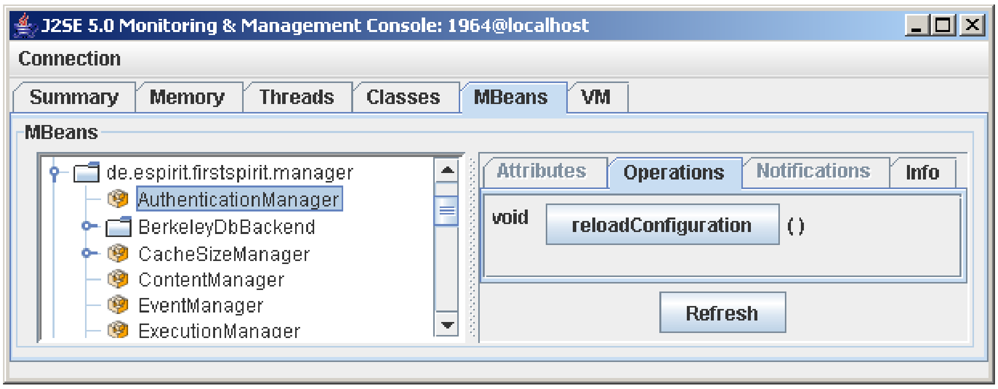


Figure 9-4: FirstSpirit JMX Console (AuthenticationManager presentation)

The AuthenticationManager manages all login and authentication options via the configuration file “fs-jaas.conf” (see section 4.3.4 page 72). The login process settings can be edited via the Server and Project Configuration (see section 7.3.13

page 231). In this case, the configuration file is rewritten and automatically reloaded.

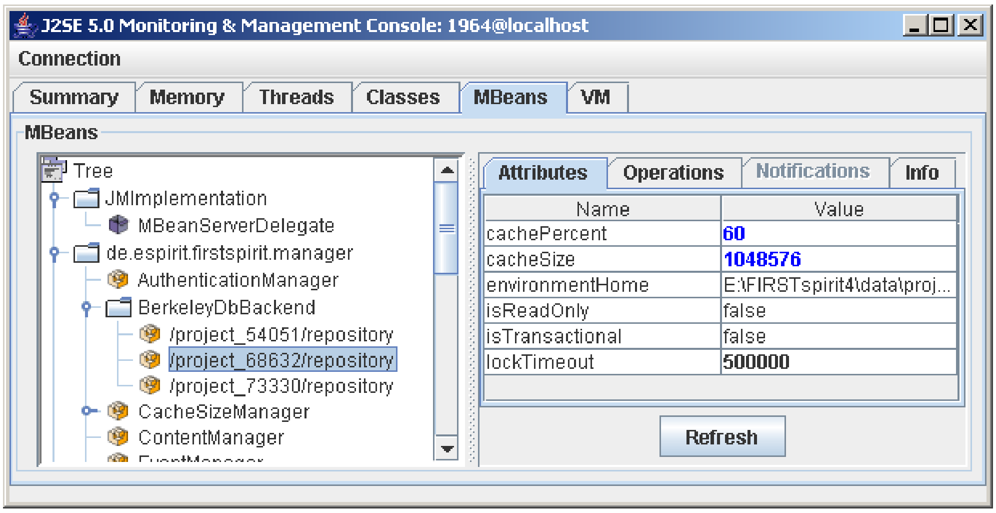
The Server and Project Configuration is recommended for carrying out changes!

9.4 BackendManager

This entry displays information on the connected backend systems. Only the Berkely DB is currently supported as a backend system; therefore, only this entry is displayed in the JMX Console. The BackendManager is a dynamic manager just like the Repository (see section 9.13) or CacheSizeManager (see section 9.5). This means that the entry is only loaded into the JMX Console if the respective project has been opened by at least one user. The information is then dynamically updated during runtime. Each project has its own MBean displayed below the "BerkeleyDBBackend" folder.

9.4.1 Attributes

The BackendManager contains the following attributes for each project:



Name	Value
cachePercent	60
cacheSize	1048576
environmentHome	E:\FIRSTspirit4\data\proj...
isReadOnly	false
isTransactional	false
lockTimeout	500000

Figure 9-5: Backend information on the JMX Console (Attributes)

CachePercent: States the percentage of cache memory consumption for the backend in relation to the main cache memory.

CacheSize: States the cache memory consumption for the backend in bytes.

All further attributes are described in the documentation of the respective backend system (see BerkeleyDB[33] documentation).

9.4.2 Operation

Besides the attributes, it is also possible to select the "Operation" tab. Use the tab to execute operations and to request statistical information (for further information see the BerkeleyDB documentation).

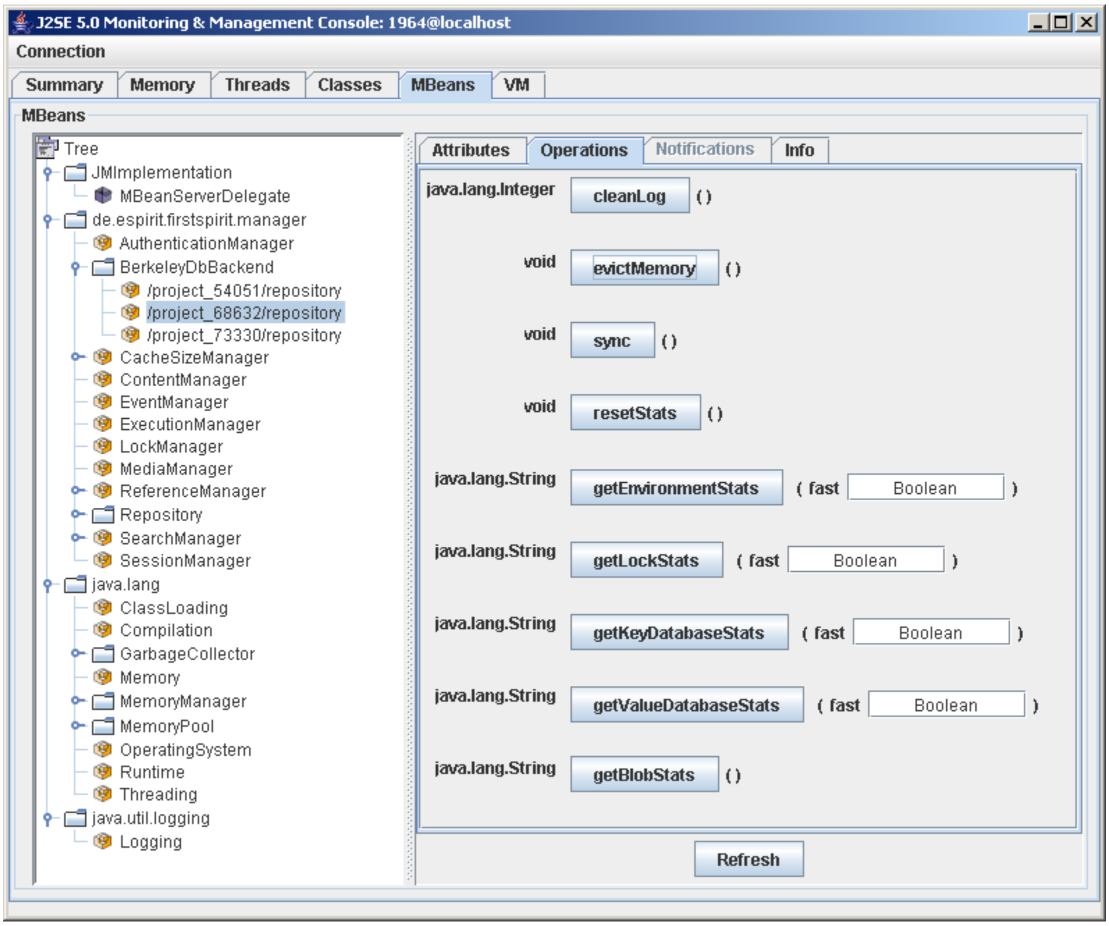


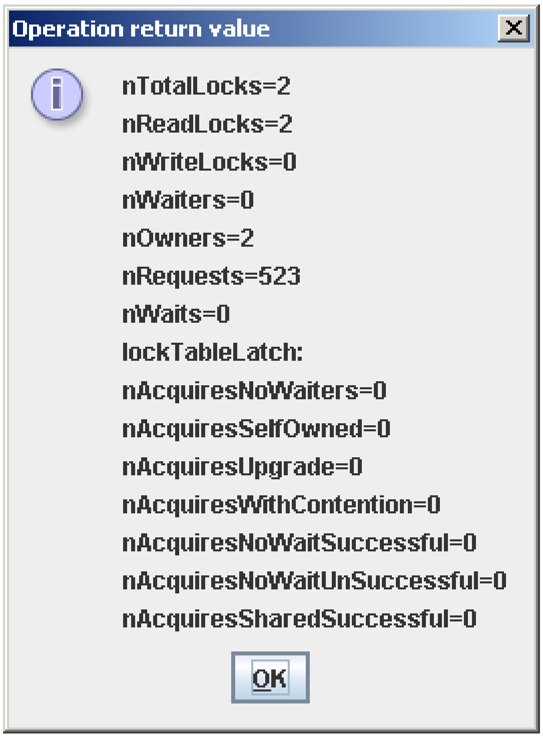
Figure 9-6: Backend information on the JMX Console (Operations)

Operations which can be executed by clicking on the respective field, e.g. on `cleanLog()`, are displayed at the top of the dialogue.

`sync()` is a frequently required operation in this area (see Figure 9-6). When executing `sync()`, all data which has been changed but not yet saved is written onto the hard disk.

[33] http://www.oracle.com/database/berkeley-db/index.html

Besides the operations, there are also statistics which can be executed by clicking on the respective field, e.g. on `getLockStats(...)` (see Figure 9-7):



```
nTotalLocks=2
nReadLocks=2
nWriteLocks=0
nWaiters=0
nOwners=2
nRequests=523
nWaits=0
lockTableLatch:
nAcquiresNoWaiters=0
nAcquiresSelfOwned=0
nAcquiresUpgrade=0
nAcquiresWithContention=0
nAcquiresNoWaitSuccessful=0
nAcquiresNoWaitUnSuccessful=0
nAcquiresSharedSuccessful=0
```

Figure 9-7: Statistical information on the JMX Console (operations)

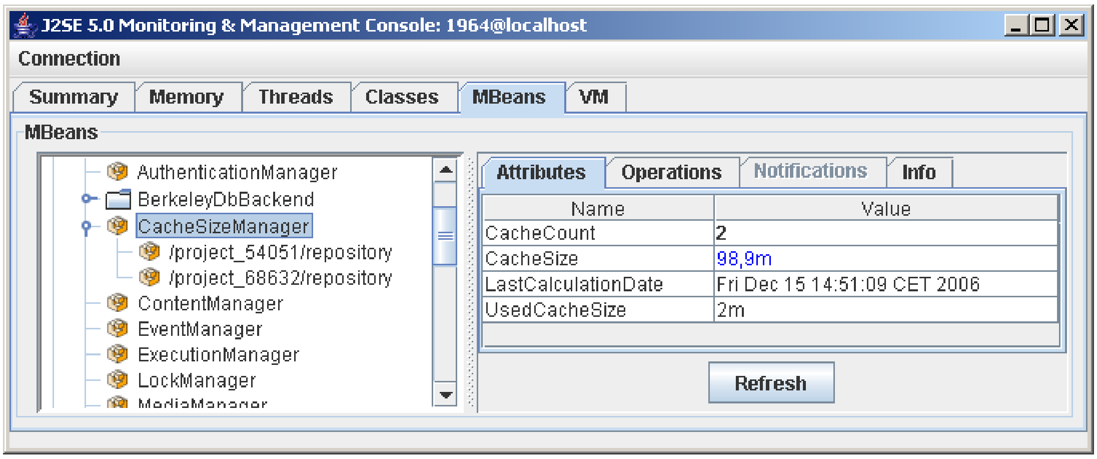
The call can take some time, especially for large amounts of data.

9.5 CacheSizeManager

The CacheSizeManager distributes the available main cache memory to the various caches. This entry presents information on the total cache memory (CacheSizeManager - see Figure 9-8) and on the individual project caches (MBeans - see Figure 9-10). The CacheSizeManager is a dynamic manager just like the RepositoryManager (see section 9.13). This means that the entry is only loaded into the JMX Console if the respective project has been opened by at least one user. The information is then dynamically updated at runtime ("Refresh"). Each project has its own MBean which is displayed below the "CacheSizeManager".

9.5.1 Attributes (all)

The following attributes are available on the CacheSizeManager layer:



Name	Value
CacheCount	2
CacheSize	98,9m
LastCalculationDate	Fri Dec 15 14:51:09 CET 2006
UsedCacheSize	2m

Figure 9-8: Total cache information on the JMX Console (Attributes)

CacheCount: Total number of the currently registered caches (all projects).

CacheSize: Absolute cache size. The value shown here results from the values set for the parameters `CACHE_SIZE` (absolute specification) or `CACHE_PERCENT` (percentage specification) in the configuration file `fs-server.conf` (see section 4.3.1.13 page 55).

LastCalculationDate: Time of the last cache memory redistribution between the caches.

UsedCacheSize: States the current cache size actually being used by all caches.

9.5.2 Operation

One operation is available on the CacheSizeManager layer:

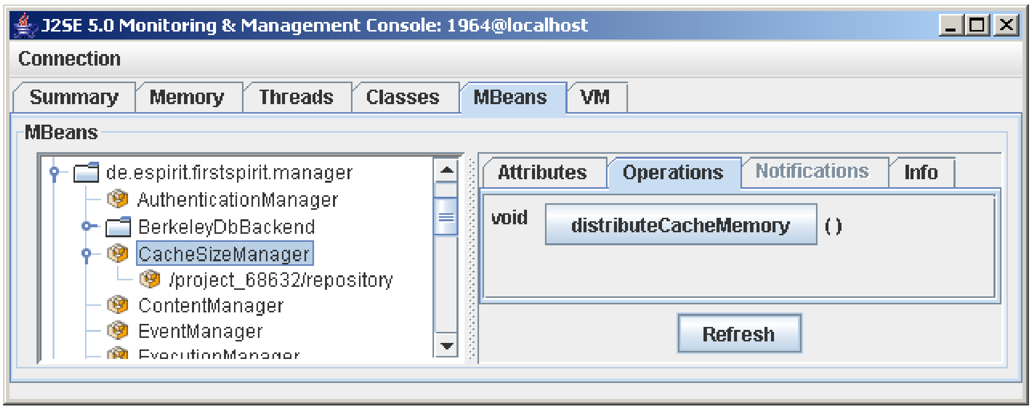
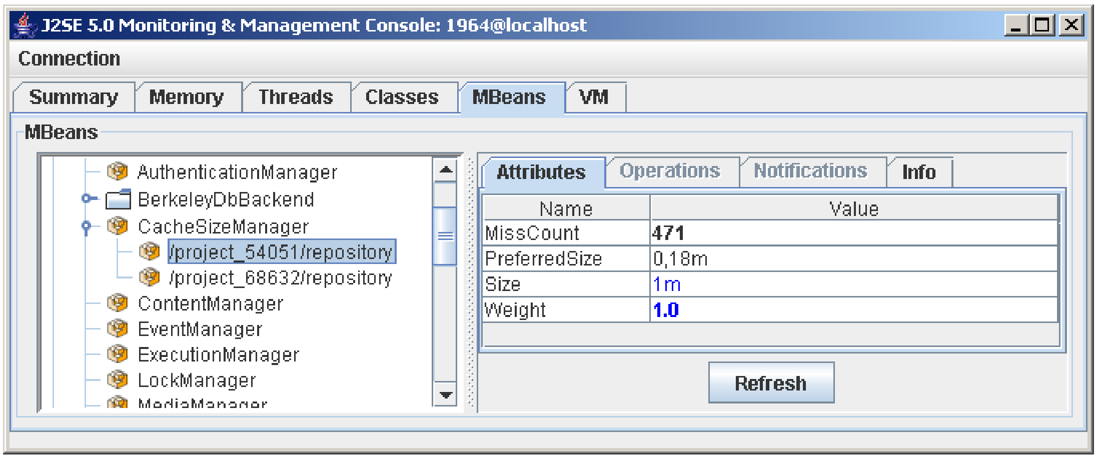


Figure 9-9: Total cache information on the JMX Console (Operations)

Click on `distributeCacheMemory()` to redistribute the available cache memory. After execution, the value saved under “LastCalculationDate” is adapted (see section 9.5.1).

9.5.3 Attributes (project-related)

Besides the total information for the cache size, each project has its own CacheSizeMBean with the following attributes:



Name	Value
MissCount	471
PreferredSize	0,18m
Size	1m
Weight	1.0

Figure 9-10: Cache information per project via the JMX Console (Attributes)

MissCount: The counter records the number of requests which could not be directly handled from the cache. These requests have to be redirected to the backend and handled by accessing the hard disk. A high or strongly increasing value indicates poor cache utilisation, resulting in long response times.

PreferredSize: The memory cache size preferred by the project cache is displayed here. This cache memory size does not always correspond to the actually allocated value (size) whose allocation also depends on the “PreferredSize”.

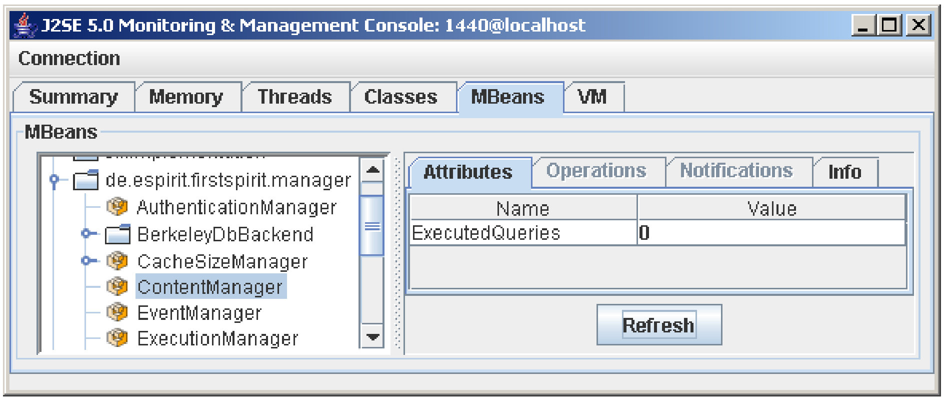
Size: The CacheSizeManager allocates 1 MB to each project cache by default. If the “Preferred Size” is higher, it might be possible to provide more cache memory. How much cache memory a project cache receives depends on the available total cache memory size (global CacheSize), the “PreferredSize” of all the projects and the weight of the individual projects.

Weight: Weight is a parameter defined via the Server and Project Configuration in the project properties (see section 7.4.2). The weight of a project determines how much cache memory a project cache receives from the CacheSizeManager, i.e. more for a higher weight and less for a lower weight.

9.6 ContentManager

The ContentManager contains information on FirstSpirit database connections and accesses.

The following attributes are available on the ContentManager layer:



Name	Value
ExecutedQueries	0

Figure 9-11: FirstSpirit JMX Console (ContentManager presentation)

ExecutedQueries: Number of executed database SQL statements during the last minute.

9.7 EventManager

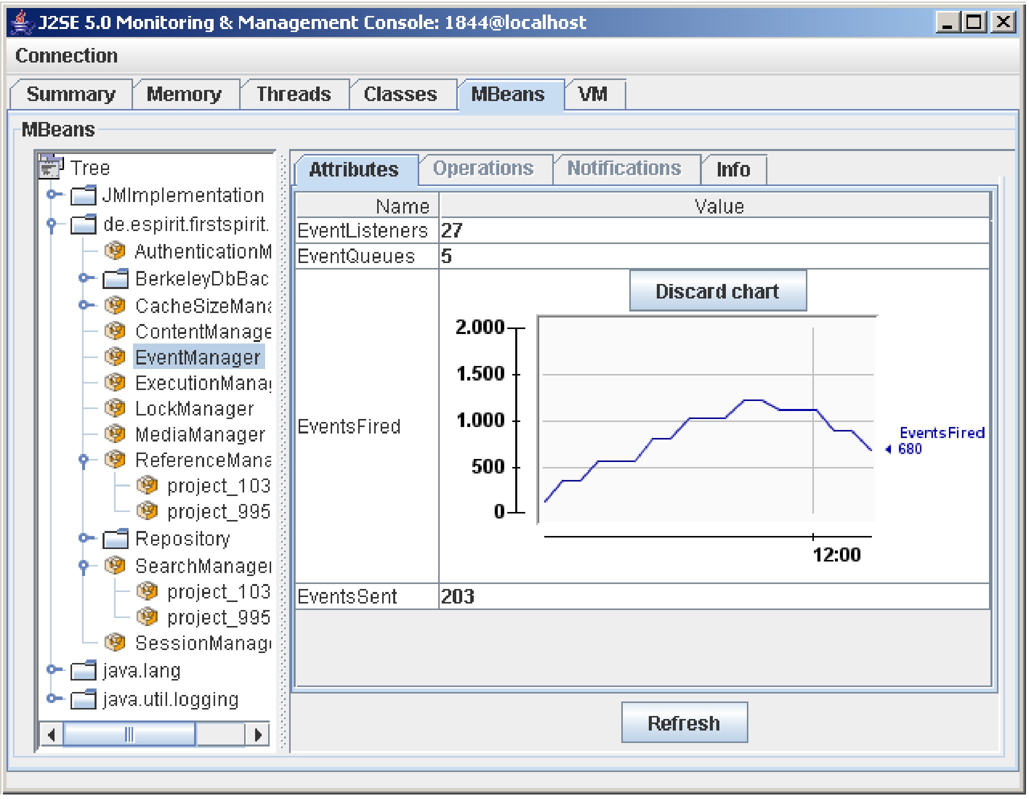
The EventManager provides information on cancelled and sent events. Many actions triggered in the FirstSpirit environment lead to specific events. For example, project loading results in the respective references being loaded via the ReferenceManager. The EventManager, therefore, links different managers (e.g. ReferenceManager and SearchManager).

Further actions which trigger events are:

- Create, edit, save and delete elements
- Load, change, deactivate and delete a project
- Create, edit and delete database contents

It is, therefore, possible to draw conclusions about the processing speed and load on the FirstSpirit Server, particularly in multi-user environments, via the EventManager.

The following attributes are available on the EventManager layer:



Name	Value
EventListeners	27
EventQueues	5
EventsFired	
EventsSent	203

Figure 9-12: FirstSpirit JMX Console (EventManager presentation)

EventListeners: Number of server and client-sided EventListeners currently registered on the server.

EventQueues: EventQueues are the server-sided equivalent to the client-sided EventListener. Since the server cannot send the events directly to the client, they are buffered until being collected. This value indirectly provides information on the number of clients connected to the server.

EventsFired: Triggered events during the last 60 seconds.

EventsSent: Actually sent events during the last 60 seconds.

9.8 ExecutionManager

In FirstSpirit, a series of actions is simultaneously executed in the background. These include, e.g., reference chart update, client call processing or document indexing. The execution of these actions occurs via the ExecutionManager.

9.8.1 Thread queue classification

The ExecutionManager manages a number of differently classed thread queues. The various tasks (e.g. client calls) are collected in a thread queue until being executed.

Thread queues can be configured depending on their classification:

`ThreadQueue.SERIAL:` Tasks which cannot be executed simultaneously are placed in the queues classified as `SERIAL`. Since only one task can be executed at a time, the queue does not have to be limited via the configuration.

`ThreadQueue.LOW:` Memory or computing-intensive tasks of which only a small number is to be executed simultaneously are put in the queues classified as `LOW` (default value 2). An example of this is the indexing of documents. The number of tasks which can be executed simultaneously can be defined via parameter `maxRunning` (see section 4.3.1.5 page 41).

`ThreadQueue.DEFAULT:` `DEFAULT` is the default classification of a queue. All the tasks not allocated to one of the other classified queues are placed in the queue classified as `DEFAULT`. The number of tasks which can be executed simultaneously can be defined via parameter `maxRunning` (see section 4.3.1.5 page 41).

`ThreadQueue.HIGH:` The queue classified as `HIGH` includes all the tasks to be executed directly, for example, high priority client calls (e.g. ping). Configuration is not required here.

`ThreadQueue.BOUNDED:` Only server calls of the clients are placed in the queue classified as `BOUNDED`. This queue can be limited in two ways. The number of tasks and the maximum capacity of the queue can be limited (see section 4.3.1.5 page 41). If the maximum capacity of the queue has been reached, further client calls are rejected by the server before a new attempt is started.

9.8.2 Processing in the ExecutionManager

The following flowchart illustrates task execution in the ExecutionManager. The values described here are available in the JMX Console (see next page) and partially in the FirstSpirit Server Monitoring (see section 8.6.5.7 page 414):

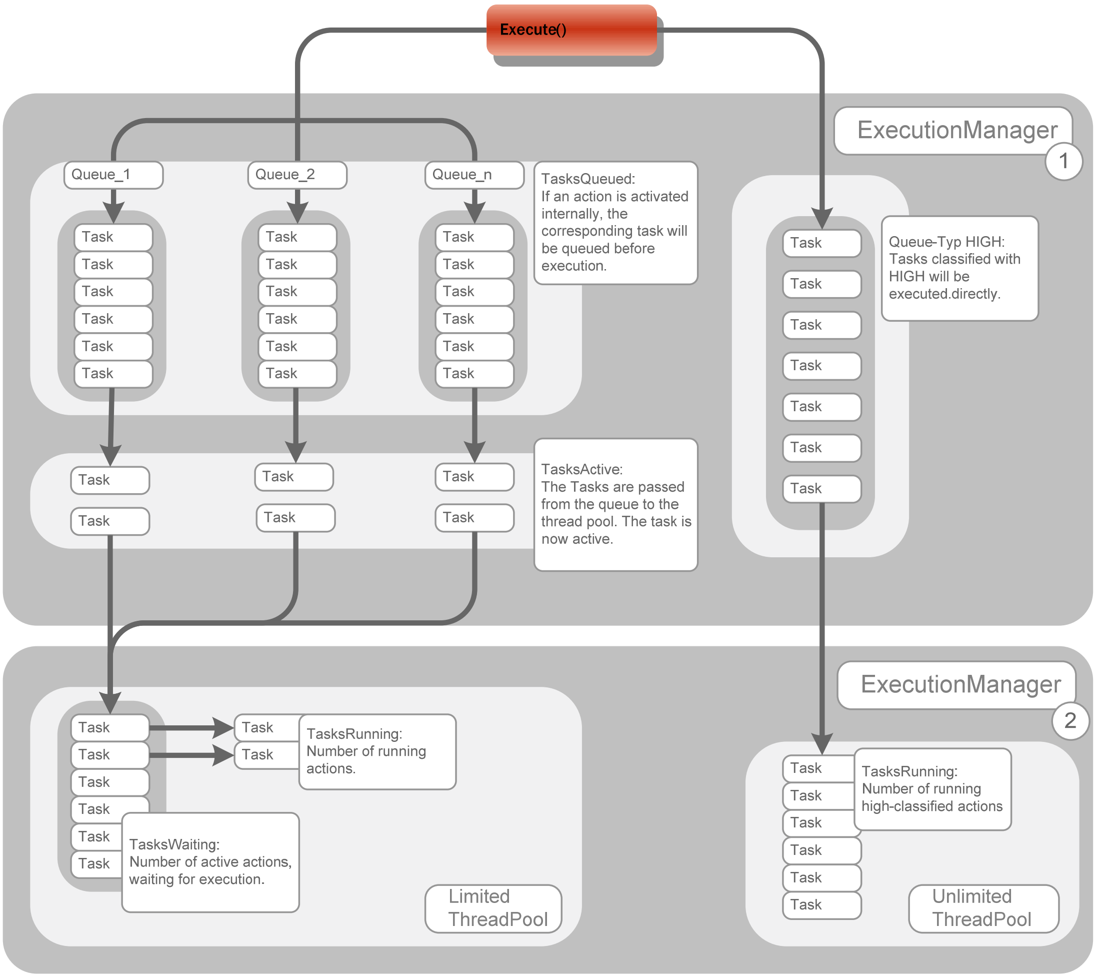
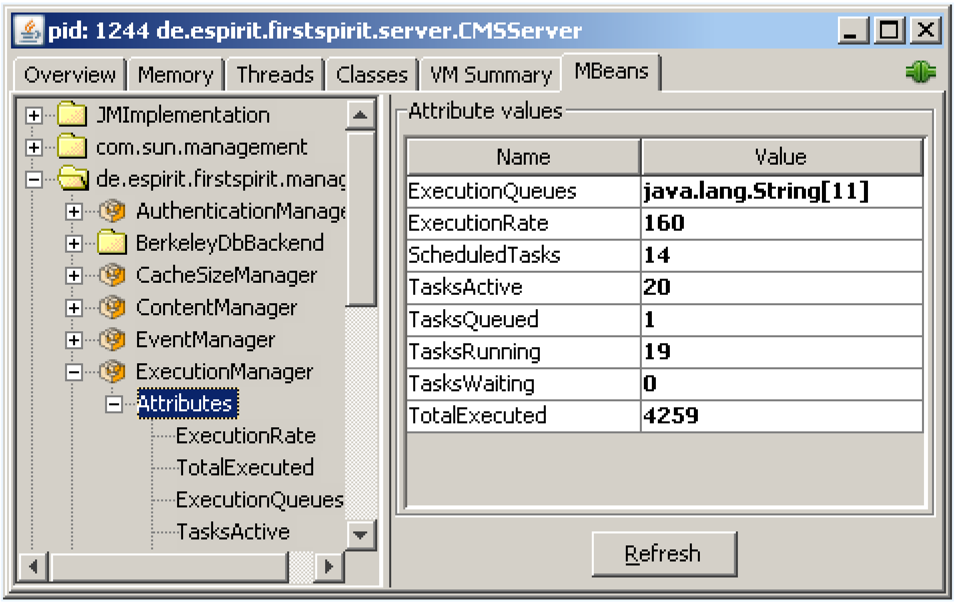

Execute()
ExecutionManager
1
Queue_1
Queue_2
Queue_n
Task
TasksQueued:
If an action is activated internally, the corresponding task will be queued before execution.
Queue-Typ HIGH:
Tasks classified with HIGH will be executed.directly.
TasksActive:
The Tasks are passed from the queue to the thread pool. The task is now active.
ExecutionManager
2
TasksRunning:
Number of running actions.
TasksWaiting:
Number of active actions, waiting for execution.
Limited ThreadPool
TasksRunning:
Number of running high-classified actions
Unlimited ThreadPool

Figure 9-13: Task execution via the ExecutionManager

9.8.3 Attributes

The following attributes are available on the ExcecutionManager layer:



Name	Value
ExecutionQueues	java.lang.String[11]
ExecutionRate	160
ScheduledTasks	14
TasksActive	20
TasksQueued	1
TasksRunning	19
TasksWaiting	0
TotalExecuted	4259

Figure 9-14: FirstSpirit JMX Console (ExecutionManager presentation)

ExecutionRate: Number of tasks executed during the last 60 seconds.

ScheduledTasks: In addition to the events triggered by user actions, there are also regular, internal actions. For example, it is checked every minute whether a session is still valid or has already expired. The value states the number of regular actions.

TasksActive A buffered, internal action (see **"**TasksQueued") is transferred from the queue to the second processing layer of the ExecutionManager (see Figure 9-13). The respective task is then active. However, is not executed, but transferred to the (limited) ThreadPool. The number of possible active tasks transferred from the queue to the ThreadPool can be limited via parameter `maxRunning` (see section 4.3.1.5 page 41).

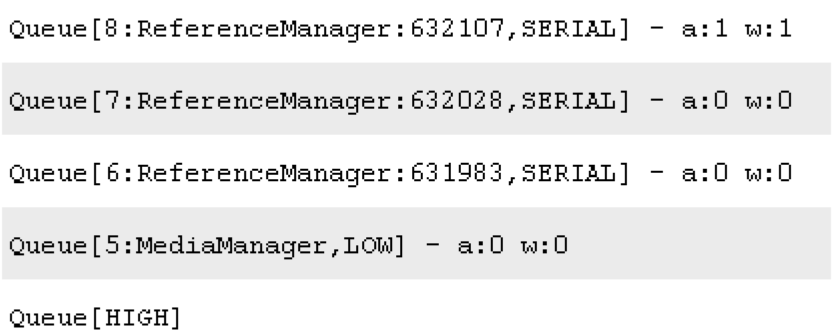
TasksQueued: If an action is triggered internally, e.g. indexing, the respective task is buffered first. The value displayed here shows the number of currently buffered actions (see Figure 9-13).

TasksRunning: The number of currently running actions (e.g. an indexing job or an event) in the limited and unlimited ThreadPool (see Figure 9-13). The number of simultaneously executed tasks (via the limited ThreadPool) can be limited via parameter `maxSize` (see section 4.3.1.4 page 40).

TasksWaiting: The number of actions which have already been transferred from the respective queue, but are still waiting for execution in the ThreadPool (see Figure 9-13).

TotalExecuted: The total number of executed actions.

ExecutionQueues: ExecutionQueues provide an overview of all the queues of the various managers. Double-click on the entry or use the navigation on the left to display the current queues (see Figure 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]
```

Figure 9-15: ExecutionManager – Queues

Each entry contains the following information:

Name of the queue: The name consists of (see Figure 9-15):

- The queue ID
- The name of the manager using the queue
- The project ID. (This value is optional. Some managers, e.g. the ReferenceManager, generate one queue per project)
- The queue class. Possible classes are `LOW`, `HIGH`, `SERIAL`, `BOUNDED`, `DEFAULT` (see section 9.8.1 page 427). Some queue classes can be parameterised via the configuration file `fs-server.conf` (see section 0, page 40 ff).

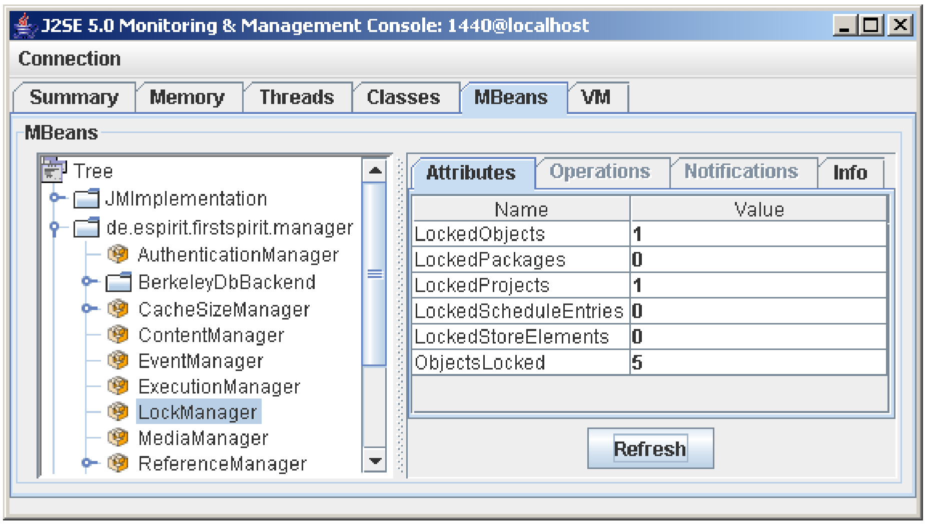
Number of active threads: The number of active threads `[a: number]` transferred from the queue to the ThreadPool (see "TasksActive").

Number of waiting threads: The number of buffered threads `[w: number]` transferred from the queue to the ThreadPool (see "TasksQueued").

9.9 LockManager

When carrying out specific actions in FirstSpirit, e.g. changing project properties or editing a page, objects have to be locked to prevent access by another user. The respective user automatically (when changing project properties) or manually (when editing a page) locks this object. The LockManager manages all the currently locked objects and provides information on the various lock types.

The following attributes are available on the LockManager layer:



Name	Value
LockedObjects	1
LockedPackages	0
LockedProjects	1
LockedScheduleEntries	0
LockedStoreElements	0
ObjectsLocked	5

Figure 9-16: FirstSpirit JMX Console (LockManager presentation)

LockedObjects: The number of locked objects (projects, elements, packages, schedule entries) being edited.

LockedPackages: The number of locked packages being edited. This value is only filled in if the licence-dependent function PackagePool is used.

LockedProjects: The number of locked projects being edited (see section 7.3.16 page 243).

LockedScheduleEntries: The number of locked schedule entries being edited (see section 7.5.4 page 318).

LockedStoreElements: The number of locked elements from FirstSpirit stores (e.g. Page-Store) being edited.

ObjectsLocked: The number of objects (projects, elements, packages, schedule entries) locked by a user during the last 60 seconds.

9.10 MediaManager

FirstSpirit provides thumbnails in the Media-Store. The MediaManager manages the information for presenting and calculating these thumbnails when creating a new medium and is also responsible for calculating the resolutions, if these are requested.

The following attributes are available on the MediaManager layer:

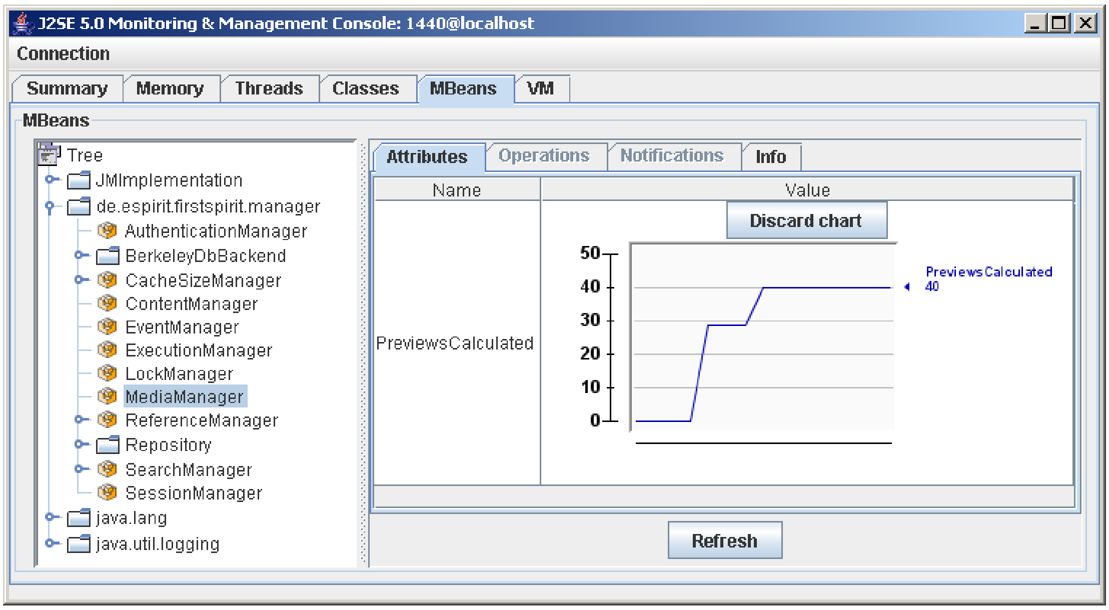


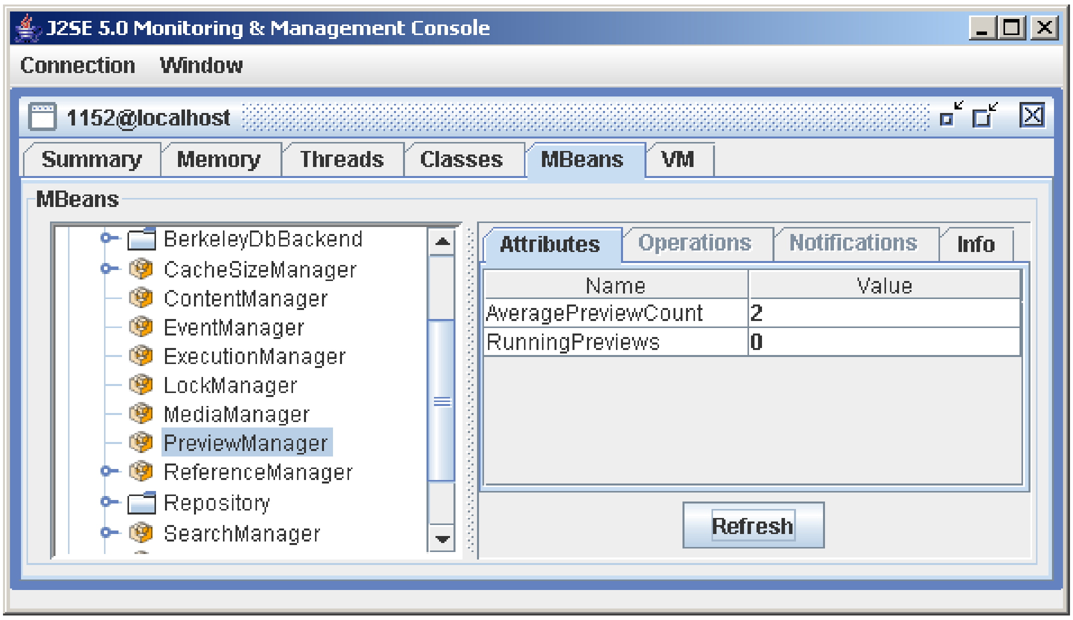
Figure 9-17: FirstSpirit JMX Console (MediaManager presentation)

PreviewsCalculated: Media for which a thumbnail has been calculated during the last 60 seconds.

9.11 PreviewManager

A preview of the currently edited contents can always be requested via FirstSpirit. A preview can be requested in the Page-Store on the page layer or in the Site-Store on the page reference layer. The PreviewManager manages all currently requested previews and provides the respective information.

The following attributes are available on the PreviewManager layer:



Name	Value
AveragePreviewCount	2
RunningPreviews	0

AveragePreviewCount: The number of pages or page references calculated during the last 60 seconds for a new preview.

RunningPreviews: The number of previews currently requested for calculation.

! *A preview request is a very computing time intensive action. Therefore, FirstSpirit offers ways to shift the preview calculation to superordinated systems. In this case, the PreviewManager cannot evaluate preview information.*

9.12 ReferenceManager

The ReferenceManager is responsible for calculating and providing the references.

! *Recalculation and conversion of references: A more performing data format is used for reference from version 4.2R4. Existing references are automatically converted and recalculated when upgrading to 4.2R4. This is executed when projects are opened for the first time on a FirstSpirit Server which has updated to version 4.2R4. The recalculation of references proceeds in an asynchronous manner. I.e. server and projects can be already used while the recalculation of references is still running. The conversion of the data format of the references is effected during the recalculation of references.*

The following attributes are available on the ReferenceManager layer:

ActiveCalculated: Current number of project references which have been already recalculated on the FirstSpirit Server (available from FirstSpirit Version 4.2).

ActiveDuration: Current duration of the presently running recalculation of project references on the FirstSpirit Server (available from FirstSpirit Version 4.2). This information is also displayed in the area "Overview – Activities" in the Server Monitoring (see Chapter 8.1.2 page 378).

ActiveQueued: Current number of the project references which are still to be calculated on the FirstSpirit Server (available from FirstSpirit Version 4.2).

ActiveRecalculations: Current number of project references which are recalculated at this moment (available from FirstSpirit Version 4.2).

ActiveRepaired: Current number of project reference which have been repaired on the FirstSpirit Server (available from FirstSpirit Version 4.2 This information is also displayed in the area "Overview – Activities" in the Server Monitoring (see Chapter 8.1.2 page 378).

CalculationRate: Number of references which have been recalculated during the last 60 seconds (references / minute) (available from FirstSpirit Version 4.0).

On the level of the ReferenceManager the following operations are available:

updateProjectReferences: References which are existing in an old version of the reference graph are automatically recalculated and converted after an upgrade to FirstSpirit Version 4.2R4 with the first access to a project. If a manual actualisation is desired, this can be started via the operation updateProjectReferences. This operation starts the recalculation of all obsolete project references on the FirstSpirit Server (available from FirstSpirit Version 4.2.433).

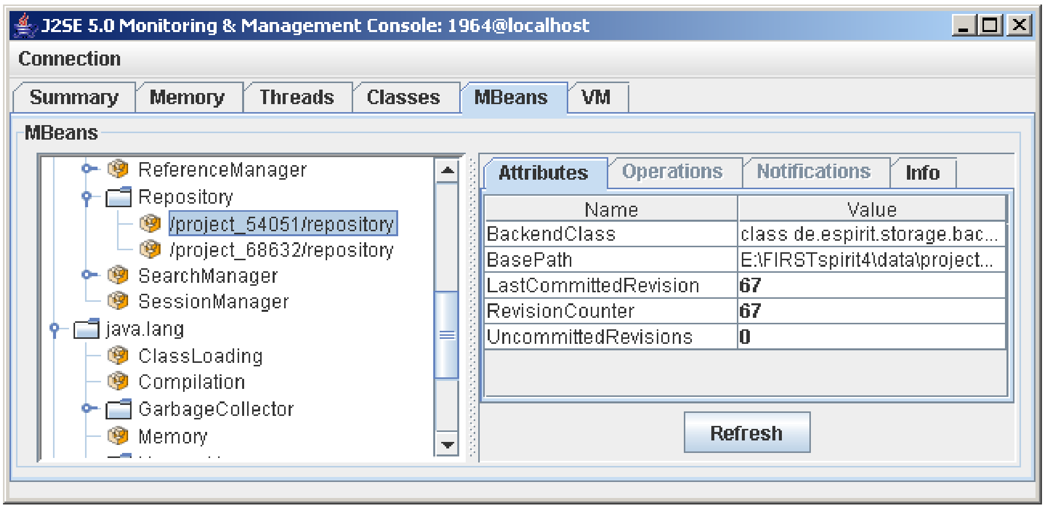
startRecalculation: This operation starts recalculation of the project references in which the most up to date existing revision for a certain project. The JConsole can be used here to transfer the project ID of the project for which recalculation of the references is to be started (available in FirstSpirit-Version 4.2.433 and higher).

The recalculation of the project references can be watched in the area "Overview – Activities" in the Server Monitoring (see Chapter 8.1.2 page 378).

9.13 RepositoryManager

FirstSpirit uses repositories for archiving and versioning project data. Each project has a repository and each repository has an MBean in the “Repository” folder of the JMX Console. The RepositoryManager is a dynamic manager just like the BackendManager (see section 9.4) or the CacheSizeManager (see section 9.5). This means that the entry is only loaded into the JMX Console if the respective project has been opened by at least one user. The information is then dynamically updated at runtime (“Refresh”).

The RepositoryManager contains the following attributes for each project:



Name	Value
BackendClass	class de.espirit.storage.bac...
BasePath	E:\FIRSTspirit4\data\project...
LastCommittedRevision	67
RevisionCounter	67
UncommittedRevisions	0

Figure 9-18: Repository information on the JMX Console (Attributes)

BackendClass: Class which implements the respective repository (only the BerkeleyDB is currently supported).

BasePath: Complete path to the project repository.

LastCommitedRevision: The number of the last revision released for this project.

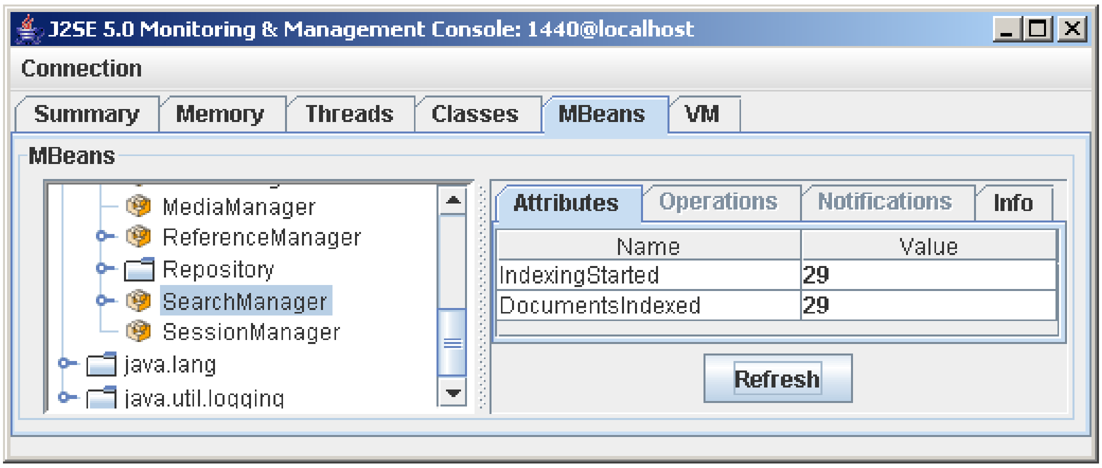
RevisionCounter: The total number of revisions for this project.

UncommitedRevisions: The total number of revisions being edited (i.e. not yet released).

9.14 SearchManager

The SearchManager is responsible for indexing and searching for elements in the FirstSpirit stores (e.g. page, sections, media files). The elements are written into the search index when creating, editing and deleting and subsequently deleted again.

The following attributes are available on the SearchManager layer:



Name	Value
IndexingStarted	29
DocumentsIndexed	29

Figure 9-19: FirstSpirit JMX Console (SearchManager presentation)

IndexingStarted: The number of elements created, changed or deleted during the last 60 seconds and which have to be indexed via the SearchManager. These actions are buffered and processed with low priority.

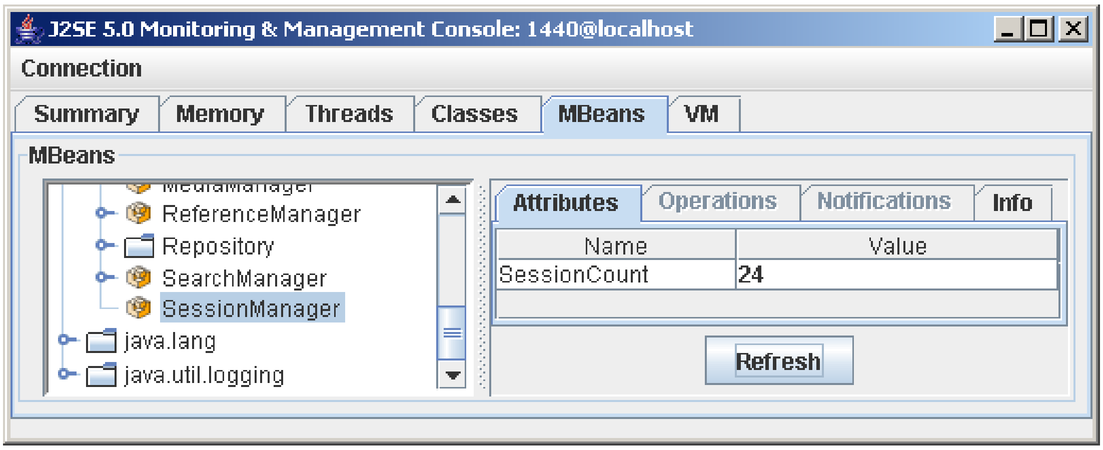
DocumentsIndexed: The number of elements indexed during the last 60 seconds (indexing process completed).

! *Full text indexing of the Media Store: With Version 4.0, documents from the FirstSpirit Media Store are indexed to enable full text search. For technical reasons, all document types are not supported. (Complete) indexing is therefore not possible in all cases.*

9.15 SessionManager

The SessionManager is responsible for session management on the server. A new session is created on the server during user login. The SessionManager checks at regular intervals whether a session is still valid or has already expired (see section 9.8 page 427).

The following attributes are available on the SessionManager layer:



Name	Value
SessionCount	24

Figure 9-20: FirstSpirit JMX Console (SessionManager presentation)

SessionCount: The number of currently valid sessions, during the last 60 seconds. Temporary sessions, i.e. sessions created by the server, are not displayed.

10 Secure deployment via rsync and ssh

The combined application of the external service programs `rsync`[34] and `ssh`[35] is recommended for a deployment via unsecure Internet connections or networks with low bandwidth.

`ssh` provides the encrypted connection between FirstSpirit Server and web server and `rsync` reduces the amount of transferred data relating to changes to the previous deployment.

The rsync and ssh client as well as a generated ssh key pair (private and public key) are required for utilisation of this deployment method on the system where the FirstSpirit Server is installed. An ssh server and the rsync client are required on the web server system. Furthermore, the public ssh key should also be copied onto this system.

The followings sections describe configuration according to the operating system. The user account “web” on the web server with the host name `www.mydomain.net` and the document directory `/var/www` or `c:\www`. are used as an example.

10.1 Web server under Unix

The ssh server has usually been installed. If not, install and activate it via the package system of the operating system. Only the client has to be installed for `rsync` (not `rsyncd`).

At first, a “web” Unix user account which receives read/write permissions on the documentary directory of the web server (e.g. /var/www) has to be created on the web server. This can be achieved, e.g., by creating a Unix group containing the user account of the web server and the newly created “web”.

The login via ssh at the “web” user account (with password) should be subsequently checked via another computer (Unix or Windows). Make sure that the ssh client uses the ssh protocol 2. Under Windows it is possible to use the ssh client `putty.exe`.

[34] rsync: http://rsync.samba.org/

[35] ssh: http://www.openssh.com/

10.2 Web server under Windows

> *Under Windows, most web servers open the files of pages to be viewed in an exclusive reading mode. This means that while providing a page to a web browser the corresponding page cannot be changed or replaced on the file system of the web server. Therefore, check whether this problem occurs prior to using this deployment method on Windows servers.*

1. Login as administrator on the web server.
2. Download setup.exe at http://cygwin.com and start it.
3. Enter, e.g., `c:\cygwin` as the installation target directory and leave the default options “Install for all Users” and “Text File Type Unix” unchanged. Choose for example `c:\cygwin\cache` as the “Local Package Directory”.
4. Upon a request for packages to be installed, retain the default setting and additionally select the packages “openssh” and “rsync” in the “Net” category for installation. Change the arrow icons, on the left next to the package name, from “Skip” to the version number (by clicking). Other required packages are thus automatically selected.
5. Installation is completed after clicking to the next window.
6. Finally, call the Cygwin shell via the Cygwin icon on the desktop or via the start menu. If, depending on the Windows system, you are prompted to update `/etc/group` and `/etc/passwd,` confirm update and follow the instructions. Subsequently exit the shell by entering “exit”, call it again and leave it open. A prompt regarding /etc/passwd should not appear.
7. Enter `ssh-host-config` in the Cygwin shell and answer the parameter query as follows:

Overwrite existing /etc/ssh_config_file?	**Yes**
Overwrite existing /etc/sshd_config_file?	**Yes**
Create local user sshd_server?	**Yes**
Should privilege separation be used?	**No**
Do you want install sshd as service?	**Yes**
Value for environment variable CYGWIN=?	**ntsec tty**

8. Start the system service “CYGWIN sshd” via the Windows administration.
9. Create a local “web” user account which receives write permissions in the document directory of the web server.
10. Update the Cygwin user database by entering the following command in the Cygwin shell:

```
mkpasswd -l > /etc/passwd
mkpasswd -l > /etc/group
```

11. Check login via ssh at the “web” user account (with password) via another computer (Unix or Windows). Make sure that the ssh client uses the ssh protocol 2. Under Windows, it is possible to use the ssh client `putty.exe`.
12. Enter the following during login under the “web” user account to enable ssh login via a key pair: `ssh-user-config`
    (Answer the prompt for generating the identity files with “no”.)

10.3 FirstSpirit Server under Unix

The client installation for ssh and rsync should take place via the package system of the operating system.

The rsync/ssh web server connection is subsequently configured:

1. Login under the FirstSpirit Server user account (fs4) or use “`su - fs4`” as root.
2. Generate an ssh key pair: `ssh-keygen –t rsa`
   Accept the default specifications and do not enter a password.
3. Install the public key on the “web” user account of the web server. Login as “web” via ssh and create the directory `.ssh` in the home directory: mkdir .ssh
   As user “fs4” enter the following on the FirstSpirit Server:
   `scp .ssh/id_rsa.pub web@www.mydomain.net:.ssh/authorized_keys`
4. Enter the following to check login via the key pair (without password): `ssh web@www.mydomain.net ls -la /var/www`
5. The document directory of the web server should now be listed without having to enter a password.
6. To test the rsync connection, enter:
   `rsync -n -vcrtz -e "ssh -l web" www.mydomain.net:/var/www`
7. The document directory of the web server should be listed again without having to enter a password. Files have not been transferred due to the option “-n”.

If a Windows system is used as a web server, replace `/var/www` by `/cygdrive/c/www`.

10.4 FirstSpirit Server under Windows

The installation packages of Cygwin[36] are recommended as ssh client and rsync under Windows. Cygwin offers a convenient package administration which enables future updates of the individual packages via the Internet without further configuration. The installation and configuration of OpenSSH and rsync on a Windows-based web server is described below:

1. Login as user with administrator rights.
2. Download setup.exe at http://cygwin.com and start it.
3. Enter, e.g., `c:\cygwin` as the installation target directory and leave the default options “Install for all Users” and “Text File Type Unix” unchanged. Choose, e.g., `c:\cygwin\cache` as the “Local Package Directory”.
4. Upon a request for packages to be installed, retain the default setting and additionally select the packages “openssh” and “rsync” in the “Net” category for installation. Change the arrow icons, on the left next to the package name, from “Skip” to the version number (by clicking). Other required packages are thus automatically selected.
5. Installation is completed after clicking to the next window.
6. Finally, call the Cygwin shell via the Cygwin icon on the desktop or via the start menu. If, depending on the Windows system, you are prompted to update `/etc/group` and `/etc/passwd,` confirm update and follow the instructions. Subsequently exit the shell by entering “exit”, call it again and leave it open. A prompt regarding /etc/passwd should not appear.

The rsync/ssh web server connection is subsequently configured:

1. Login as user with administrator rights on the Windows system of the FirstSpirit Server.
2. Call the Cygwin shell via the start menu.
3. Generate a ssh key pair:
   ```
   ssh-keygen -t rsa
   ```
   Accept the default specifications and do not enter a password.
4. Use the Windows Explorer to change the access permissions to the file
   ```
   c:\cygwin\home\username\.ssh\id_rsa
   ```
   to ensure that the SYSTEM user account is also granted reading permissions. This is necessary, since the FirstSpirit Server is running under the SYSTEM user

[36] http://cygwin.com/

account.
"username" is in this case the current user name with which "ssh-keygen" has been called during the last step.

5. Install the public key on the “web” user account of the web server. Login as “web” via ssh and create the directory `.ssh` in the home directory:
   ```
   mkdir .ssh
   ```
   Enter the following in the Cygwin shell on the FirstSpirit Server:
   ```
   scp .ssh/id_rsa.pub web@www.mydomain.net:.ssh/authorized_keys
   ```
6. To test the login via the key pair (without password), call the prompt on the Windows system of the FirstSpirit Server (cmd.exe) and enter:
   ```
   c:\cygwin\bin\ssh web@www.mydomain.net ls -la /var/www
   ```
7. The document directory of the web server should now be listed without having to enter a password.
8. To test the rsync connection, enter:
   ```
   c:\cygwin\bin\rsync -n -vcrtz -e "c:\cygwin\bin\ssh -l web"
   www.mydomain.net:/var/www
   ```
9. The document directory of the web server should be listed again without having to enter a password. Files have not been transferred due to the option “-n”.

If a Windows system is used as a web server, replace `/var/www` by `/cygdrive/c/www`.

10.5 FirstSpirit project configuration

The ssh/rsync deployment entry occurs via the FirstSpirit client for the Project and Server Configuration:

1. Create a new schedule entry via the Project Configuration (see section 7.5.4 page 318).
2. Add the “Generate project” action to the schedule entry (see section 7.5.9.2 page 332).
3. Add the “Execute script” action below the previous action (see section 7.5.9.4 page 346).
4. There is an “ssh/rsync deployment script” in the appendix (see section 12.5 page 478).
5. The web server parameters are entered as script parameters:
   ```
   webhost=www.mydomain.net
   webuser=web
   webpath=/var/www
   ```
   If FirstSpirit runs under Windows, also add the following parameters:
   ```
   ssh=c:\cygwin\bin\ssh
   rsync=c:\cygwin\bin\rsync
   ```

```
privkey=/home/username/.ssh/id_rsa
```

"user name" is in this case the current user with which "ssh-keygen" has been called. "/home" is used in fact here under Windows because Cygwin maps this to the Windows profiles directory.

6. Subsequently test the entered configuration via the “Test” button and, if necessary, check the displayed error message log (see section 7.5.9.4 page 346). This deployment can only be used if the test is error free.

11 User permission configuration

11.1 Introduction

This chapter outlines the mechanisms for permission assignment and permission check provided by FirstSpirit and their precise application. The following sections only deal with permission assignment for the generated site (i.e. user permission assignment) and not with project permission assignment (i.e. editorial permissions) or permission assignments for workflow execution. (For further information on permission assignment see the FirstSpirit Manual for Editors).

FirstSpirit strictly differentiates between editorial permissions and user permissions. While editorial permissions apply to all operations which can be executed by an editor (e.g. create/change/delete pages), the user permissions only apply to the "visitor" of the generated site and are, therefore, always linked to the used personalisation system. If FirstSpirit Personalisation[37] is used as the personalisation system (not mandatory), a very close relation can be established. (see section 11.2.3 page 454)

Within the scope of editorial permissions FirstSpirit specifies the number of operations (create/change/delete/release, etc.). These operations can be provided with permissions for persons or groups. Person/Group management is also carried out by FirstSpirit (even if an LDAP system can be connected). Therefore, the operations and groups are (relatively) fixed within the scope of editorial permissions.

In contrast to the editorial permissions which relate to processes in the FirstSpirit project, the user permissions exclusively relate to the generated and deployed site. The application of a login page usually indicates that a project works with user permissions.

Within the context of user permissions, FirstSpirit defines neither the operation nor the group structure, since each project implemented with FirstSpirit has completely different user permission requirements. Usually it is sufficient to interpret user permissions as "Permission to view an object". However, the "Change" or "Print" operations may also be relevant in addition to the "View" operation. In this case, a

[37] See FirstSpirit Personalisation documentation

distinction has to be made between the “View” and “Print” operations within the scope of user permissions.

Please note that there is a relation between editorial permissions and user permissions in exactly two cases:

1) Page preview:
In this case the editor is also a user – the editorial permissions “View” and the user permissions “View” coincide with each other and have to be linked appropriately.

2) Changing data of the live site[38]:
In this case the user is also an editor – the user permissions “Change” and the editorial permissions "Change" also have to be linked appropriately.

The link is usually created via an additional login request, i.e. the user logs in as an editor or vice versa.

In addition, the following features have been realised in FirstSpirit Version 4.0[39]:

- SSO: If an SSO module is used, the transition from editor to user takes place transparently without a password request (login module: FS SSO).

- Selectable user persective for the editor: In order to check whether a user permission configuration functions as desired by the editor, a simple changeover of the user permission is desirable. This is possible by configuring a special login servlet for the preview which enables explicit group selection.

11.1.1 Define user permissions

User permission assignment is always based on groups, since experience has shown that managment on the user level leads to major problems, e.g. for arranging representatives.

In order to structure and, therefore, facilitate permission assignment, it is assumed that groups can have a hierachical structure – i.e. a group can contain several sub-groups.

[38] See FirstSpirit Integration documentation

[39] See FirstSpirit Personalisation documentation

FirstSpirit helps to redefine these group structures or to import them from an existing system (e.g. LDAP).

Irrespective of its origin, the group hierarchy is presented to the editor in a tree view in which it is possible to configure the permissions. The permission component is thus a special input component which can be used to assign permissions on the basis of a hierarchical group definition. This permission definition exclusively refers to the runtime system and not to the editorial system, i.e. no editorial permissions. The permission component is usually used within the scope of the metadata. Nevertheless, it could also be used in the Page-Store or Content-Store.

Besides the group hierarchy, there is also a relation to the tree structure of the FirstSpirit administrations which is also interpreted as hierarchy.

The tree structure of the FirstSpirit administrations represents an inheritance relation for the permission assignment. Therefore, the following always applies: If user permissions have not yet been defined in a tree object, the permissions of the parent object apply. Due to this inheritance definition it is quite easy to define permissions for subordinated pages, e.g. on a folder layer.

The inheritance is, therefore, defined as “not additive” – i.e. the permission definition in an object overwrites all “superordinated” definitions.

Since this extremely simple inheritance model is not always suitable, there is a number of options to project-specifically define “plausibility rules” for the permission assignment (e.g. “if something is allowed for a superordinated group, it cannot be forbidden for a subgroup” or “if somebody is allowed to view an object, he/she must also be allowed to enter the superordinated subtree, otherwise he/she would never be able to reach the object”).

11.1.2 Check user permissions

After describing the user permission definition, different methods of checking user permissions are presented below.

The definition of permissions only makes sense if these permissions are also evaluated and considered during document provision.

This requires a runtime component which enables checking. To carry out a check, the user has to be identified and his/her groups determined. This function is provided by the FirstSpirit Personalisation module.

Another logical consequence is the need to generate an effect from the permission evalutation – i.e. a reaction. Sometimes the reaction occurs within the scope of personalisation.

This involves parts of a page or even the complete page being protected by special tags (FirstSpirit Personalisation tags[40]). This reaction type is only possible with JSP pages. This method cannot be used to protect pure HTML pages, PDF documents or images. For this reason the module FirstSpirit SECURITY is available in addition to this concept (see chapter 11.3.2 page 456). This module prevents document or file provision, depending on the permission configuration, on the HTTP server layer. A solution limited to media from the Media-Store and linked to the permission component is available as a "secure media" concept (see chapter 11.3.3 page 458). For more information about "Protection of personalised content" see chapter 11.3 page 159.

There is an analogy of the permission component in the runtime system (target/actual comparison of the permission configuration of the object with the user configuration) (see chapter 11.3.2 page 456). Basically, it is important to use the same group hierarchy relations.

For these purposes a filter will be used, capable of managing the provision of non-active documents. This "multi-access-control filter" can be configured to the needs of the project (see chapter 11.3.2.2 page 358).

These mechanisms are described in detail in section 11.2.3 page 454 ff.

11.1.3 Permission module

In FirstSpirit, the user permission assignment and check take place via the permission module. The module consists of two components:

- Editor (permission component) - (see section 11.1.3.1 page 448)
- Service (PermissionService) - (see section 11.1.3.2 page 449)

11.1.3.1 Permission component (CMS_INPUT_PERMISSION)

The permission component is a combination of GUI and render component. Use this component to extend the FirstSpirit-Client with user-specific input options, in this case permission definition. The permission component is usually used as an input component in the metadata tab. The component is (project-specifically in a metadata

[40] See FirstSpirit Personalisation documentation

form) parameterised with a unique group document and cooperates with the appropriate service which loads and provides the group definitions from the server (see section 11.1.3.2).

See the "FirstSpirit Manual for Editors (JavaClient)" for application of the permission component.

11.1.3.2 Permission service

The permission service is a server component which can be addressed via the permission component. It is a special FirstSpirit Server service responsible for managing group and user configurations. As a system service the permission service can be activated via FirstSpirit Server Monitoring (for configuration of the file `fs-server.conf` via FirstSpirit Server Monitoring see section 11.4.2 „Activate the permission service" page 463).

11.2 Architecture

11.2.1 Introduction

As already indicated above, the permission check is a complex aspect which concerns many parts of the FirstSpirit system. This section, therefore, presents the relation and dependencies of the individual parts on the architecture level. The main components of the permission check in FirstSpirit are:

a) Permission component (client): Defines the target permission for objects (usually via metadata).
b) Permission component (server): Defines the group and user structures on which permission definition is based.
c) Web applications (preview/staging/live): Are used to carry out a permission check during document provision.
d) External data sources: Provide the basis for group structure definition. An external (viewed from FirstSpirit) data source (e.g an LDAP server) is used here.
e) Internal data source: Provide the basis for the group structure definition. An internal (viewed from FirstSpirit) data source (e.g. a CS4 database) is used here.

The relations and the dataflows between the components are essential for understanding interaction between the individual components.

11.2.2 Overview

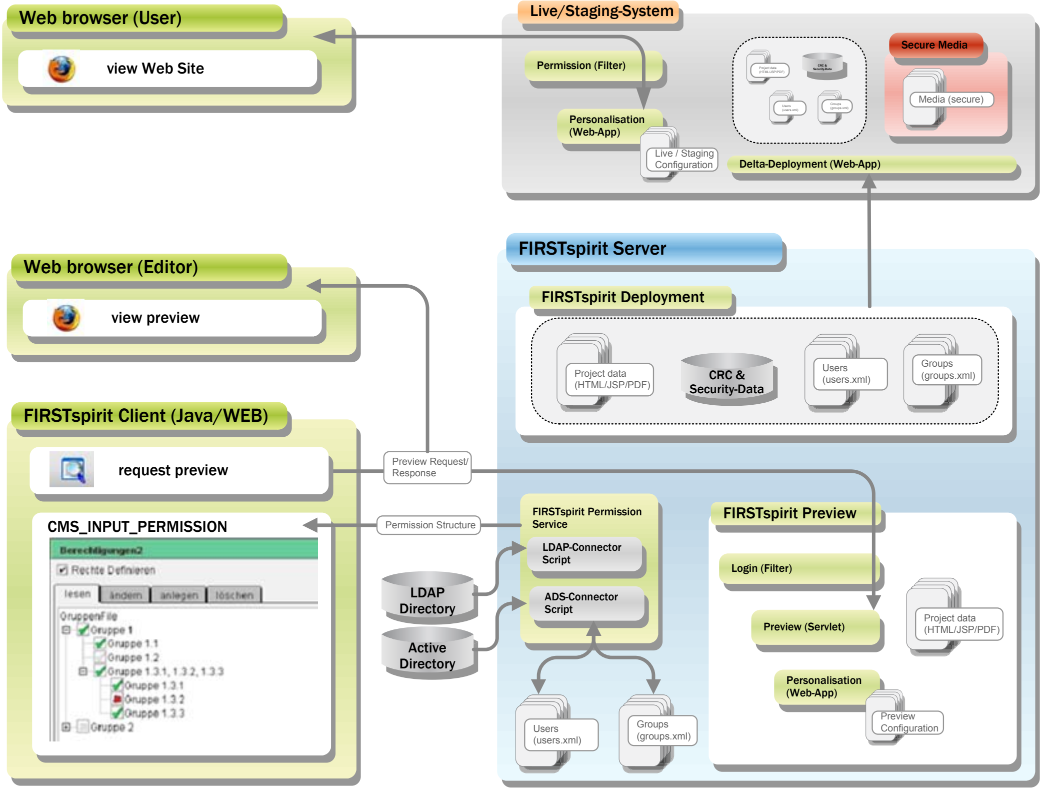


Figure 11-1: Live system overview

FirstSpirit-Clients: Communicate exclusively with the FirstSpirit server and never directly with, e.g., an LDAP server or database (for further information see section 11.2.2.1 page 451).

FirstSpirit Server: Processes all client requests, manages accesses to external resources (databases etc.), generates preview pages, attends to the staging and live system (for further information see section 11.2.2.1 page 451).

Staging system: (Also "generation directory") is used for the final check of the total system including, if necessary, occurred application integration. All web applications and modules of the live system have to run here. In contrast to the preview system, the directory and file names are identical to the structures of the subsequent live system. The configurations of all the web applications essentially correspond to the ones of the live systems, except for absolute paths and URL prefixes (for further information see section 11.2.2.2 page 452).

Live system: The system visible to the end user. All the relevant data within the scope of deployments is transferred into the live system (for further information see section 11.2.2.3 page 453).

11.2.2.1 FirstSpirit-Client, Server and preview generation

The permission component is usually used as an input component in the metadata tab. The component is (project-specifically in the metadata form) parameterised with a unique group document. The permission component uses this name to determine the corresponding structure in the FirstSpirit Server. This takes place via the permission service – this service is a special FirstSpirit Server service responsible for managing group and user configurations (for the configuration of the file `fs-server.conf` see section 11.4.2 „Activate the permission service“ page 463).

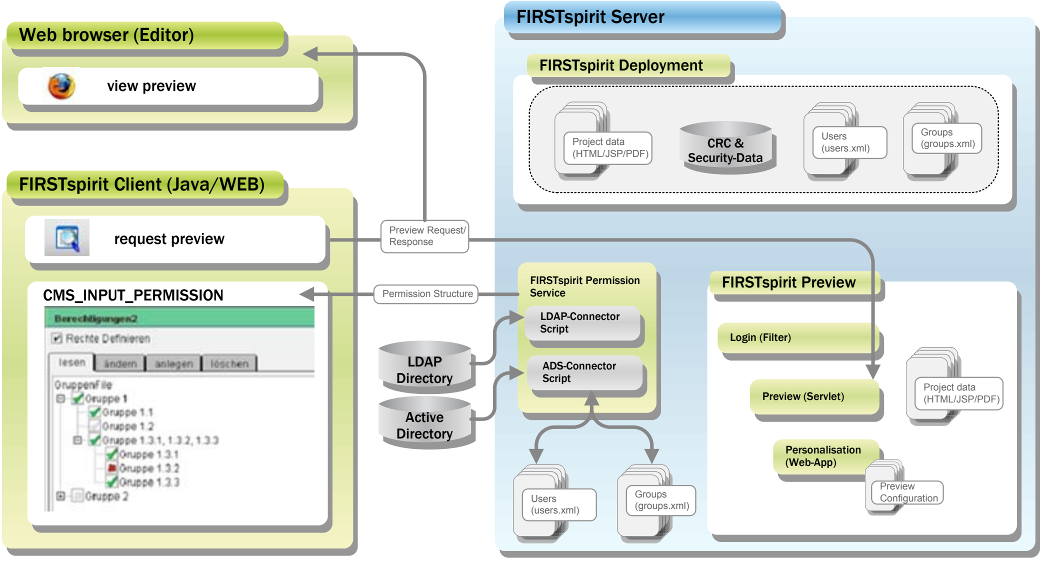


Figure 11-2: Live system extract

The group and user configurations managed by the “permission service” are configured in the service configuration file (“`service.ini`”) (see section 11.4.3.1 „Structure of the service configuration file `service.ini`“ page 464). The respective XML files (see FirstSpirit Manual for Developers) can be generated manually (via FirstSpirit Server Monitoring, see section 8.6.1.7 page 400), or automatically via a connector script based on an exiting user/group managment system (e.g. LDAP or Active Directory) (see FirstSpirit Manual for Developers).

Using a group/structure definition the editor can subsequently carry out a permission definition for an object in the FirstSpirit-Client.

As described in section 11.6 “Application in the project“ it is possible to use the permission definition as “target permission” for the personalisation (see section 11.3.1 page 455), or within the scope of the access protection using the Access Control Database and the Multi-Access Control Filter (see chapter 11.3.2.2 page 457).

At this point a further security mechanism steps in. This mechanism has no relation to the user permission definition, but should nevertheless be mentioned here. The "preview servlet". The preview servlet is responsible for monitoring the access permissions during a preview request (see chapter 11.3.3.1 page 458). The preview servlet provides the content and the request is internally redirected to the web server. This architecture is necessary if an HTTP server-specific extension is to be used, e.g. ASP or PHP extenstions in the preview. (Therefore, the mechanisms required for FirstSpirit V 3.1 via the security proxy do not apply.)

While the preview servlet only steps in on the access layer, the "multi-access-control filter" works on the object layer (see chapter 11.3.2.2 page 457). The main difference to configuration in the live system is that the files required by the FirstSpirit web applications (e.g. content schema or OR runtime, or `user.xml` and `groups.xml` for personalisation) can be accessed directly in the preview, while copies of the files have to be deployed in the live system.

11.2.2.2 Staging system (generation)

The staging system differs from the FirstSpirit preview system, particularly the file storage organisation. While all files are generated with "artificial" names and without directories in the preview system for performance reasons, the structure of the staging system is identical to the live system structure. Therefore, it is also possible to integrate applications which do not principally run within the preview (due to the already mentioned differences).

The basic structure of the HTTP server infrastructure is identical to the preview system structure (see section 11.2.2.1 page 451)

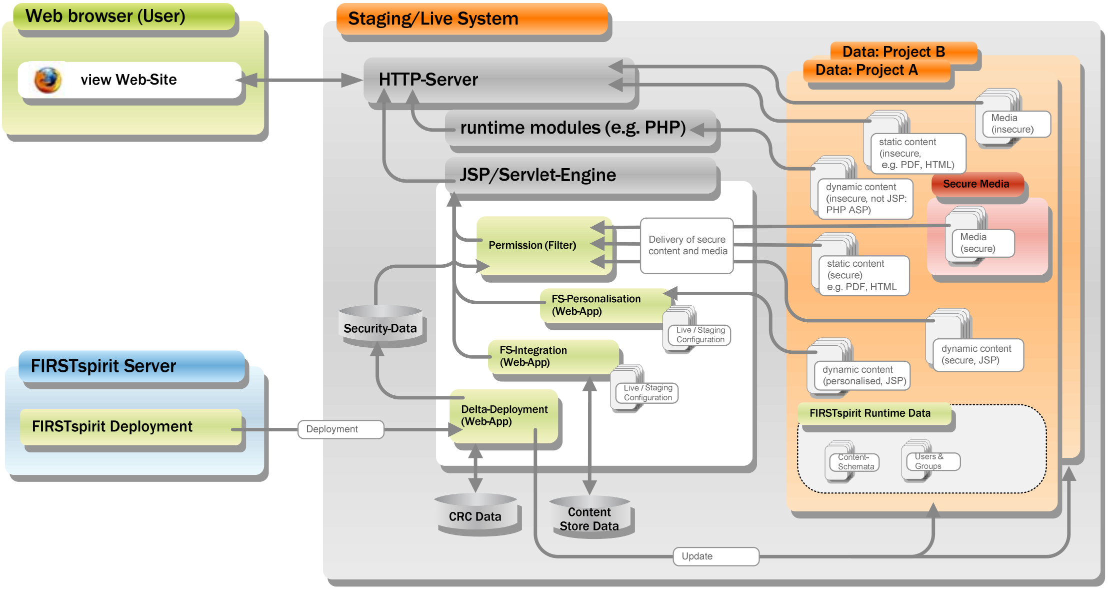

Web browser (User)
view Web-Site
Staging/Live System
HTTP-Server
runtime modules (e.g. PHP)
JSP/Servlet-Engine
Permission (Filter)
Delivery of secure content and media
FS-Personalisation (Web-App)
Live / Staging Configuration
FS-Integration (Web-App)
Live / Staging Configuration
Security-Data
FIRSTspirit Server
FIRSTspirit Deployment
Deployment
Delta-Deployment (Web-App)
CRC Data
Content Store Data
Update
Data: Project B
Data: Project A
Media (insecure)
static content (insecure, e.g. PDF, HTML)
dynamic content (insecure, not JSP: PHP ASP)
Secure Media
Media (secure)
static content (secure) e.g. PDF, HTML
dynamic content (secure, JSP)
dynamic content (personalised, JSP)
FIRSTspirit Runtime Data
Content-Schemata
Users & Groups

Figure 11-3: Live and staging system

The multi-access-control filter is only effective if the object is really provided by the servlet engine. This aspect will be illustrated in more detail using a PDF file which, e.g., has been generated from a page of the Page-Store, but is not a "secure medium". In an unfavourable case, the PDF document is provided directly via the HTTP server. The multi-access-control filter, which is part of the servlet engine, is unable to prevent the provision). It has to be ensured that the HTTP server request is redirected to the servlet engine where it is processed.

Please note that the content provision takes place via the servlet engine. This method is less efficient than direct provision via the HTTP server. Another disadvantage of the staging system and of the live system is that they communicate directly with the Access Control Database to check the permission.

11.2.2.3 Live system

In contrast to the staging system, the FirstSpirit Server cannot control the web application configuration in the live system. Therefore, configuration has to take place manually. One possibility is to use the automatically generated configuration files of the FirstSpirit web applications as the starting basis. Usually only a few paths have to be adapted to the live system (see Figure 11-3).

The live system is updated by the deployment manager located on the FirstSpirit Server. The deployment manager is responsible for the compilation of all relevant data and its upload onto the live system. In the live system, access check management occurs either independently or on the basis of the information of the

Access Control Database (see chapter 11.3.2 page 456).

11.2.3 Permission check via FirstSpirit

11.2.3.1 Quick start guide

The quick start guide presents the execution sequence of the respective steps to activate the permission check in FirstSpirit:

1. Installation of the permission service on the FirstSpirit Server (see section 7.3.14 page 232)
2. Start permission service: After module installation, the services are started and stopped either via FirstSpirit Server Monitoring (see section 8.6.2.4 page 402) or via the Server and Project Configuration (see section 7.3.14 page 232).
3. Adapt system configuration files (`service.ini`, `groups.xml` and `users.xml`) of the permission service (see section 11.4.3 page 464)
4. Create metadata template (see FirstSpirit Manual for Developers) `CMS_INPUT_PERMISSION` and `secure media`
5. Configure web components via the Server and Project Configuration (see section 7.4.17 page289).
6. Configure permissions via the Server and Project Configuration (see section 11.4.5 page 467)
   - Specify the input component for secure media
   - Specify the input component for a permission check
7. Configuration of the Multi-Access-Control filter (see chapter 11.3.2.2 page 457)

See also FirstSpirit SECURITY documentation.

Application scenarios and individual configuration options are described in detail below.

11.3 Protection of personalised project content in FirstSpirit

FirstSpirit has different concepts to protect project content from unauthorised access:

- Personalisation of content (see chapter 11.3.1 page 455)
- Checking the access rights to an object via the Access Control database (see chapter 11.3.2 page 456)
- Access protection for media within FirstSpirit preview ("secure media") (see chapter 11.3.3 page 458)

11.3.1 Personalisation

The module FirstSpirit Personalisation[41] consists of an easy-to-use Java TAG library for personalising JSP pages, i.e. page display can be completely or partially prevented. Whether a part area of a page is visible or not depends on the user permission. This is decided by comparing the target permission of the page to the user's actual permission.

The target permission (Who is allowed to view a document or a part of the document?) takes place via the permission component. The personalisation carries out the evaluation, i.e. a comparison of the actual configuration, resulting from the user's login context, with the target configuration. Within this context there is a close (logical) relation between the groups which are used in the permission component and the groups to which the "group module" of the personalisation refers.

<u>Example:</u> If a user belongs to a group which does not appear in the permission component, it might not be possible to set permissions for the user.

This relation between the group model in the permission component and the one in the personalisation module can be established as follows:

1. The generation of the group definition file for the permission component and the "group module" of the personalisation evaluate the same external data sources (e.g.: LDAP server or AD server). In this case, data sovereignty is completely external.

2. The permission component generates a group definition file and the personalisation module uses this data or refers to this file. The permission component can generate this file by requesting an LDAP server. The

[41] Module FirstSpirit Personalisation (see documentation PERS40EN_FirstSpirit_Modules_Personalisation)

difference is that the personalisation module does not request the LDAP server, but uses the data basis generated by the permission component.

While the “group module” of the personalisation is configured to the modes “Content-Store” or “LDAP” in the first case (and the runtime environment requires a permanent connection to the data source), the “group module” has to be configured as “group service” in the second case and then uses the same XML files as the permission component. Therefore, the second case does not require permanent access to an external resource. Nonetheless, it has to be ensured that the group configuration files are also deployed on the live system during deployment configuration (see section 11.4.4 page 466).

Permission checks can only be forced on JSP pages through use of personalisation in FirstSpirit. The personalised hiding of links can therefore be used to make access to non-JSP documents difficult (e.g. PDF files or pictures). However, this does not provide reliable protection as delivery can be forced by direct entry of the links. The module should therefore always be combined with the standard FirstSpirit SECURITY module (for details of concept see chapter 11.3.2 page 456).

The FirstSpirit Personalisation module is an additional function which has to be purchased.

For further information see the FirstSpirit Personalisation documentation.

11.3.2 Checking access rights via the Access Control Database

Therefore, protection must be provided at the level of the file delivery to realise reliable protection for objects which cannot perform a permissions check themselves. In FirstSporit the FirstSpirit SECURITY module is available for this purpose. When a project is generated a local Access Control database is created which manages all permissions information for the individual project content. Comparison/synchronisation of this local Access Control database with the live system takes place via the FirstSpirit SECURITY module (via the CRC Transfer Servlet – see FirstSpirit SECURITY module documentation). The module can be used, among other things, to define a filter ("Multi-Access Control Filter"), which reliably prevents delivery of all objects which match the filter criteria (The server does not deliver the protected files).

The familiar concept of "secure media" from FirstSpirit Version 3.1 can also be realised in the live system via the FirstSpirit SECURITY module and a filter individually adapted filter to the required "Secure Mediy" directory (or the medial file)[42].

Configuration takes place via FirstSpirit Server and Project Configuration (see chapter 7.4.16 page 287).

For further information, see documentation for the FirstSpirit SECURITY module.

11.3.2.1 Access Control Database

A sub-area of the FirstSpirit SECURITY module is the Access Control database. An Access Control database consists of several "Access Control Lists" (ACL). The Access Control database manages any project file information (e.g. pages, media). Apart from many other different types of information, for example the object's CRC 32 checksum required for differential upload (cf. chapter 7.5.9.3.3 Seite 343), the Access Control database can also be used to save the access rights to an object (i.e. rights of a user of the generated site – "user rights"). With the help of individually configured filters, which are also made available via the module, access to generated project content can be controlled in this way (see "Multi-Access Control Filter", chapter 11.3.2.2)

The local Access Control database of a schedule is created with the first generation and is updated with each further generation.

For further information, see documentation for the FirstSpirit SECURITY module.

11.3.2.2 Multi-Access Control Filters

The Multi-Access Control Filters can be used to define different filters for project content for which special access protection is required within the generated or deployed content. The task of the Multi-Access Control Filter is to check a certain class of objects before delivery with respect to their permissions configuration. The filter uses the information from the Access Control database for the check.

The Multi-Access Control Filter is realised as a servlet filter which must be configured in the staging and Live system in the relevant web application. A specific generation directory (or an individual object) can be given, which is to be checked by the filter. The validity areas (scope) of the filter can be defined within the scope of the mapping. To do this, a URL pattern is defined with the URL classes and specific file

[42] FirstSpirit SECURITY module (see documentation SECU40DE_FirstSpirit_Modules_Security)

filters can be described. The Multi-Access Control Filter can be further adjusted using parameters, for example by specifying specific access rights only to be checked or by limiting filtering to specific files (e.g. PDF files only). Several different filters can be configured within a configuration.

When defining the URL pattern, always consider that servlet execution prior to providing an object which matches a filter criterion results in increased computing effort. URL patterns should, therefore, always be kept as “small” as possible. This means that “/” should never be mapped if “*.pdf” is sufficient.

For further information, see documentation for the FirstSpirit SECURITY module.

11.3.3 Secure Media concept

A concept which users are already familiar with from FirstSpirit Version 3.1 are the so-called "Secure Media". Here specific files of the Media Store are protected against unauthorised access:

- Within FirstSpirit Preview generation via a servlet (Preview Servlet) (see chapter 11.3.3.1 page 458)
- Within the Live systems via a filter (Multi-Access Control Filter) (see chapter 11.3.3.2 page 459).

Both mechanisms, both the servlet and the filter, check the user's permissions before delivery.

11.3.3.1 Secure media within FirstSpirit preview

Within FirstSpirit preview, access protection for precisely one media directory can be defined in the "Permissions" area within the project properties (see chapter 7.4.16 page 287). There a Media Store folder is defined as a secure media folder. All content within the folder are then protected against unauthorised access within FirstSpirit preview (only there!).

Delivery of the media (within the folder) is prevented via a servlet ("Preview Servlet") within the preview generation. The task of the preview servlet is to monitor access permissions when a preview is requested (see chapter 11.2.2.1 page 451). Either the user rights are evaluated which are deposited for an object in the input components for user rights. The preview servlet checks against the PermissionService. The permissions deposited here correspond to the information deposited for an object in the Access Control database but are more up to date ("Current" status). If this information is not available the internal "Editing Rights" of

FirstSpirit are evaluated (for media: "can see").

The preview servlet is available via the standard web application fs4preview. Configuration settings extending beyond definition of the secure media folder are not necessary.

Installation of the web application FirstSpirit Security WebApp is not necessary in the "Preview" web area.

11.3.3.2 Secure media within the live system

> ! *Through the introduction of the FirstSpirit SECURITY module, changes have resulted in FirstSpirit Version 4.0 (and 4.1) for the configuration (in live status), as the access protection here is no longer limited to media but includes all project content.*

To protect secure media within the live system it is necessary to install the FirstSpirit SECURITY module and to configure an individual filter adjusted to the secure media directory (see chapter 11.3.2.2 page 457)[43]. In this case the rights are not checked via the preview servlet but based on information from an Access Control database (see chapter 11.3.2.1 page 457).

For further information, see documentation for the FirstSpirit SECURITY module.

11.3.4 Scope

All the mechanisms decribed above are designed to check a permission configuration. But when should which methods be used and how are they interrelated? Mandatory requirements for permission checks are:

- Utilisation of the FirstSpirit permission component
- Utilisation of the module FirstSpirit Personalisation

It is also possible to use a different permission component or personalisation module, but this would require adaptations.

Basically, permission checks on the section layer (i.e. within a page) are only possible via the FirstSpirit personalisation module. If independent objects (e.g.

[43] see documentation SECU40DE_FirstSpirit_Modules_Security.pdf

pages, media or PDF documents) are to be protected, the “secure media” mechanism (see chapter 11.3.3 page 458) or the “Checking of access rights via the Access Control Database” (see chapter 11.3.2 page 456 can be used in addition to personalisation.

11.3.5 Permission definition

11.3.5.1 Hierarchical semantics

The following aspects display the fundamental problem of using an input component for the permission definition:

1. The groups usually form a hierarchy which has to be normalised appropriately. This means conversely: Users with special functions have to be normalised as a separate group (see section 11.3.5.2 page 460).

2. The stores also form a hierarchy which is often used for inheriting permission definitions (see section 11.3.5.3 page 461).

3. A modification of inherited permissions is desired.

The interconnection of these three aspects results in a significant complexity for permission definition, since:

1. there needs to be a definition of what occurs when a subgroup is allowed to do something that a superordinated group is not. Analogue: Prohibition

2. the semantics of the store has to be defined:
   a) none (e.g. Media-Store/Page-Store), i.e. only inheritance.
   b) hierarchy (e.g. navigation via the Site-Store), i.e. inheritance and limitation

3. a distinction between “allowed”, “prohibited” and “inherited” is required.

11.3.5.2 Hierarchical semantics (groups)

A group hierarchy is an organisation in which groups can consist of individual groups.

In FirstSpirit, permissions can be defined on each group hierarchy layer. This functionality has numerous advantages, especially when configuring user permissions, since it is possible to define permissions on a superordinated group which are subsequently valid for all subgroups. This means the number of required configurations can be kept to a minimum even for numerous groups.

In the FirstSpirit permission component, it is possible to define one of the following values on each group hierarchy layer:

- Allowed
- Forbidden
- Inherited (i.e. corresponds to the setting of the superordinated group)

Permissions can be defined on each layer – the FirstSpirit permission component does not make any assumptions regarding the "correctness" of the configuration. Plausibility checks or validations can be realised project-specifically.

The introduction of group hierarchies also leads to a number of problems:

1) What happens with "contradictions", e.g. when a subgroup is allowed to do something that the superordinated group is not?

2) Is a superordinated group an independent entity?

3) What happens if the group hierarchy changes?

Since it is impossible to answer these questions universally for each application case, the user permission concept offers a means to map all possible answers. This means:

1) The list of the allowed and forbidden groups is managed separately and can be evaluated individually. Individual priority strategies can, therefore, be realised.

2) A superordinated group is an independent entity if it has been allocated an ID. If an ID has not been defined, the ID of the group is "calculated" as the union of all the IDs of the subgroups.

3) Changes to the group structure are covered via ID maintenance and notification mechanisms.

11.3.5.3 Hierarchical semantics (stores)

The permission definition is closely connected to the semantics of the stores. There are two possibilities:

1. no hierarchical semantics, except for inheritance (e.g. Media-Store/Page-

Store)

2. Inheritance and constraint hierarchy (e.g. navigation via the Site-Store)

In the first case, the store tree does not define hierarchical semantics except for the inheritance (e.g. Media-Store/Page-Store). In this case, various anomalies might occur, e.g.:

- A user has the required permissions for a document but cannot access the document, since he/she does not have permission for the page to which the document is linked.
- A link can also point to a document for which the user does not have permission.

Both cases are actually incorrect configurations. However, they are difficult to detect and, therefore, difficult to prevent.

In the second case, the store defines a constraint hierarchy in addition to the inheritance hierarchy. This means the number of authorised persons along the tree can be limited, but not extended. This is recommended if the tree hierarchy has the form of a hierarchical menu, since access to a “subordinated” tree element can only take place via the “superordinated” node in this example. Therefore, a permission extension does not make sense, since the superordinated entry point is missing.

In summary, behaviour in the second case can reduce the chances of misconfigurations in certain cases. However, the configuration visualisation demands are higher.

The FirstSpirit implementation offers the required infrastructure to realise both variants.

11.3.6 Definitions

The following definitions apply when setting user permissions:

1. The store structure only has hierarchical semantics for the inheritance. Limiting the permission configuration along the store tree is not intended. Nevertheless, mechanisms have been provided which allow for the validation of the permission definition with project-specific semantics (e.g. script hooks for the “Check” button or “on-load” / “on-save” scripts).

2. A modifying inheritance is basically excluded. Each permission definition has

the character of a permission definition point (i.e. the permissions are copied and then modified). In order to use this concept in the field, additional help functions are realised (see convenience methods, section 11.6.2.1 page 470).

3. The following three states are defined in the group tree:

   - ☑ Allowed – green tick
   - ☒ Forbidden = red cross
   - ☑ ☒ like parent = either grey tick or grey cross.

   Click through the three states. If necessary, the grey icons in the suordinated subtree might also be changed.

4. Dependency propagation: Default strategy (grey), i.e. the children have the same state as their parent by default, but it is possible to carry out an explicit change.

These settings result in a manageable behaviour of the permission component. The cause and effect are predictable, since the complete configuration state can be viewed at a glance. Limitations in regard to flexibility, particularly for maintenance, are reduced by a number of convenience methods.

11.4 Configuration

11.4.1 Introduction

The permission component consists of an input component displayed in the client and used by the editor for defining permissions. The input component determines the group configuration defined for the precise project by communicating with a FirstSpirit Server component. The server component sets up the group hierarchy (e.g. on the basis of an LDAP tree) and transfers the data to the FirstSpirit-Client.

11.4.2 Activate the permission service

The permission component and the permission service are FirstSpirit modules. The module is included ín the FirstSpirit scope of delivery and has to be installed on the FirstSpirit Server first (see section 7.3.14 page 232).

After module installation, the permission service is started and stopped either via FirstSpirit Server Monitoring (see section 8.6.2.4 page 402) or via the Server and Project Configuration (see section 7.3.14 page 232).

11.4.3 Configure the server component

The server component is configured via the service configuration file `service.ini` in which the general settings are carried out and optionally via a number of group definition files in XML format (see section 11.4.3.1). The service configuration directory for modules is located under the FirstSpirit directory in the subdirectory “conf/modules”. The individual service configuration files are located in a subdirectory with the name of the component (here: `System.PermissionService`) under “services”. The permission component is located in the file `services.ini` in the following directory: `conf/modules/System.PermissionService`

Extend the file `service.ini` to ensure the set-up of a group hierarchy based on LDAP queries. Since this might not be possible via static parameterisation, specify a “generating BeanShell script” as a parameter, if necessary. It is then possible to provide this script with an appropriate initial LDAP context.

11.4.3.1 Structure of the service configuration file `service.ini`

The file `service.ini` configures the permission service. The central task of the permission service is to provide the respective input components in FirstSpirit with group hierarchies. Various group hierarchies identifiable via a unique name can be defined here. Each project can, therefore, define an individual group hierarchy or multiple group hierarchies can be used in one project.

Basically, the group hierarchies can derive from two different sources:

1) From an explicitly specified XML file (created manually, or externally in an automated process). These XML files can be manually generated via FirstSpirit Server Monitoring (see section 8.6.1.7 page 400).
2) Via a script: The group hierarchies can be automatically generated via a connector script based on an existing user/group management system (e.g. LDAP or Active Directory).

In the second case, an XML file is also created (by the script) for buffering the results. The script is called by the service at defined intervals and can, if necessary, change the XML file. A script is usually used for generating a group file from an LDAP server.

The INI file contains the following global parameters:

`interval` = Period in seconds during which the ini file is checked for changes.
`documents` = List of the symbolic names of available group hierarchies.

If a group hierarchy is to be generated from an XML file, the following parameter is required:

NAME.path = Path to the group XML file

If the group hierarchy is to be generated via a script, a number of parameters are required:

NAME.path = Path to the group XML file
NAME.users = Path to the user XML file
NAME.script = Path to the BeanShell script (e.g. for automatic generation of user and group files - users.xml and group.xml).
NAME.script.intervall = Interval in seconds during which the script is to be called.

Default configuration of the file service.ini:

```
## global params
# -------------
# check each x seconds for changes
interval=20

# symbolic names for documents
documents=GroupsFile

#
# document specific params
# ------------------------
GroupsFile.path=groups.xml
```

Furthermore, it is possible to specify parameters for the LDAP connection:

NAME.ldap.URL = URL of the LDAP server
NAME.ldap.userDN = Login for the LDAP lookup
NAME.ldap.password = Password for the LDAP lookup
NAME.ldap.version = 2 (LDAP protocol version)
NAME.ldap.ssl = 0|1 sets whether the LDAP connection is established via SSL.

An LDAP context (javax.naming.directory.*) provided to the script is generated from the LDAP parameters specified here.

Example for a “service.ini” with LDAP configuration:

```
# 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=gruppen1.xml
GruppenLdap.script=gruppen1.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 Configuration for the deployment

The required files are deployed via a deployment servlet (see section 7.5.9.3.3 page 343). This is achieved by creating a deployment schedule entry in the Server and Project Configuration (see section 7.5.4 page 318).

The Deployment Servlet is a fixed part of the web application in the web server. This servlet enables the local file list to be compared (synchronised) with the web application in the event of deployment.

Through the CRC 32 checksum calculation for each file generated:

- changed and new files only can be transferred.
- generated files can no longer be deleted in the web application (optional).

This ensures that all up-to-date is generated but only changed data is also transferred. The data comparison (synchronisation) is carried out on the basis of the calculated CRC 32 checksums.

For further information, see documentation for the FirstSpirit SECURITY module.

11.4.5 Permission configuration via the project properties

Carry out the project-specific settings for permission evaluation in the Server and Project Configuration under the menu item “Permissions” (see section 7.4.16 page 287)

11.5 Application in the LIVE system

11.5.1 Servlet server configuration

Once all the required files and deployed files are available on the live system, the configuration files of the servlet server (e.g. Jetty, Tomcat, etc.) have to be adapted.

For configuration it is important that the directory “WEB-INF” is located either:

1. in the “target directory” from the deployment settings (see section 7.5.9.3, page 337 ff.). The “target directory” is then the web application directory which has to be mapped:

   Jetty:

```
<Call name="addWebApplication">
  <Arg>/</Arg>
  <Arg>TARGET DIRECTORY</Arg>
  <Set name="extractWAR">false</Set>
</Call>
```

   Tomcat:

```
<Context path="" docBase="TARGET DIRECTORY" />
```

Or:

2. next to the “target directory” from the deployment settings. In the second case, the “project directory” has to be mapped from the deployment settings.

   Jetty:

```
<Call name="addWebApplication">
  <Arg>/</Arg>
  <Arg>PROJECT DIRECTORY</Arg>
  <Set name="extractWAR">false</Set>
```

```
</Call>
```

Tomcat:

```
<Context path="" docBase="PROJECT DIRECTORY" />
```

11.6 Application in the project

11.6.1 Manual group definition and ID assignment

The implementation of the permission infrastructure assumes that each node in the group tree can either be clearly identified or has no meaning. The ID attribute in the “groups.xml” file provided by the server is used for this task.

Basically, it is possible to assign an independent, unique ID to each group. This assignment must occur if a group has no further subgroups (i.e. each leaf has to have an ID). If an ID is not to be assigned to group, the group ID is implicitly defined from the union of all subgroup IDs (transitive).

The semantics of an ID assignment means that persons can only exist in groups with an ID. If a group contains subgroups but not persons, it is not necessary to assign an ID. Such a group is only used for demonstration purposes. If at least one person is contained in a group, it is necessary to assign an ID. When assigning an ID, ensure that the IDs correspond to the group IDs determined during personalisation. Particularly observe that the group hierarchy in the personalisation module is not evaluated, since normalisation has already been executed by the permission component!

The composed ID is returned while evaluating the permission configuration:

```
All (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)
```

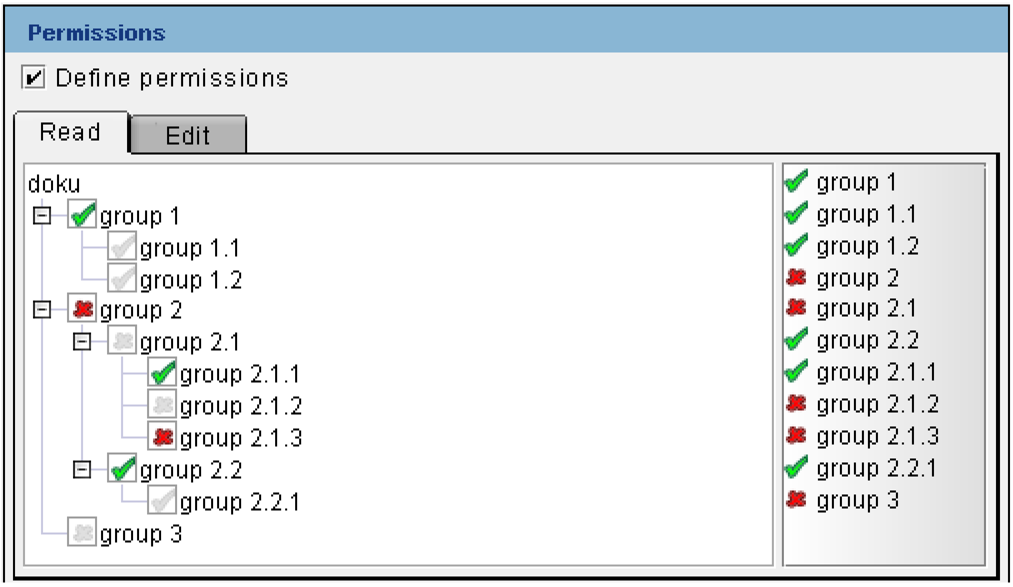
In this example “group 1” does not have an explicit ID. Therefore, the ID results from the union of all IDs of the subgroups. In this case “7” and “10”. If, e.g., a configuration is returned from the “group 1” layer, the result “7, 10” is returned. This means that there are no persons in “group 1”, but only in “group 1.1 (7)” and “group 1.2 (10)”.

This is different in the case of “group 2”. This group has the ID 9 and additionally diverse subgroups. This means that there are persons either directly in “group 2” or in the subgroups.

In contrast to the previous strategy, only a maximum of one ID per node will be allowed in the “group.XML” file. But it is also possible to omit the ID. The previously rarely used method of implicitly carrying out the group hierarchy evaluation with “clever” ID assignment is no longer supported, but is now an explicit function of the permission component.

11.6.2 Configuration sematics

The implications of a permission configuration in connection with a selected ID schema is not always trivial and is explained in more detail in the FirstSpirit Manual for Editors:



Permissions

☑ Define permissions

Read | Edit

```
doku
⊟ ✓ group 1
      ✓ group 1.1
      ✓ group 1.2
⊟ ✗ group 2
   ⊟ ✗ group 2.1
         ✓ group 2.1.1
         ✗ group 2.1.2
         ✗ group 2.1.3
   ⊟ ✓ group 2.2
         ✓ group 2.2.1
   ✗ group 3
```

```
✓ group 1
✓ group 1.1
✓ group 1.2
✗ group 2
✗ group 2.1
✓ group 2.2
✓ group 2.1.1
✗ group 2.1.2
✗ group 2.1.3
✓ group 2.2.1
✗ group 3
```

Figure 11-4: Permission component example

The new permission component is capable of assigning permissions for different operations. The operations are basically orthogonal, i.e. there is no “Operation A requires B” or “A leads to B”. Such functions are realised via project-specific validations/correction scripts which are called at appropriate periods (see the description in the “FirstSpirit Manual for Developers”).

The presentation of additional operations occurs as tabs (see Figure 11-4 “Read” and “Edit”).

11.6.2.1 Convenience methods

The decision not to support a “modifying inheritance” results in the following problem during project “maintenance”:

Permission definition changes: Permission definition changes do not influence the hierarchically subordinate part trees in the tree structure which start with a permission definition node (“Copy-on-Change” semantics of the permission definition point). These change classes must be used on each permission definition node in the subordinate part trees. To facilitate this task, it is possible to transfer the state of the subordinate node to a group node via a context menu each time the configuration is changed. After selecting the “Propagate permissions” context menu, a window opens which displays a list of all subordinate permission definition points. Select the respective nodes in this window and the state of the part tree on which the context menu has been opened is transferred to all subnodes after confirmation.

12 Appendix: Configuration files

Configuration files as created during FirstSpirit Server installation with default values in the installation target directory under `conf/`.

> ! *When copying configuration examples from the manuals (PDF files) it is necessary to ensure that all line breaks are correctly copied. If the characters are, for example, incorrectly coded on copying, this can result in problems with the configuration.*

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

A line with parameter values within the configuration file fs-wrapper.conf may not contain comments, e.g.:
`wrapper.startup.timeout=30 # Comment`

```
#*****************************************************************************
# 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 Maximem 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.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.NTLMLoginModule optional
domains=":my-netbois-domain;:my-active-directory-domain";
       // de.espirit.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 script

Further information to deployment via rsync / ssh see section 10 page 439.

```
//! Beanshell

// 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="-vcrtz";
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");
```